Proceedings of the

EUROSIM'98

Simulation Congress

EDITED BY

Kaj Juslin

for

The Federation of European Simulation Societies

HOSTED BY
Helsinki University of Technology
Espoo, Finland, April 14-15, 1998

Preface

EUROSIM'98 is the third conference of the Federation of European Simulation Societies. It brings together people from different organisations, interest groups and regional societies each third year to promote the advancement of modelling and simulation in industry, research, education, and development.

The EUROSIM'98 Simulation Congress is organised by SIMS - Scandinavian Simulation Society in co-operation with the other EUROSIM member societies: AES - Asociación Española de Simulación, ASIM - Arbeitsgemeinshaft Simulation, CROSSIM - Croatian Society for Simulation Modelling, CSSS - Czech & Slovak Simulation Society, DBSS - Dutch Benelux Simulation Society, FRANCOSIM - Société Francophone de Simulation, HSS - Hungarian Simulation Society, ISCS - Italian Society for Computer Simulation, SLOSIM - Slovene Society for Simulation and Modelling, UKSIM - United Kingdom Simulation Society.

Other Sponsoring Societies are: CASS - Chinese Association for System Simulation, JSST - Japanese Society for Simulation Technology, LSS - Latvian Simulation Society, PSCS - Polish Society for Computer Simulation, ROMSIM - Romanian Society for Modelling and Simulation, SCS - Society for Computer Simulation.

The local Scientific Committee at Helsinki University of Technology is formed by: Prof. Sirkka-Liisa Jämsä-Jounela (Chair), Laboratory of Process Control and Automation; Prof. Raimo P. Hämäläinen, Systems Analysis Laboratory, Prof. Heikki Koivo, Control Engineering Laboratory, Prof. Carl-Johan Fågelholm, Laboratory of Energy Engineering and Environmental Protection, Dr. Riitta Smeds, Laboratory of Business Process Modelling, The International Scientific Advisory Committee is formed by the board members of EUROSIM. General congress chair has been ,

MONILA has made the printing. *AREA Congress Team* has been engaged in the practical arrangements and *LORDHOTEL* for the Congress dinner.

The emphasis of the congress has been to provide a forum for **just on time** research presentations. The collection of papers was made by e-mail. The last paper updates arrived only one week before the congress. This proceedings document in three paperback volumes of totally 630 pages of contributed papers is intended as working material, during and after the congress. Selected papers are proposed for reviewing and enhancements aiming at inclusion in special issues of the Simulation Practice and Theory journal published by Elsevier Science Publishers B.V.

My sincere thanks to everybody involved to make the congress a fruitful and rewarding experience.

Kaj Juslin
General Congress Chair

Contents

B8 UNIT OPERATIONS 2

B9 COMMUNICATION

D6 ENTERPRISE

C6   ADDITIONAL TOPICS

GENETIC OPTIMISATION OF FUZZY POLICIES IN MODELS OF MANAGED SYSTEMS

Jacques LeFèvre [1, 2,3] , P. Weller [3,4]

1- LAIL URA CNRS D1440, Ecole Centrale de Lille, BP 48, F59651, Villeneuve d'Ascq, Cedex, France, jlefevre@lails1.ec-lille.fr
2- Sherrington School of Physiology, UMDS – St Thomas's Campus, London - SE1 7EH – UK, j.lefevre@umds.ac.uk
3- IDEA.SIM LTD, Modelling and Simulation Consultancy, Lowther Hill 7, London, SE23 1PZ, Tel/Fax: 44 (0) 181 690 80 37
4- MIM Centre, City University, Northampton Square, London – EC1V OHB - UK

Introduction

This paper discusses first the structure of the models used today to represent managed systems. We then present a new modelling method called Transformation System Dynamics (TSD) [1]. It provides a simpler and more intuitive model description than Forrester System Dynamics [2], which is the currently predominant method in this field. TSD is based on a kinetic description of the main stores, transformation processes and control policies of transformation rate (TRCPs) in a system. In the second part, we show briefly how and why to fuzzify most of these TRCPs. We represent them by neuro-fuzzy networks which may be designed explicitly, by learning from data or finally by genetic algorithms and genetic programming.

The pros and cons of modelling managed systems with Forrester System Dynamics (FSD)

We live in a world of complex managed systems (private and public enterprises, socio-economic development, environment and health care...). More and more, either as managers or even simply as stakeholders, we need to understand how and why these systems behave. These behaviours are dynamic and caused by the interplay of many factors acting in complex feedback loops and at different time scales. To understand them, we need to think about long sequences of intertwined causes and effects. We human beings are not well equipped to deal with such multi-factorial and circular reasoning. Obviously, the use of good computer models and simulations may be of great value if ... we can develop them.

Obviously, these managed systems are often very complex and ill defined. Building realistic and detailed models is most of the time impossible and this is not what we want. Our goals are rather to make explicit, to develop coherently and to test rigorously the mental models used by people when dealing with these systems in order to test them, to improve them and to discover all their sometimes well hidden consequences. Since these mental models are much simpler than reality, they are more realistic targets for mathematical modelling. Obviously, the methods used depend on the goal and level of details of our studies. In this paper, we are concerned with strategic, long range and global phenomena and policies. At this level, the method currently favored by most modellers is Forrester System Dynamics (FSD) [2]. It is a lumped approach focusing on *"levels"* i.e. characteristic quantities defined by the solution of a differential equation defining their rate of variation. Some of these levels are *"extensive"* and analog to tanks or stocks (with a conservation principle, e.g. money, matter, people, products...) and some are not (e.g. motivation, knowledge, tendency to buy...). Modelling a system starts by listing its *"levels"*. A *"network of flow processes"* links these levels together emptying and filling the various stocks. These flow processes are analog to uni-molecular chemical reactions. However, the algorithms f (.) determining the flows may be much more complex than the ones used in chemical kinetics. We may for instance use Volterra functional, differential operators or even expert systems. Generally, we have therefore:

$$A \xrightarrow{\quad f(a, B, x_1,...,x_n, u_1,...,u_m, t) \quad} B \tag{1}$$

In (1), A and B are stocks and the flow f=-dA/dt = dB/dt is defined explicitly by an algorithm dependent in general of A, B, other stocks $x_1,... x_n$, external variables $u_1,...u_m$ and t. Any FSD model is thus given by the diagram representing of N reactions (Fig.1). Many systems may be usefully studied at this kinetic level. Moreover FSD is supported by very good packages (e.g. STELLA™, trademark of HPS Inc. Hanover, USA). These facts explain its success (journals, books, professional societies, commercial and industrial applications).

However we believe that FSD should be much improved in several respects. (1) The restriction to flows with only one origin and one destination (uni-molecular) often results in messy spaghetti-like flow networks (continuous lines) showing few similarities with the structure of the real systems, as we perceive them in our mental models. (2) The control structures (interrupted lines) are also too messy and often too detailed and thus impede rather than favor understanding. (3) The flows may only be described by classical mathematical
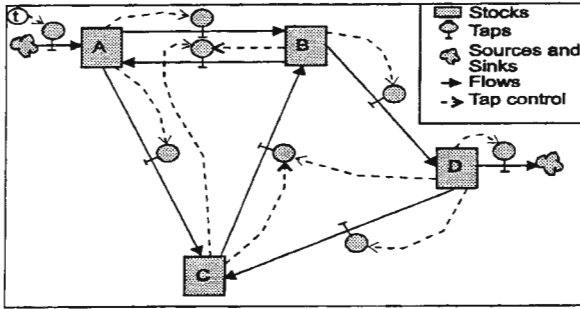
formulations, which are often over-detailed in applications where most of our knowledge on the intervening factors is fuzzy. In the last few years, one of us (JL) has developed a new graphical modelling method called "Transformation System Dynamics (TSD)" which eliminates these problems and includes FSD as a subset.

Fig.1: FSD: a network of generalised reactions describing flows between stocks (continuous lines). The flow rates are defined by equations given in the taps and dependent on the variables linked to the taps by interrupted lines (the flow control structure).

Transformation System Dynamics: A new FSD with Bond Graph-like notations

Our goal is just to introduce TSD's main concepts on an example (Fig.2) to show that it represents managed systems by diagrams which may be understood easily. Like in FSD, A TSD model is made of two parts:

- *A level and flow network* (see Figure 2 for a simple example). It is made of levels (e.g. [Blo], [Rip]) linked by continuous arrows which denote the flows defined by transformation processes (e.g. <Transfer>, Prod>) with characters < and > indicating the directions of flows. In contrast with FSD, each process is analog to a generalized pluri-molecular reaction. It represent processes like Production (materials + resources give products), Sales (orders + products give money and satisfied customers), Population growth (people + food give more people), Epidemic development (sick people + healthy people give more sick people), Bio-reaction (bacteria produce bacteria and products) and many others. Using general notations, we have:

$$nA + mB + ... \xrightarrow{\quad f(A, B, C, D, x_1 ,..., x_n , u_1 ,..., u_m , t) \quad} pC + qD + ... \tag{2}$$
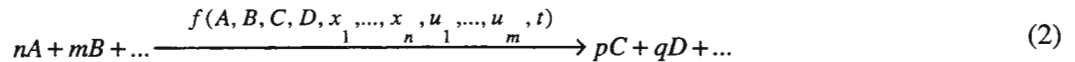
This process accepts different amounts (coefficients n, m) of several resources (A,B ...) and transforms them into products (C,D,...) produced also in different quantities (p, q). The symbols J (junctions) show the reunion of these inputs (A, B .. in packs having the proper amounts nA and mB of each resource and their separation (C,D,...) at the exit of the transformation. The transformation rate (number of packs transformed per unit time is defined by an algorithm f (...) which may be algebraic but also, as we said before, much more complex including for instance time delays, differential or functional operators and even fuzzy rules or neural networks. Very often the flow computation may be factored in two parts

$$f = f_c (A, B,..., C, D,...) \otimes f_d (x_1 ,..., x_n , u_1 ,..., u_m , t) \tag{3}$$

Where $\otimes$ is a suitable operator; f_c (c for close) gives the dependency on the levels directly linked to the flow (donors and acceptors) and f_d (d for distant) expresses how other variables influence the flow. Like the flows, the packing coefficients n, m, p, q... may be controlled by similar equations.

- *A flow control structure* (interrupted lines) sending the values of some variables (mainly stocks) to square boxes which may be decomposed in hierarchical block diagrams and define the algorithms (2) or (3) computing the transformation rates. If the flow algorithm is expressed in the non-separated form (2), all the variables appearing in it must be linked to the square box, which then send its result to the flow process. In the separated form (3), the box only receives the variables of the distant part and computes f_d. The flow process receives the values of the close variables through a causality process similar to the one existing in bond graphs and computes f_c before to compose f_c and f_d. In the model of Fig.2, this causality process is not illustrated and only f_d , the distant control structure, has been represented. It is obviously impossible to give here all the details (see [1]). The reader is invited to study the legend of Fig.2, which is intended to show that a FSD diagram may be understood easily and provides a natural description of a system. Obviously, the system illustrated there is a bit naïve; realistic models are much bigger but as easy to grasp.

Currently, we have used TSD with more than 150 students (managers, informaticians, engineers, and biologists) and in many fields (business, production, regional development, ecology, epidemiology, biotechnology, ...). In each case, TSD provided a simple solution much easier to understand, maintain and improve than the FSD models. It is implemented as a library TSD-LIB™ in the modelling and simulation package 20-SIM™. (TSD-LIB: trademark of IDEA.SIM LTD, London, UK; 20-SIM: trademark of ControLLab BV, Enschedde, NL)
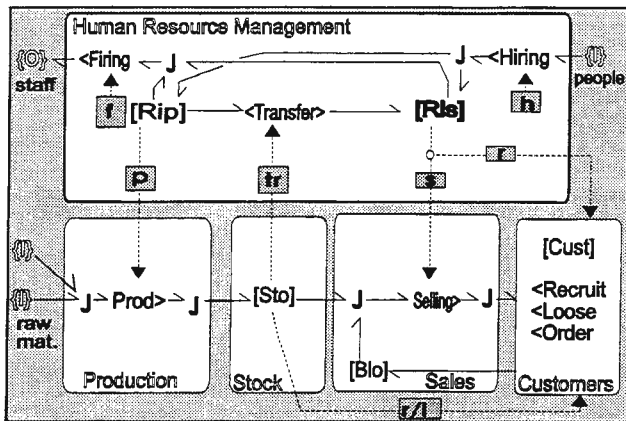
Fig.2: A simple TSD model showing 4 sectors of a company (Human resource management, Production, Storage and Sales/Services) and a Customer sector. Two kinds of raw materials enter from sources {I} at the lower left side. They are joined in a junction J and enter into a production process Prod> which puts its products into a warehouse [Sto]. From there, the products are paired in a J with orders from a store of backlog orders [Blo] coming from the customers. These pairs go into a sales process sending its products to the customer sector. This sector is unspecified in the figure which only shows that is has a store of customers [Cust] and three processes: recruitment of customers from a pool (market), loss to competitors and placement of orders. The human resource sector has two stocks [Rip], resources in production and [Ris], resources in sales. People may be hired to, transferred from and fired from these two stocks by the appropriate processes (e.g. people hired separate at a junction J and go either toward [Ris] or [Rip]. The rates of the processes are computed by a flow control structure expressed in separated form (3). The Figure also shows the distant control structures f_d (interrupted lines and control boxes) for each process. For instance, the block P computes the distant part of the production rate from RIP, the resources in production. Similarly, the block Tr computes the distant part of the staff transfer rate from [Sto], the current stock of products. For the sake of simplicity the causal structure determining the close control structure (f_c) has not been represented. It transfers to each process the values of its donors and acceptors. These values are then used with the distant informations from the control boxes to compute the flow rate according to algorithms like (3).

Representation of internal system controls and management policies in TSD

We have only scratched the surface of TSD and many of its concepts have not been discussed. In other works, we have made our example more realistic by including for instance a multi-step production with JIT and explicit money accounting, a detailed view of staff training and employment, a separate consideration of various product lines, a de-aggregation of the sales/service department and a detailed market model. However, Fig.2 serves its simple purpose: to suggest that TSD diagrams correspond well to the way we see our systems. Indeed, we can explain it easily to any manager and at the same time use it as a full model definition from which 20-SIM automatically generates the simulation code (for both 20-SIM™ and MATLAB™, (MATLAB, a trademark of MathWorks Inc., Natick, USA). Looking back to the problems mentioned earlier, our claim is that TSD solves the first two: the flow and control structures are now intuitive, simple and hierarchic, thus promoting the design of advanced models. In the rest of this paper, we will address the third problem: the fact that the representation of rates and packing coefficients by classical algebraic formula is often unrealistic. Control structures have often two different parts: (1) laws determined by the system itself and which cannot be controlled by us and (2) management actions. We call them respectively *"internal system controls (ISC)"* and *"management policies (MP)"*. An example of such a composite structure is given in Fig.3, which gives a possible model of the staff transfer process <Transfer> in Fig.2

As the example shows, in most managed systems, the definition of ISCs and MPs by exact mathematical formulas seems a bit contrived and artificial. Why this curve and not another slightly different? Why this formulation or parameters? Why these variables and not others (for instance in our example, Blo and Cust)? Obviously, we may test various curves and parameters by stochastic sampling. We might use these tests to produce robust results. However, even if we do so, the experts of the real system, when asked, will only produce qualitative verbal rules like *"If our stock of products is getting rather small, we had better to increase the production rate by transferring some staff to production quite fast"*. This kind of knowledge is better suited to fuzzy rules and its use in modelling managed systems is long overdue. It is an important subject and its implementation in TSD will now be described.

Close Influences (RIS, RIP), transmitted along flows

$$I_{RIS} = Max\ (0,\ H(a,thr,expo,\ RIS)$$
$$I_{RIP} = Max\ (0,\ H(a,thr,expo,\ RIP)$$

Distant Influence (Sto), computed by box Tr

$$y = maxtr \cdot tabl(u);\ u = (DesSto-Sto)/DesSto$$

Transfer rate, computed in process <Transfer>

$$f_{trans} = If(y>=0)\ then\ (k^*y^*e_{in})\ else\ (k^*y^*e_{out})$$

Neglected: Eventual influences of Blo and Cust..?
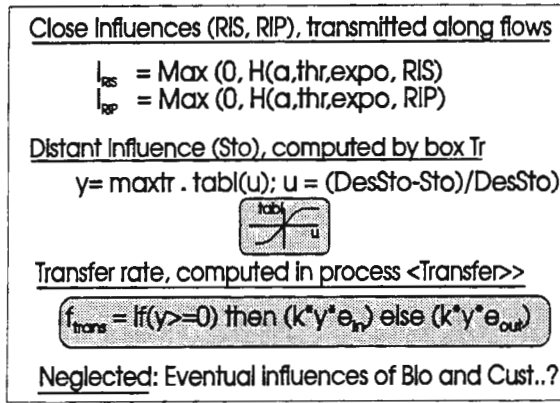
Fig.3: Determination of <Transfer>, the rate of staff transfer from Ris to Rip or Rip to Ris (see Fig.2). It depends on three influences. First we have the two close influences of [Ris] and [Rip]. They are not shown on Fig.2 but may be visualized as transferred along the flow lines joining these stores to the process. They are computed by the expression of I_{RIS} and I_{RIP} in Fig.3. H is a Hill function and a, thr and expo are its parameters. These close influences are under management control (MPs). The third factor is the distant influence of the stock Sto computed by the block Tr defining the equation of y shown in the Figure. Maxtr is a gain (maximum transfer rate) and tabl is a tabular function. It depends on a variable u expressing the fractional error between the current stock Sto and its desired value DesSto. This distant influence is also an MP. Finally, the flow rate is computed by the last equation, supposed to be independent of management and thus an ISC.

Fuzzy internal system controls (ISC) and management policies in TSD.

A full fuzzification of a TSD model could be obtained by applying the extension principle to its differential equations and obtaining fuzzy differential equations [3]. We could also use qualitative simulation methods and restrict ourselves to computing qualitative behaviour [4]. However, this is not what we want. We need to preserve the crisp quantitative aspects of the flow balances expressed in junctions and in stock equations (sum of flows$_{in}$ - sum of flows$_{out}$ = stock derivative). We want also to be able to couple classical crisp formulations with fuzzy rules. In fact, the only model parts we want to fuzzify are some flow-computing rules but their inputs and outputs must stay crisp numbers. This points toward a method which considers the control structures as independent fuzzy controllers in the Sugeno sense taking its inputs as numerical values from the stocks and auxiliary variables and delivering control variables as de-fuzzified crisp values. In Fig.4, we show such a control system for a modified and more general version of the previous example.

Example of rules
if(Blo is large and Sto is small) then (y is very large positive)
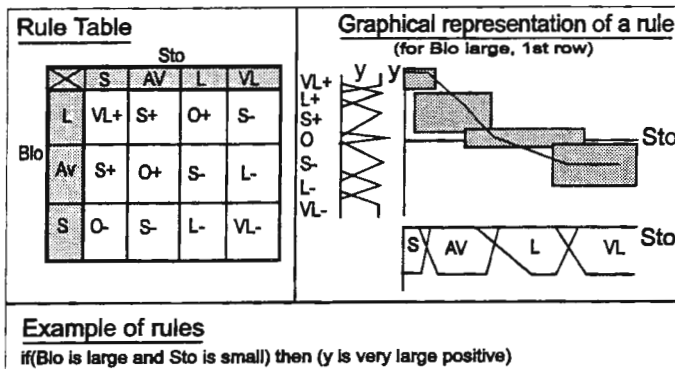
Fig.4: a fuzzy controller implementation of a "common sense policy" deducing the staff transfer rate (positive means toward RIP) and expanding on Fig.3 by using values of Blo and Sto. The fuzzy linguistic values are VL (very large), L (Large), Av (average), S (small), O (zero). The table gives the linguistic rules, an example of which is given below. A graphical representation of a set of four rules describing the first row (Blo = L) of the table is given on the right with an illustration of the fuzzy subsets involved. The four grey patches describing these rules show roughly the inverted sigmoidal shape used in Fig.3. They could be obtained by a fuzzy clustering method from real data showing the actions of human experts in an interactive simulation game.

As long as its number of variables is small or if it is organised hierarchically, a controller like the one above is much easier to formulate than a multi-variable algebraic function. It may also be explained very easily to non-mathematicians. Finally it may be learned from real data gathered by observing human experts playing an interactive simulation game. In our opinion, the inclusion of such Sugeno-like controllers in TSD is a significant step towards real world applications. We will now describe their realisation by trainable neuro-fuzzy systems, which allow us to use both symbolic linguistic and sub-symbolic numerical levels.

Neuro-fuzzy implementation of control rules (management determined and internal)

Fig.5 give the elements of a neuro-fuzzy formulation of the controller of Fig.4. It shows that fuzzy formulations are just parallel implementations of functions similar to those used for the control structures in Fig.2. They have however two important advantages. Being based on a fuzzy semantics, they are designed, understood and explained much more easily than algebraic formulas. Moreover, by placing them in a neural network perspective, we get the possibility to use learning methods and to make them adaptive. This is important in modelling managed systems where classical methods are so often insufficient.

In Fig.5-a, we show a neuron-like implementation of the rule given at the bottom of Fig.4. It shows that each rule is just a generalized artificial neuron with three modifications: the usual weighting formulas are replaced by functions w(x) given by the fuzzy membership functions of the inputs; the summation of activation is replaced by a fuzzy operator (frequently a AND or a t-norm operator) and the squashing function is replaced by the inverse of a simplified membership function of the output variable (Sugeno singleton). Each of the twelve rules of Fig.4 is given by a similar neuron. Since the input fuzzy sets are overlapping, several of these rules fire with different degrees for each set of values of the inputs. The final output of the controller is then computed by a parallel implication method. This is illustrated in Fig.5-b.
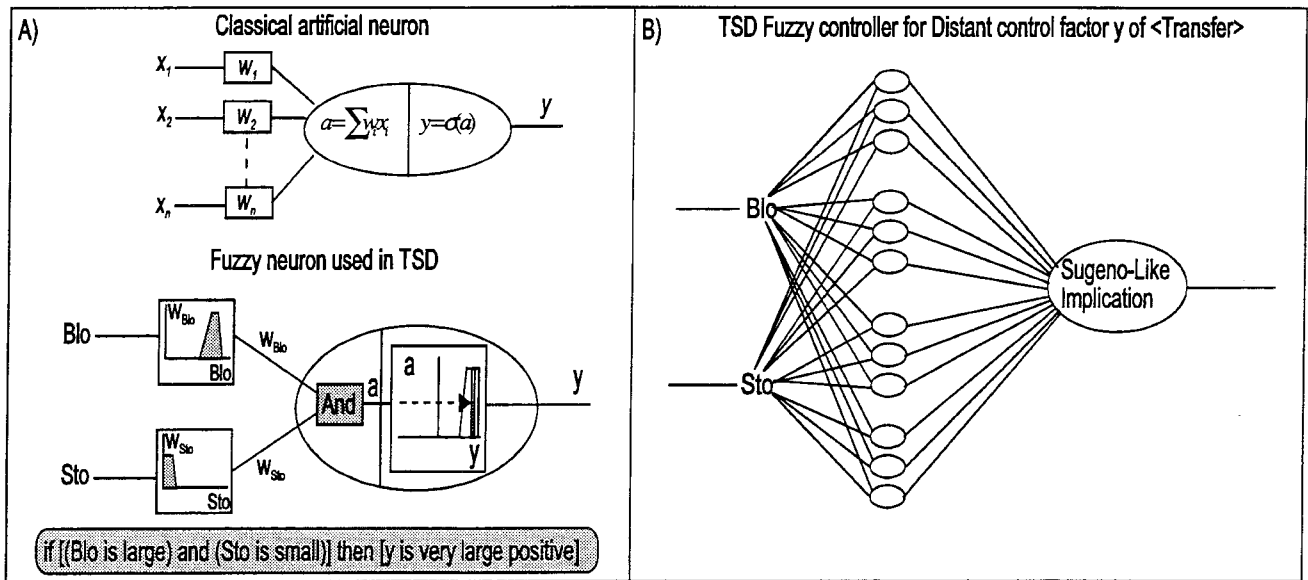
Fig.5: A) A neuro-fuzzy node for the rule of Fig.4 and its comparison with an artificial neuron. Each of the rules of the controller of Fig.4 becomes such a fuzzy neuron. B) The whole controller with al parallel Sugeno-like implication.

From Fig.5, we may draw two conclusions: (1) The architecture of classical neuro-fuzzy networks is much more simple than the one of usual neural networks. This pinpoints toward many generalizations like hidden units, recurrent networks, associative networks and controllers with pre-processed inputs (averaging, smoothing, fading memories). (2) As it is well known, the number of rules (i.e. neurons) increase very rapidly with the number of input variables. It is thus very important to decompose such controllers. For instance, for the transfer process previously discussed, we could consider independent fuzzy systems for each of the three influences (close and distant) and even another fuzzy system for the flow law itself. Obviously, since we have many processes to consider, the number of controllers becomes quite high and fast design methods are a must. We should also consider that each controller has many aspects to optimise. For instance, we could consider incomplete systems in which different sets of input variables are connected to each rule. Then the design of the condition and action predicates becomes an issue. Similarly, the choice of the membership functions and of the number of fuzzy levels may also be optimised. To obtain good policies is obviously an important aspect of TSD. In managed systems, it is often the goal of the whole exercise. For this reason, we will describe now how we use genetic methods to design these controllers.

Genetic design of efficient policies

Besides understanding and education, an important goal of modelling managed systems is to test-drive our ideas on efficient management. Usually, experimentation and expertise result in a reasonable set of MPs for the various process rates and packing coefficients. However, much remain to be done:
- The mathematical solutions obtained may be made more efficient by optimising their parameters.
- Fuzzy control systems may be improved by adjusting their number of levels and membership functions.
- More ambitiously, we may also try to find not only different parameters but also completely new solutions by modifying the structure of the existing equations or fuzzy rules and even inventing new ones.

Usually, users have explored only a very small subset of all these possibilities. They are indeed limited by the time available and by their own inventiveness. To help them overcome these two limitations, we are now developing methods and tools based on genetic algorithms and programming [5,6]. After preliminary experimentation, we have identified the major problems confronting an evolutionary approach to policy optimisation in TSD:

- Our problems are frequently high dimensional (large genes and population) and the evaluation of a generation of policies by simulation with several test cases may take a very long time.
- It is difficult to formulate accurately the efficiency functions of our problems. They are often multi-criteria and frequently all we have is a fuzzy set of evaluation rules.
- Another problem is that very often users may tell us what they like but they are not able to make their evaluation procedure totally explicit.

To address these problems, we have currently developed our GA applications along the following lines:

Parallel computing: The scale problem has been solved by using a parallel LAN-based GA [7]. We distribute the generation to be evaluated to the machines in the network which then simulate their test cases in parallel. The network transfer time is usually negligible. It is currently possible to use very cheap dedicated PCs stripped of all the unnecessary hardware. This solution may become attractive for instance in a consultancy activity where many large TSD models and many policies have to be optimised. In the same line, it is worthwhile to remark that the fuzzification introduced here is certainly very useful from a linguistic point of view but, once a fuzzy controller is implemented, it computes just an ordinary algebraic function like any other one. Since most hypotheses on which a TSD model is based may be slightly changed, we still have the need for testing various hypotheses and looking for robust conclusions. This may be done by stochastic sampling with chosen probability distribution for parameters. This again involves many simulations and therefore the need for the dedicated Lan-based parallel implementation described here.

Multi-criteria Optimisation: We have developed a GA using a ranking evaluation of the members of a population against several criteria and niche formation mechanisms. We use then a Pareto evaluation and the population evolves by distributing itself along the boundary of the feasibility region.

Hybridization and breeding: Since our users have at best incomplete evaluation functions, we must give them a way to introduce their own, hopefully consistent biases into the process. We do it by hybridizing the classical GA with other methods of optimisation (gradient, ad-hoc) and even with an operator which is simply to introduce user-given modifications in the population. This hybridization is done by providing in addition to the roulette wheel selecting the members of a family, a second roulette wheel selecting the operators randomly with constant or adaptive probabilities. Finally, we allow the users to be part of the evaluation process by introducing their own subjective evaluation as one of the criteria.

Inventing new rules: We are now considering the extension of these methods to genetic programming in order to work directly at the syntactic level of the policies (like in a Holland's classifier system).

Conclusion

Taken together, the development of TSD, the fuzzification of its policies and their optimisation by genetic methods are important extensions to Forrester System Dynamics. Our feeling is that our tool gives us a much better tool to address real world problems in managed system simulation.

References

[1] LeFèvre J.: Application of kinetic bond graphs in corporate and production management systems, pp. 226 – 232, in "Commande des systèmes industriels", Belfort AFCET – IMACS Symp., 1997.
[2] Coyle R.G.: System dynamics modelling, Chapman and Hall, London, 1996
[3] Klir G. and Yuan B.: Fuzzy sets and fuzzy logic,Prentice Hall, New York, 1995
[4] Faltings B. and Struss P.: Recent advances in qualitative physics, MIT Press, Cambridge USA, 1992
[5] Pedricz W.: Fuzzy evolutionary computation, Kluwer, 1997
[6] Koza J.R.: Genetic programming, Bradford MIT Press, Cambridge Usa, 1994
[7] Weller P.: A Lan-based parallel GA applied to neural network design, MSc Thesis, City U.London, 1994

IMPLEMENTATION OF FUNCTIONAL SIMULATION AND ANIMATED VISUALISATION ON HETEROGENEOUS PARALLEL ARCHITECTURES

J.Habibi. M.N. Alpdemir and R.N.Zobel
Department of Computer Science
University of Manchester
Oxford Road Manchester M13 9PL, U.K
Tel: +44 161 275 6210/6189
Fax: +44 161 275 6204
email: habibij@cs.man.ac.uk, alpdemim@cs.man.ac.uk, rzobel@cs.man.ac.uk

ABSTRACT

This paper considers the use of simulation and visualisation in the concept of virtual design and manufacture, with the aims of shorter time to market at lower cost and risk. In particular, integrating of existing tools and the development of an integration structure are emphasised. Specific aspects addressed include functional simulations and geometric models, associated with animation and visualisation and the development of the distributed simulation. A theoretical design is proposed to integrate simulation modules and existing visualisation tools in a heterogeneous parallel computing network. The aim is to create a dynamic interactive system where users can display and manipulate remote data in a transparent fashion. The use of distributed interactive simulation is also considered.

This paper expands on the design of functional simulations employed to support the design process to animate the visualisation of the product functionality by linking them together over the network and covers the implementation of the package in detail.

INTRODUCTION

Many simulation tools are being used in a wide range of applications in both manufacturing and process industries [2][8]. These tend to be used in isolation and are frequently not integrated with other tools used in specification, design, prototyping, testing, evaluation and maintenance. The costs associated with such systems make it critical to be able to forecast system behavior before an actual prototype is constructed. Modeling and simulation techniqes allow the complexities of actual systems to be prototyped in the form of virtual prototypes with significantly more acuuracy than traditional simulation methods. Although the simulation times often grow beyond practical limits, distributed simulation methods have the potential to reduce these times in a scalable fashion. Further, the essential use of visualisation, simulation models and animation for design engineers is also not well integrated.

Visualisation of the appearance and functionality of a product as it develops is important for both the manufacturer and the customer.

Providing an animated visualisation of the functionality of a product in a distributed networked system is therefore highly desirable. The following discussion relates to a new substantial project aimed at improving the situation, by integrating a simulation with the animated visualisation over the network.

As shown in previous work [6] A dot-matrix printer is used as a vehicle for development of the concept. It is appropriate because it combines discrete event and continuous simulation elements with functionality which can be visualised in an animated way at several levels of complexity. Simulations are based on C++ for discrete event

subsystems, with ESL [7] used for the continuous parts and VISTA [5] used for animating the geometric model. Low level simulations of generic functions are carried out and interfaced to the visualisation system.

One of the early realisation in this research was the enormity of the scope of implementation. This activity was hampered by the delay of softwares such as CAD and animation which was the original choice of Geometric model. In some sense this was fortunate since investigating alternative mechanisms led to the realisation of the potential of VISTA and related developments. Another major hindrance has been the lack of interface support for the C++ version of ESL with VISTA. Nevertheless, as much of the design was implemented as possible to illustrate the principles as set out in the following sections:

EXAMPLE SYSTEM

The application used to demonstrate our investigation is a simulation of the dot matrix printer as an example of a complex system. The overall system is divided into geometric models and functional models which further divide into functional subsystems, until a uniform description method for the behaviour of each subsystem can be found. Usually, each subsystem comprises only one physical principle. For the example of the dot matrix printer the functional decomposition yields the following two subsystems: The mechanical system (continuous time model) and the printer control system (discrete event base model).

THE MODEL FRAME

Figure 1 shows the proposed structure of the complete simulation of integrating the functional model and the geometric model for a dot-matrix printer in the distributed environment.

The function of the program modules are described below:

The functional model runs a simulation of a print head entity, driven by a notched belt drive across the paper. As the print head moves in the functional model, its essential state information is propagated over time to the geometric model via a stream of entity state PDUs. The geometric model monitors this stream and updates the internal state information concerning the print head. The geometric model also has an image generator as a component to visually display the simulated
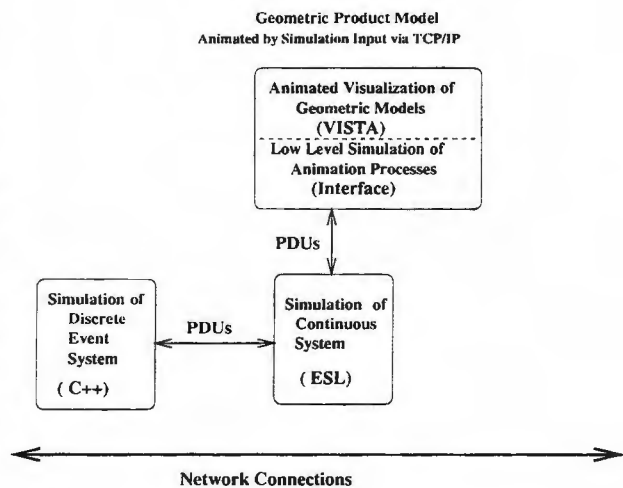
Figure 1: Integrating Networked Functional Simulation and Animated Visualisation

environment and entities. Note that a geometric model may be simulating a set of its own entities, whose state information would be propagated back to the functional model via the same mechanisms.

GENERAL IMPLEMENTATION STRATEGY

The simulation program is organised as three separate processes (possibly operating on three separate computers). The processes are: simulation of the mechanical part, simulation of the digital part (printer control process), and the geometric model visualisation. The simulation system uses ESL (European Space Agency Simulation Language), C++ and VISTA (Visualisation and Technical Animation software) which have been chosen to implement the three processes, respectively.

One of the aims of this work is to show that generic submodels developed with different approaches (i.e. continuous, discrete, geometric) and implemented in different languages (i.e. ESL, C++ and VISTA) can be linked together to form a coherent model of a large system. ESL is a continuous system simulation language providing the user with all of the facilities required to perform simulations. ESL is an advanced simulation system that allows calling of user routines (implemented in C, FORTRAN and C++). The mechanical system of a dot-matrix printer was im-

plemented with ESL. C++ is used for the discrete event simulation. VISTA is a visualisation software system designed towards the generation of animated sequences of industrial and technical activities.

Using I-DEAS as the modelling package

I-DEAS is a solids-based, simulation driven software package for mechanical engineering [1]. The package is highly capable and is intended to be used for the integration of the design, analysis, drafting and manufacturing into each phase of product development.

There are several reasons to be mentioned for the selection of this package as a geometrical modelling environment:

1. It is an advanced package allowing the design of complex products applying techniques like constructive solid geometry, feature definition, assembly modelling and entity management.

2. Naming the individual components(objects) in an assembly is possible, which facilitates an object oriented approach to the animation of the model. Individual components can be referenced by their names given during the model generation.

3. It is a well-known widely-used modelling package used in both academia and industry, which means that any succesful software project using I-DEAS as the geometric modelling tool will find acceptance more easily.

4. The animation package chosen for our project (i.e. VISTA) accepts its geometric models in the Universal File Format which is created by the I-DEAS package.

Integrating I-DEAS with VISTA

There is a geometrical interface between the powerful solid modelling progam IDEAS (from SDRC) and VISTA. To import geometric models into the VISTA environments we have to convert models produced by IDEAS package into VISTA.

The components of the dot matrix printer were first generated individually as solid geometry objects with attached names. Then, they were assembled to compose the printer using assembly

modelling feature of I-DEAS package. An assembled model of the printer can be seen in Figure 2. This model was then saved in the universal file format and transfered to the VISTA environment. The I-DEAS package used runs on HP workstations and VISTA runs on Silicon Graphics workstations. The Universal file format is a text file generated by the IDEAS package to facilitate the transfer of the generated model to different graphics environments.
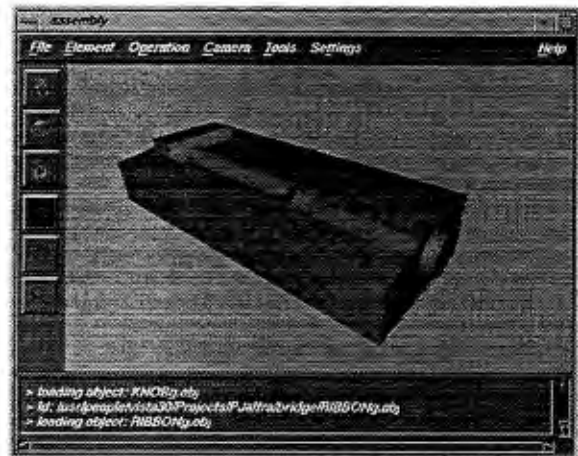
Figure 2: An Operational View of the VISTA Environment

Implementation in ESL and VISTA

This part describes the implementation of interface module within a Client-Server architecture, based on the TCP-IP communication protocol.

The Environment for the Rendering and animation of the simulated model

While choosing the environment for the animated visualization of the simulation models, three different alternatives were considered: PHIGS [4], OPENGL [9] and VISTA. Two of these, PHIGS and OPENGL, are graphics libraries for defining and manipulating three dimensional graphical objects. They are low level libraries in the sense that substantial amount of programming is required to build an environment with capabilities such as geometric object database management, high qual-

ity object rendering, run-time interfacing to simulators for object animation, run-time session management etc.

However, they are extremely flexible, in the sense that they can be used to create a graphics sub-system for visualization and animation of the objects in a virtual environment to suit any set of operational specifications.

VISTA, on the other hand, is a high level graphics package, built using SGI's GL library. It accepts geometric objects generated by an advanced CAD package (I-DEAS), has considerable amount of tools to modify characteristics of objects (such as orientation, coloring and material properties, texture mapping), to generate assemblies of objects form the object repository, and has also a well defined run time interface to programs (simulators) written in ESL. It is also possible to create off line animations by using *anima* (the animation tool) and record them as video files.

It, however, has a tight framework and it is intended to be used within a limited range of applications.

For the work defined in this paper VISTA was chosen as the animation environment, for the reasons listed below:

1. It is flexible enough to demonstrate the concept of linking hybrid simulations to a visualization system to animate the functionality of a product.

2. There is an existing run-time interface between ESL (the language used to simulate the functionality of the product) and VISTA which enables ESL programs to talk to VISTA environment.

3. The main purpose of this study is not to design and implement a graphics system, but to demonstrate results of simulating object functionalities in a distributed environment. Therefore it is not feasible to use low level graphics libraries since substantial amount of graphics coding is required if they were to be used.

The VISTA Environment

The operation of the VISTA environment is shown in Figure 3. The geometric models are generated using an advanced CAD package, I-DEAS, and exported in Universal file format, as explained above.
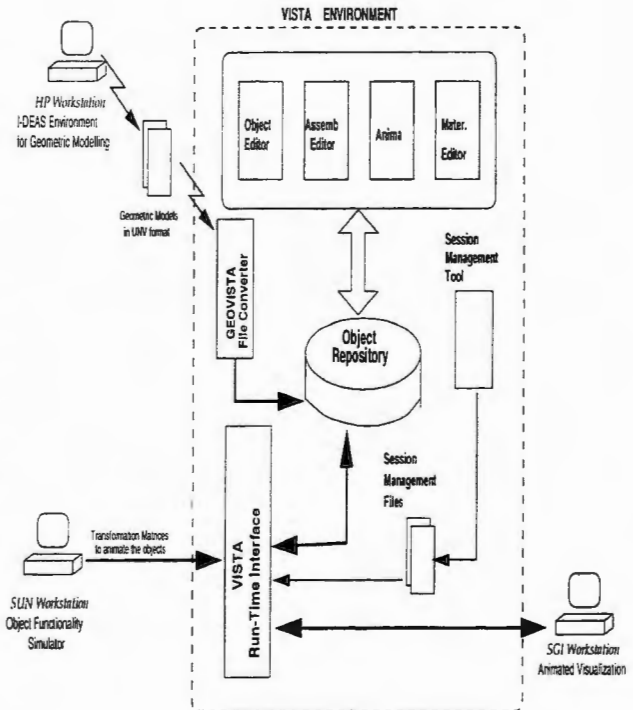
Figure 3: An Operational View of the VISTA Environment

They are then converted to VISTA *object* and *assembly* formats by a software module called *geovista*, and put into the object repository. The user can then edit these objects to change their orientation, color and material properties, create texture mapping, and define animation sequences for off-line animation, by using the tools provided by the VISTA package. The object editor tool is used to view the object in wire frame or shaded modes. The assembly tool can be used to create assemblies of objects by transforming the objects to desired positions relative to each other. The material editor tool is used to determine the material properties of the objects and to map textures to the surfaces of the objects. Using the animation tool it is possible to attach objects and lights to each other, to form a hierarchy of objects. These facilitate dependent motion of objects in a hierarchical manner. The run-time interface module of VISTA is for on-line simulation. It adopts the client-server philosophy and uses TCP-IP protocol. It is this particular interface which gives VISTA its capability to be used as a visualization module of an interactive simulation environment. It supports the idea that the geometric represen-

10

tation of a virtual product can be viewed with its functionality attached to it, provided that the functionality of the object is simulated in the form of some external code and the output of this code is fed to the VISTA environment in the appropriate format.

The Communication between ESL programs and VISTA
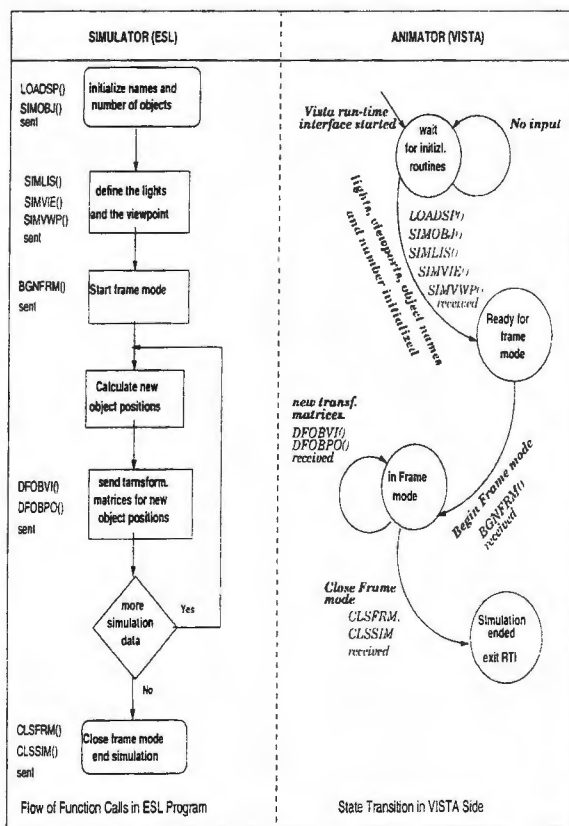
Figure 4: An operational wiev during communication between ESL programs and VISTA

The run-time interface module is activated from the Vista workstation. A communication channel is then opened between vista and simulators running on a remote workstation (SUN workstations in our case). The run-time interface then sits and waits for function calls to initialize the visualization environment parameters, such as the light positions, view ports, camera positions, initial object positions etc. It then expects to receive a set of transformation matrices which redefine new ob-

ject positions over a period of time to create the desired animation.

Figure 4. shows a typical scenario in which an ESL program invokes the run-time interface functions to prepare VISTA for displaying animated objects. It then enters a loop in which calculated transformation matrices, defining new object positions in 3D space, are sent to VISTA. The name of the functions are given for each phase. Most of these functions expect several parameters. They are a set of external FORTRAN subroutines which are called from within an ESL program. The routines are divided into three groups:

(1) initialisation and state change routines, which switch VISTA between its different protocol states, (2) simulation definition routines, used to define the simulation environment, and (3) frame definition routines, used to describe object movement between frames.

In order to build the executable code, a FORTRAN program is produced from the ESL source code and during compilation it is linked to the interface routines. In general each FORTRAN interface subroutine maps to a single VISTA command. Detailed explanation about run-time interface functions is given in [3].

Real-time Synchronization

A simple effective mechanism is used for monitoring real-time performance, and ensuring real-time synchronization. A simulation is only capable of real-time operation if the simulation of a time period, can be accomplished in a real-time period which is shorter. If this criteria is achieved, it is necessary to slow the simulation computation so the simulation time is synchronised with real time. This is achieved by the simulation waiting after each communication interval until real-time becomes equal to simulated time. A single external ESL routin "synchx" which is available in the ESL run-time library is used to achieve these objectives.

Linking the continuous time simulation and discrete event simulation

In general, if two simulation systems are to be connected by a message passing mechanism to work together, two different pieces of software are required for the communication between them. These are:

(1) The common interface which defines the messages (i.e. PDUs) for communication between the simulators,

(2) The communication library which is a collection of functions that are called by the particular simulator. This library is unique to each different simulation system (e.g. one library for ESL and another for C++), but in general the same library can be used for communication with any other simulator which reduces the implementation effort remarkably when many different connections should be established. Figure 5 shows a graph of two connected simulation systems, in the same way it is possible to connect more than two different simulators. It is also possible to achieve a simple parallelisation by connecting two or more models that are simulated on different processors.
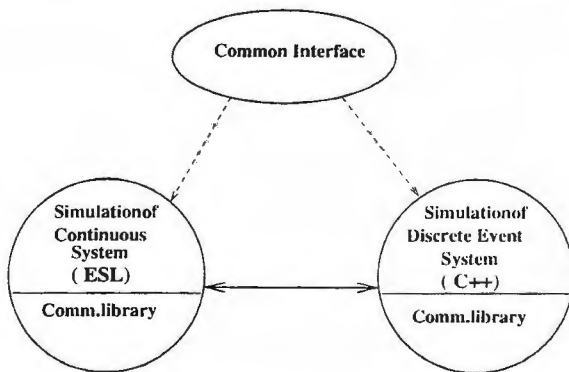
Figure 5: Integration of the Two Simulators

The continuous simulator is controlled via an ESL main program which waits for the incoming messages and calls the appropriate ESL submodel according to the type of the message. It accesses the network through a set of C functions which currently make use of the PVM facilities for message passing. Switching to a different networking tool can be achieved by changing these functions only.

The interested reader is referred to[6].

RESULTS

One of the interesting outcomes is that hardware can be relatively easily interfaced but software can prove a lot more intractable. For example, the initial choice was to control the continuous part simulation with a C++ program. The program would accept messages from the discrete part and run the necessary ESL submodels. Running ESL submodels is achieved through an ESL segment; and the C++ version of ESL enables calling segments from a C++ program. However, problems occurred when we adding the VISTA animation tool to the environment.

VISTA has an ESL interface which has been written in FORTRAN. This FORTRAN code enables VISTA to run as an ESL remote segment. Unfortunately it is not compatible with the C++ version of ESL. When attempting to use both libraries, error messages occurred due to multiple definition of some objects. It is also not possible to call a segment of the FORTRAN version of ESL from a C++ program.

The second attempt was to control the continuous part's simulation directly with ESL. An ESL main program waits for the incoming messages and calls the appropriate submodel according to the type of the message. The FORTRAN version of ESL was used, so that VISTA can be utilised. The main ESL program accesses the network through a set of C functions. This organisation of the continuous simulation part was successful.

It has been observed that using the ESL-VISTA interface routines, a reasonably realistic representation of the functionality simulation can be achieved for the level of complexity we have adopted. The printer example consists of seven main parts and can be classified to be a simple model. In circumstances where transforming one object (part) at a time is sufficient to create the desired animation effect, which is the case in our study, ESL-VISTA interface performs quite efficiently. It has been proposed, however, that in cases where genuine simultaneous (and/or independent) animation of many different objects is required a different approach must be devised. In the printer example, sequential transformations are generated to create stepwise rotations or displacements along different axes where object dependecy relations and the hierarchical structure of the objects are taken into account. This has created the smoothness required for the realistic animation.

CONCLUSIONS

This paper has presented an implementation of a system for integrating the functional simulation and geometric model of dot-matrix printer motion for 3D visual simulation. Distributed simulations are shown to reduce the simulation time

required for multicomputer configuration if each part of model implemented in different languages and with differing approaches.

Although the concept was introduced using an example from the area of dot-matrix printer simulation, it is quite general and our current work is also related to its usage in distributed simulation of a space craft system.

The underlying techniques employed are straightforward and build upon several well-known software systems and capabilities. However, practical realisation brings additional problems.

References

[1] *The Introduction to I-DEAS, User Manual* . SDRC, Milford, Ohio, 1990.

[2] Klous Braun and Philipes Solot. Factory Design and Production Scheduling Using Object Oriented Simulation in the Chemical Industry. In A. Verbraeck and E. J.H.Kerckhoffs, editors, *Proc.European Simulation Symposium ESS 93*, pages 204–211, Delft NL, Oct. 1993. The Society for Computer Simulation International, SCS.

[3] J. C. G-C. Chamorro. *RELAY VISTA/ESL*. ESTEC, Noordwijk, the Netherlands, 1994.

[4] Kasper Joseph E. and Arns David. *Graphics Programming with PHIGS and PHIGS PLUS*. ADDISON-WESLEY PUBLISHING COMPANY, NEW YORK, 1993.

[5] CASA Division Espace. *VISTA Executive Summary*. 1995.

[6] Kucuck B. Habibi J. and Zobel R. N. INTEGRATING SIMULATORS WITH THE DIS CONCEPT AND PVM . In *Proc. The Third Annual Computer Conference of Computer Society of Iran*, Uni. of Science and Technology, Tehran, Islamic Republic of Iran, Dec., 22-24 1997. CSI, CSICC'97.

[7] Pearce J. Hay J. and Crosbie R. *The ESL User Manual*. iSiM Simulation, Salford Univ. Business Service Ltd, 1992.

[8] Vlatka Hlupic. Sofware For Manufacturing Simulation: A User's Survey. In A. Verbraeck and E. J.H.Kerckhoffs, editors, *Proc. European Simulation Symposium ESS 93*, pages

109–114, Delft NL, Oct. 1993. The Society for Computer Simulation International, SCS.

[9] J. Neider M. Woo and T. Davis, editors. *Open GL Programming Guide, Second Edition*. Addison-Wesley, 1997.

About The Authors

Jafar Habibi graduated with a B.Sc. degree from Tehran School of Computer Sciences and an M.Sc. degree from Tarbiyate Modares University, Iran. He subsequently joined as a lecturer to the academic staff of Sharif University of Technology Department of Computer Engineering, Iran for five years. He is currently carrying out research work for his PhD as a postgraduate student in Computer Science, University of Manchester, U.K.

M. Nedim Alpdemir has a BSc degree in Computer Engineering from the Middle East Technical University (METU), received in 1990, and an MSc degree in Advanced Computer Science from the University of Manchester, received in 1996. Since then, he has been carrying out research for PhD in Manchester. His main research interests are visualization and computer graphics in the context of distributed simulation.

An Electrical Engineering graduate of London University in 1963, Richard Zobels' first simulation experience was in 1962-66 at Sperry Gyroscope. His Manchester PhD., (1970) was on hybrid analog digital computing. As Lecturer and Senior Lecturer his research interests include signal processing, simulation in concurrent engineering, and synthetic environments for rapid product prototyping and manufacturing. He is Past Chairman of UKSS, Secretary of EUROSIM, and was until 1994 a Director of SCS.

CORBA in Simulation Tasks

D. Adamski, M. Hiller
Fachgebiet Mechatronik, Fachbereich Maschinenbau
Gerhard-Mercator-Universität - Gesamthochschule Duisburg
Lotharstr. 1, D-47057 Duisburg
adamski@mechatronik.uni-duisburg.de

Simulation programs are generally implemented for special hardware platforms and operating systems, but developers of simulation models, who are the users of these programs, often work on different platforms or the tools used are licensed only for special computers. The goal of this paper is to describe another way of implementing simulation software on heterogeneous computer networks. Particular components of a simulation program are provided as platform independent services, and only those services which are necessary for the current simulation task are combined to a program. This is realised with the help of CORBA.

1 Introduction

Today's information technology world is a heterogeneous one. With a mixture of mainframes, UNIX workstations and PCs information is shared over large networks with many different operating systems and network protocols. Furthermore, there is currently no way for all software developers to use the same programming language or the same hardware platform. One solution for this situation is the development of standards for communication across both system and language boundaries. Thus it will be possible to distribute applications to different platforms, obtaining the best developing environment or the best performance for each program task. This means that, for example, with a simulation program, the user interface may be on a PC, the calculation carried out on a high speed workstation and the on-line animation of the results presented on a graphic workstation.

Generally the market for simulation software provides a large number of multipurpose and special purpose simulation programs with high functionality. Nevertheless an additionally large number of self-written programs are used for simulation tasks. Often a lot of time is spent implementing service routines and peripheral software devices such as graphical output to the display or the printer, analysis tools or messaging systems. To realise this in the times of structured programming large software libraries had to be included in the program. Later on, after the introduction of object-oriented programming, terms like *reuse* and *inheritance* became more and more important. Thus the development time for software decreased, but each program is still growing in size with every additional function it provides.

The costs of developing a powerful simulation environment increase with expectations of ease of use and functionality. The time where it was enough that a program contains a few specialised functions, reachable via a command-line interface are gone. The expectations of the users have changed considerably and so today's software systems must have graphical user interfaces, integrated database capabilities and must be able to cope with complex data structures like graphics or multimedia objects [MOWBRAY95]. The large number of requirements leads to huge and incomprehensible monolithic programs with increasing maintenance costs. The example of modern text editing programs shows the consequences of this strategy. These programs try to provide nearly every possible function and often overcharge the user, who normally wants to use only some particular features. Another approach is to request only those capabilities which are necessary for a special task. This requires the partition of the total functionality into single functionalities concentrated in components and an appropriate mechanism, which is able to bind this requested functions to a suitable program. The basis has to be a client/server system, where the server provides a service in form of a program component requested by the client. The distribution of this service on a network enables several users to demand a particular service, although they are using programs with different compositions on different hardware platforms. This evolution in software development is shown in figure 1.1.
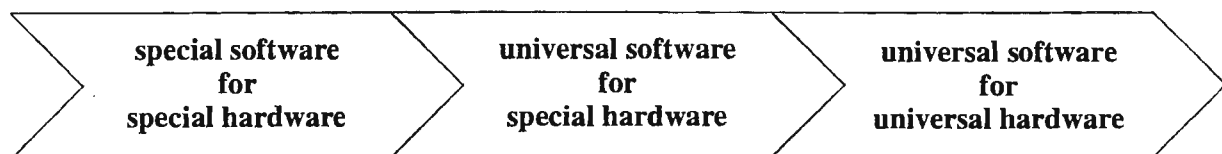
figure 1.1: Evolution of Software Development

A necessity for such a system is a comprehensive catalogue of valuable components, thus a combination of a complete and powerful simulation environment will be possible. To take full advantage of the distributed system this catalogue of components has to be reachable from different hardware platforms. How this can be realised is discussed in the next sections.

2 Distributed Systems

Essential for distributed systems is to overcome the hardware and operating system borders. The backbone of this idea is the *middleware*. This is the interface between applications written in a particular programming language and the operating system appertaining to a certain hardware platform. So that the developer is free from dependencies on communication protocols, operating systems and hardware platforms. He must only move information from one program to another, placed somewhere inside a network, not regarding anything platform specific like, for example, byte order. Generally there are several different methods for distributing objects [TALARIAN97]:

- *Message Oriented Middleware* (MOM)
- *Object Request Broker* (ORB)
- *Remote Procedure Call* (RPC)
- *Transaction Protocol Monitors* (TP Monitors)

In this paper the Object Request Broker is discussed, where the question of whether the other program runs in the same process or on the same machine or on the same network or on another network across the internet, should play no role, because the mechanism for accessing local objects is the same as the one for far ones. Thus our program acts like a client requesting a service from a server which location, operating system or hardware platform is unknown and it is not necessary to know anything about that. The way of writing software is changing. To what, will be discussed in the next sections.

2.1 Distributed Objects

The essence of an object is, that it encapsulate its code and data, has a name and has an interface describing its capabilities. Its functions, described in the interface, can be invoked by an other object which knows about its interface. The classical object which is used in programming languages like C++ is only known by the compiler and within the capabilities of the programming language it only can communicate with objects in the same program. From outside it is invisible and unreachable. Moreover, the objects in a program have to be compatible with one another, which is self-evident if they are written in the same programming language and running on the same machine. Clearly, this is a completely other case if the objects are distributed and the application runs parallel on different machines with different hardware and software platforms. Those objects have to be binary components, accessible from other objects with method invocation [ORFALI97]. Thus a way of invoking its methods has to be described in an interface. The main idea is that a programmer can choose components from several vendors and connect them for building new applications as easily as the well known 'plug-and-play'-standard for arranging hardware components. A communication standard which can do this, is described in the following subsection.

2.2 CORBA

Since the non-profit industry consortium *Object Management Group* (OMG) is founded in 1989[1] the number of companies which try to formulate an open standard for a distributed object architecture increases up to 700. The members of the OMG are hard- and software vendors, but also airlines and other large companies using computer networks. They had found a way to formulate a mighty standard of high quality and widespread acceptance. The two computing paradigms *object orientation* and *distributed computing* are brought together in the *Common Object Request Broker Architecture* (CORBA). CORBA defines the ingredients, the behaviour, and the interfaces of a system based on the later described Object Request Bus. The OMG does not distribute CORBA as a product, it tries to certify the compliance of the commercial products with the agrees-upon standards [TAYLOR93]. Thus the software vendors can write the implementations of CORBA in their own way, they only have to fulfil the standard. Thus the user can choose between several implementations considering different aspects.

2.2.1 The Software Bus - ORB

The backbone of CORBA, the mechanism for communication between the clients and servers, is called the *Object Request Broker* (ORB). All basic object interaction capabilities are provided by the ORB. This is the middleware which allows a client to invoke a method on a server, which may be placed on the same computer or one across a network

---

[1] The 11 foundation members of the OMG show the band-width of the idea of distributed computing: 3COM, American Airlines, Canon, Data General, Gold Hill, Hewlett Packard, Philips, Prime, Softswitch, Sun, and Unisys [BENNATAN95].

[ORFALI97]. Figure 2.1 shows which interfaces between the application and the ORB are available. The ORB core can be seen as the basic interface to the transportation layer of the network protocol. The IDL Stubs and Skeletons are used to define the interfaces between the objects, written in a particular programming language and the ORB. They contain information on how an operation can be invoked and which parameters are necessary. How they are generated automatically is discussed in subsection 2.2.2.
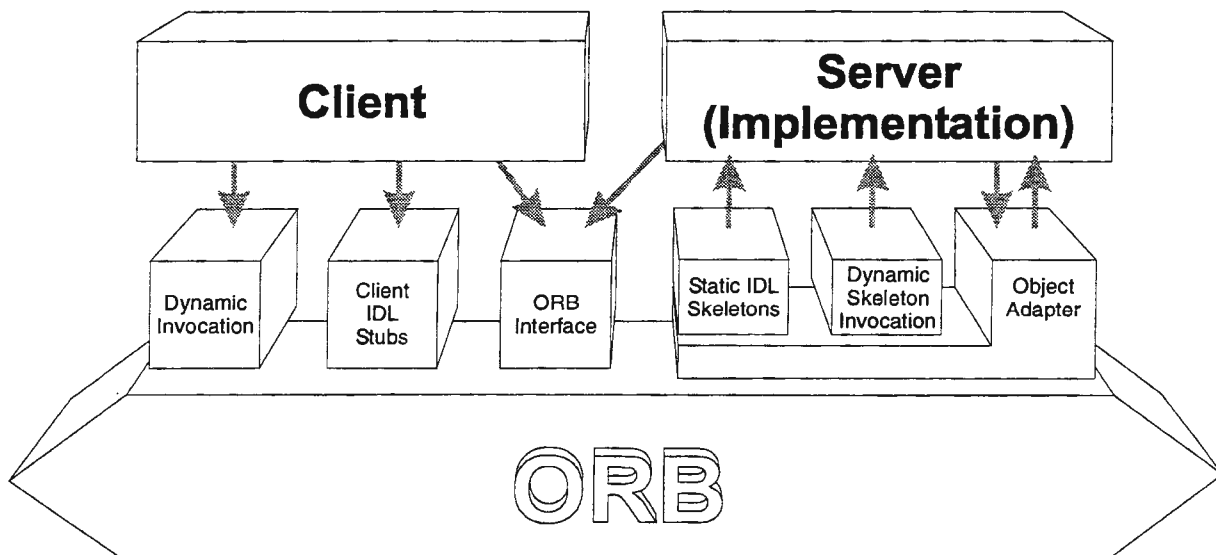
figure 2.1: The Structure of the Object Request Broker Interfaces [OMG95].

The client may invoke any available method on the server without knowing its location or the programming language used. The ORB finds the correct implementation across language or system borders. The advantage for the developer is that he does not have to be concerned with network protocols or byte ordering when he transfers data between different platforms.

The ORB is like a central software bus, where any application or component is plugged in providing or requesting services at runtime. The ORB takes the request from the client and transmit it with the parameters given to the server. There, the requested method is invoked and the result is returned via the ORB to the client. Although we are speaking about clients and servers the classification of client/server systems has changed, since every component may play the role of a client and a server at the same time. A client requests a service, which is provided by a server, but it is also possible that the client provides services, which are requested by other clients. The object which is the originator of an object invocation is a client. If it is the recipient it is a server [MOWBRAY95]. Thus it is not necessary to handle objects in a special way to become a client or a server.

For including existing applications or components even if they are not object-oriented one can write so called *wrappers* or *adapters*. These are interfaces which use an Object Request Broker to communicate, for example, with a self-written application. Thus the whole treasure of existing software can be used, assuming that programmable interfaces exist. The interfaces could be written in IDL, a standard which is described in the following subsection.

2.2.2 Interface Description with IDL

One of the advantages of CORBA is that it is operating system and platform independent. Furthermore, there exist several interfaces to common programming languages like Ada, C, C++, Smalltalk or Java. This independence requires a neutral description language with the possibility to map a description in one of these languages to one of the chosen platforms. This description language is called *Interface Definition Language* (IDL), which is a subset of C++ with additional keywords. IDL however, is only declarative, which means that no details about the implementation are included. Within the interface specification the types of objects are defined. So, an interface consists of a set of named operations and their parameters, following the idea that an object owns data, called *attributes*, and provides services, called *methods* or *operations*. Inheritance of methods and attributes from interface descriptions is supported. So, in an interface a particular object implementation publishes its operations and how they may be invoked. This information is needed by potential clients [OMG95].

16
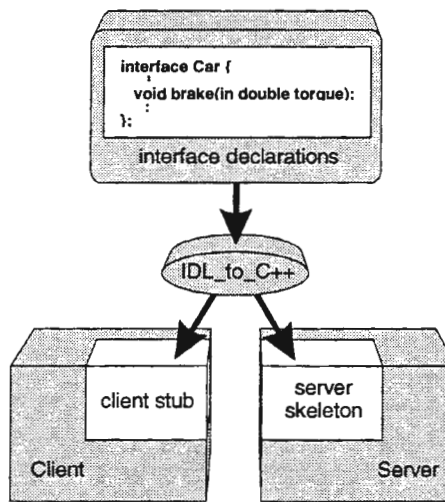
figure 2.2: Mapping from IDL to C++

With an IDL compiler the description of the interface is mapped to a specific programming language like C++. The client stubs and the server skeletons are generated automatically (figure 2.2). The client program is linked with the stub file, which looks like a local function call to the client, although it is an interface to the ORB that performs marshalling to encode and decode the operation's parameters in a suitable format [MOWBRAY95]. The skeleton communicates with the server implementation interface. The implementation of the server has to be done by the programmer by hand. The existence of the ORB is assumed and it is used as a part of the operating system.

3 Distributed Simulation

Most programs consist of one monolithic block. Client/server technology has divided it in two, still monolithic, halves. The idea of distributing objects is to split the program into a lot of single, self-managing components which can be assembled '*in an infinite number of lego-like arrangements*' [ORFALI96]. The main idea is to insert only those components into the simulation program which are required for the specific task. Thus one can choose the integrator, the input and output facilities or the analysis tools as needed. The goal is to develop single components which may be reused for several different tasks. What advantages are given by distributing a simulation environment? Before this question can be answered, we have to settle the question: What are the components of a simulation environment?

3.1 Components of Simulation Software

First, we have to distinguish between the numerical, modelling and peripheral parts. The numerical part normally consists of several types of integrators, iterators, methods for calculating the eigenvalues or natural responses, event handling, and so on, but not every component is used at the same time. Thus it makes sense to minimise the numerical part, inserting only the components actually required. The structure of the model is strongly dependent of the particular simulation task, and is normally assembled from several modules with special functionalities. Whether it is possible to distribute the simulation model depends on its structure. If it is possible to divide the model in several modules which can be calculated separately, it is an advantage to distribute these modules on different machines on a network, as long as the benefit in calculation time is greater than the effort required for data transfer.

The part which is discussed in this paper is the peripheral containing, for example, the graphical user interface, help and documentation facilities, analysis tools, animation capabilities, data and model interfaces to other programs, pre- or post-processors, and so on. The idea is to establish interfaces for each function, thus making them available to different programs. For this an extremely general interface description has to be found. With the possibilities of inheritance, special interfaces for special cases could be defined later. An example, with the combination of a simulator and an animation tool, realised at the Department of Mechatronics, is described next.

3.2 Example

Over the last few years the Department of Mechatronics has been developing the simulation environment FASIM_C++ for three-dimensional vehicle dynamics simulation [ADAMSKI97]. In co-operation with the German automotive component supplier Robert Bosch GmbH in Stuttgart this tool has been used for the development of automotive controllers, like anti-lock braking systems, traction control systems and vehicle dynamics control. The environment contains a

modular vehicle model bound with the numerical components, is provided for HP-UX and MICROSOFT WINDOWS NT systems. Additional programs have been included, a graphical user interface, WINFAS, and an online-animation program, AMS, for MICROSOFT WINDOWS NT (FIGURE 3.1). Furthermore, there exist interfaces to MICROSOFT EXCEL and MATLAB/SIMULINK.

FASIM C++

WinFas   MS Excel   Matlab   AMS
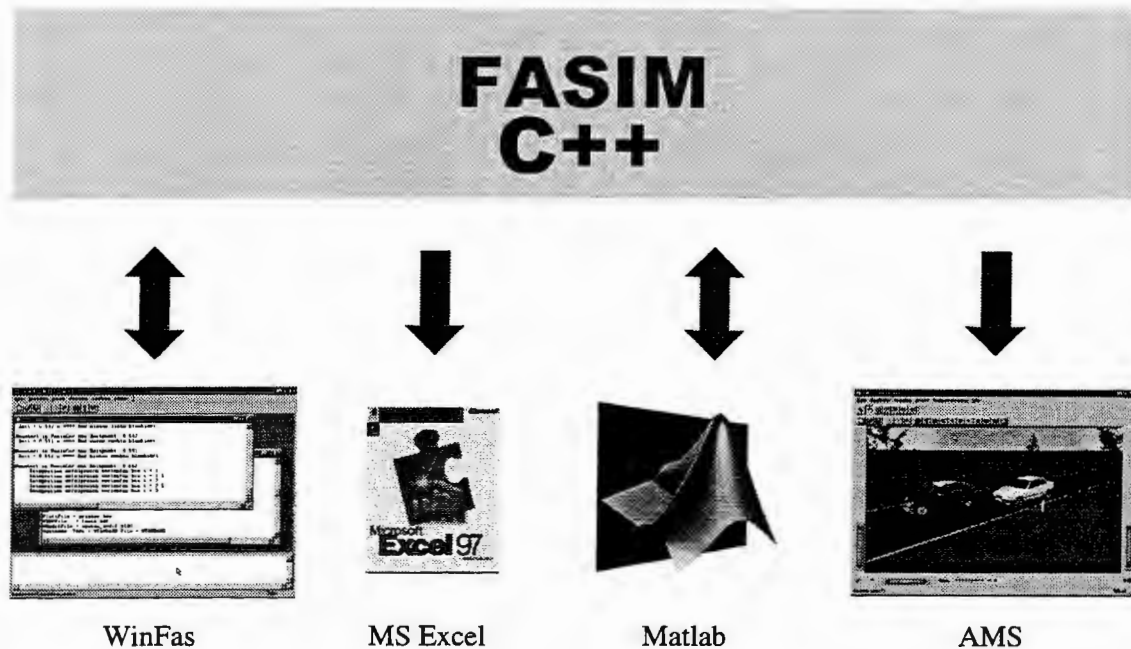
figure 3.1: Data and Functionality Transfer

Thus it is possible to calculate on fast but expensive UNIX-Workstations and control the simulation from inexpensive PCs. The communication between the single programs was realised via TCP/IP with a overlaying in house developed, proprietary transfer protocol, with control sequences to invoke particular functions. The use of these programs with other simulation tools is difficult because of the proprietary transfer protocol. For this reason the programs have been given CORBA-interfaces. WINFAS is specially written for FASIM_C++, but nevertheless it is easy to include new communication channels because the programmer only has to insert a method call, like is required for a local method. During simulation, commands and data have to be transmitted between the simulator and WINFAS. The animation tool AMS was also bound via TCP/IP to FASIM_C++. If for instance the vehicle has to send its current orientation to AMS with TCP/IP, a method is called, which transfers the orientation parameters to the central sending method. This then encodes them and transmits them with a defined control sequence via TCP/IP to a central receiving method of AMS. This interprets the control sequence, reads the data with the correct decoding format from the TCP/IP-stream, and retransmits the data to recipient (figure 3.2a). There is no implicit data type checking, and generally complex data types have to be transformed.
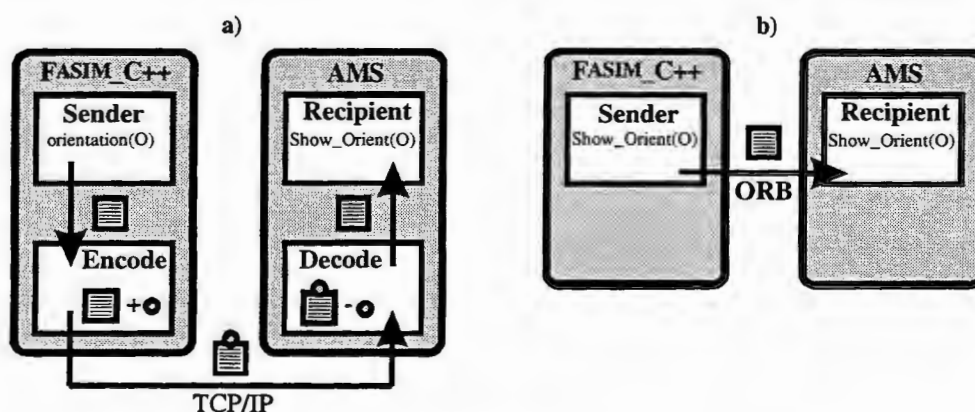
figure 3.2: Data transfer with TCP/IP (a) and CORBA (b)

With CORBA the vehicle method directly calls a method which transmits the data via the ORB to the recipient in AMS (figure 3.2b). Now it is easy to combine any new programs with AMS, directly calling the operations for animation. Complex data types, like structures or arrays can be defined in the interface definition using IDL, thus the programmer can use them as local structures. Therefore orientation and position can now be sent as vectors, with implicit data type checking.

4 Conclusions

Generally, distributing single simulation tasks has two main goals:
- Assembling the simulation environment in a comfortable way.
- Increasing the computing performance.

In some cases these are contradictory aims. For instance, it is very comfortable to use an integrator component, because one only has to pay attention to its interfaces and not to its implementation. The obvious disadvantage is that the whole state vector has to be transmitted to the integrator, and afterwards back to the system. The implementation of the integrator depends on the simulation task. For high performance, approaching real time simulation, combining the integrator and the system in one process is normally a more effective way. However fast and easy assembly with tested numerical components, the distributed simulation system may be preferred. For this reason distributing the peripheral program parts should be the first step to decrease the program size and the development time. Nevertheless, there is a huge potential for developing fully distributed simulation programs.

Finally, we have to answer the question about the advantages of distributed simulation systems. To take full advantage of such a system, we must have a comprehensive catalogue of valuable components, which are accessible from different hardware and software platforms. Then we can assemble small and smart simulation programs for specific simulation tasks. The development times and costs will decrease, furthermore, single components are easier to maintain than large non-transparent monolithic programs. Often several developers are involved for programming such simulation environments, in several companies or universities, all using different hardware platforms or programming languages. This is nearly insignificant, if they are using CORBA and defining their interfaces in IDL. Moreover, it is possible to include existing software by writing CORBA interfaces called *wrappers*. With CORBA it is possible to produce open systems, which can be expanded by a programmer or user.

Acknowledgements

[ADAMSKI97]   D. Adamski, C. Schuster and M. Hiller; *Fahrdynamiksimulation mit FASIM_C++ als Beispiel für die Modellierung mechatronischer Systeme*; VDI-Bericht 1315, pp. 117-141, VDI, Düsseldorf, 1997

[BENNATAN95]   R. Ben-Natan; *CORBA: A Guide to Common Object Request Broker Architecture*; McGraw-Hill, New York; 1995

[MOWBRAY95]   T. J. Mowray and R. Zahavi; *The Essential CORBA: Systems Integration Using Distributed Objects*; John Wiley & Sons; New York; 1995

[OMG95]   Object Management Group; *CORBA Overview*; http://www.omg.org; 1995

[ORFALI96]   R. Orfali, D. Harkey and J. Edwards; *The Essential Distributed Objects Survival Guide*; John Wiley & Sons; New York; 1996

[ORFALI97]   R. Orfali, D. Harkey and J. Edwards; *Instant CORBA*; John Wiley & Sons; New York; 1997

[TALARIAN97]   Talarian Corporation; *Everything You Need To Know About Middleware*, USA, 1997

[TAYLOR93]   D. A. Taylor; *Object-Oriented Information Systems*; John Wiley & Sons; New York, 1993

AN MPP SIMULATION METHOD FOR INHOMOGENEOUS DISTRIBUTED PARAMETER SYSTEMS

L. Dekker, S.W. Brok, Delft University of Technology, The Netherlands

ABSTRACT

In this paper parallel modelling of linear inhomogeneous 1D distributed parameter systems is considered. Two modelling methods in relation to MPP simulation are discussed: a two-point series expansion method and for comparison also an eigenfunction expansion method.

1. INTRODUCTION

In general for computational analysis of both complex and large-scale systems massively parallel computers are necessary. Therefore it is of great importance to develop parallel methods through system modelling aimed at creation and preservation of parallelism in all modelling steps. Such parallel modelling will result in parallel computer models that are optimally efficient, scalable and portable.

In this paper two methods for modelling of linear inhomogeneous 1D distributed parameter systems based on the system concept generalized steady-state are presented. The applicability of the generalized steady-state method[1] to parallel modelling and parallel simulation of linear distributed parameter systems is discussed in a number of papers. This paper emphasizes the application of the gss-method to linear inhomogeneous 1D distributed parameter systems.

In section 2 it is shown that the two-point expansion of an analytical function in its even derivatives on the boundaries of a given interval can also be applied to the construction of the generalized steady-state solution of inhomogeneous 1D distributed parameter systems. This concept is elucidated for some analytical examples. Section 3 deals with an alternative approach, the so-called eigenfunction expansion method. And in section 4 the MPP modelling aspects of both methods will be discussed.

2. TWO-POINT EXPANSION METHOD

The aim in developing the two-point expansion method was the creation of efficient massively parallel models. The method and its application are illustrated for the 1D diffusion system (2.1) with state $z_d(x,t)$:

$$\frac{\partial z_d}{\partial t} - \frac{\partial^2 z_d}{\partial x^2} = f(x,t); \quad 0 \leq x \leq 1, t \geq 0; \tag{2.1}$$

$$z_d(0,t) = \varphi_0(t); z_d(1,t) = \varphi_1(t); z_d(x,0) = \gamma(x)$$

and the 1D hyperbolic system (2.2) with state $z_h(x,t)$:

$$\frac{\partial^2 z_h}{\partial t^2} - \frac{\partial^2 z_h}{\partial x^2} = f(x,t); \quad 0 \leq x \leq 1, t \geq 0; \tag{2.2}$$

$$z_h(0,t) = \varphi_0(t); z_h(1,t) = \varphi_1(t); z_h(x,0) = \gamma(x); \frac{\partial z_h(x,0)}{\partial t} = \eta(x)$$

The distributed source term $f(x,t)$ and the boundary conditions $\varphi_0(t)$, $\varphi_1(t)$ are assumed to be analytic functions. The solutions of the diffusion system (2.1) and the hyperbolic system (2.2) depend on the boundary conditions $\varphi_0(t)$, $\varphi_1(t)$, the inhomogeneous term $f(x,t)$ and the initial conditions of the state and its derivative state: $\gamma(x)$ and $\eta(x)$. Both the inhomogeneous term and the boundary conditions can be conceived to represent input signals for this distributed systems. For a stable system it holds that an input signal, satisfying certain conditions, forces a so-called generalized steady-state solution. For both the diffusion system (2.1) and the hyperbolic system (2.2) the generalized steady-states $z_{de}(x,t)$, $z_{he}(x,t)$ can be conceived to be the superposition of the generalized steady-states $z_d s_e(x,t)$, $z_h s_e(x,t)$, forced by the source term $f(x,t)$ in the case $\varphi_0(t) = \varphi_1(t) = 0$ and the generalized steady-states $z_d b_e(x,t)$, $z_h b_e(x,t)$, forced by the boundary

---

[1] In the paper also the acronym gss is used which stands for generalized steady-state

conditions $\varphi_0(t)$, $\varphi_1(t)$ in the case $f(x,t)=0$. In practice most attention has been spent to the homogeneous case $f(x,t)=0$. In many preceeding publications, a.o. [1], [4], [5], the way how to determine the generalized steady-state $z_{be}(x,t)$ is discussed for different types of distributed parameter systems.

2.1 Source-input generalized steady-state

First the way of determining the generalized steady-states $z_d s_e(x,t)$ and $z_h s_e(x,t)$ will be explained. It is known from literature [2] that a function $f(x)$ of one variable, satisfying certain conditions, can be expressed within an interval as a two-point expansion of the even derivatives of $f(x)$ in the boundary points of the interval. Applying this to $f(x,t)$, considered as a function of x in the interval $[0,1]$, yields that $f(x,t)$ can be written at each time t as a two-point expansion:

$$f(x,t) = \sum_{k=0}^{\infty} \{ g_k(x) f^{(2k,0)}(0,t) + g_k(1-x) f^{(2k,0)}(1,t) \} \qquad (2.3a)$$

where $f(0,t)$ and $f(1,t)$ represent the derivatives of $f(x,t)$ with respect to x of order $2k$ in $x=0$ and in $x=1$ for each time t. The coefficients $g_k(x)$ and $g_k(1-x)$ are polynomials of x of degree $2k+1$. These polynomials satisfy the recurrent relation:

$$g_k^{(2)}(x) = g_{k-1}(x), g_k(0) = g_k(1) = 0, k \geq 1; g_0(x) = 1 - x \qquad (2.3b)$$

Further properties of these polynomials are described in a.o.[1].

1D diffusion system

The generalized steady-state $z_d s_e(x,t)$ is the infinite sum of the generalized steady-states $z_d s_{ek}(x,t)$ and $z_d s_{ek}(1-x,t)$, belonging to the terms $g_k(x) f^{(2k,0)}(0,t)$ and $g_k(1-x) f^{(2k,0)}(1,t)$, $k=0,\mathrm{L},\infty$ in the two-point expansion (2.3a). Denote the functions $f^{(2k,0)}(0,t)$ and $f^{(2k,0)}(1,t)$ respectively as $u_k(t)$ and $v_k(t)$.

Consider now the generalized steady-state $z_d s_{ek}(x,t)$ corresponding to the general term $g_k(x)u_k(t)$ of the first of the two subseries of (2.3a). This gss $z_d s_{ek}(x,t)$, forced by $u_k(t)$, is the solution of:

$$\frac{\partial z_d s_{ek}}{\partial t} - \frac{\partial^2 z_d s_{ek}}{\partial x^2} = g_k(x)u_k(t); \quad \varphi_0(t) = \varphi_1(t) = 0 \qquad (2.4a)$$

That's to say $z_d s_{ek}(x,t)$ can be constructed as an infinite linear combination of the time-derivatives of $u_k(t)$ with space-dependent coefficients:

$$z_d s_{ek}(x,t) = \sum_{n=0}^{\infty} \{ p_n(x)u_k^{(n)}(t) + p_n(1-x)v_k^{(n)}(t) \} \qquad (2.4b)$$

Substitution of the first subseries of (2.4b) in (2.4a), yielding:

$$\sum_{n=0}^{\infty} p_n(x)u_k^{(n+1)}(t) - \sum_{n=0}^{\infty} p_n^{(2)}(x)u_k^{(n)}(t) = g_k(x)u_k(t) \qquad (2.5a)$$

results in the set of recurrent equations:

$$p_0^{(2)}(x) = -g_k(x); p_n^{(2)}(x) = p_{n-1}(x), n \geq 1 \qquad (2.5b)$$

Together with property (2.3b) and the condition that $z_d s_{ek}(x,t)$ must be zero on the boundaries yields:

$$p_n(x) = -g_{k+n+1}(x), n = 0,1,\mathrm{L} \qquad (2.6)$$

In a similar way the general term of the other subseries of (2.4b) can be handled. Hence, it appears:

$$z_d s_e(x,t) = -\sum_{k=0}^{\infty} \sum_{n=0}^{\infty} \{g_{k+n+1}(x) \overset{(2k,n)}{f}(0,t) + g_{k+n+1}(1-x) \overset{(2k,n)}{f}(1,t)\} \tag{2.7}$$

where $f(0,t)$ and $f(1,t)$ represent the derivatives of $f(x,t)$ with respect to x of order $2k$ and with respect to t of order n in respectively $x=0$ and in $x=1$.

This outcome for $z_d s_{ek}(x,t)$ satifies the homogeneous boundary conditions and the inhomogeneous diffusion equation as can easily be verified by substitution in (2.1).

1D hyperbolic system

The generalized steady-state $z_h s_e(x,t)$, forced by $u_k(t)$, is the solution of

$$\frac{\partial^2 z_h s_{ek}}{\partial t^2} - \frac{\partial^2 z_h s_{ek}}{\partial x^2} = g_k(x) u_k(t); \quad \varphi_0(t) = \varphi_1(t) = 0 \tag{2.8a}$$

Again, $z_h s_{ek}(x,t)$ can be constructed as an infinite linear combination of the time-derivatives of $u_k(t)$:

$$z_h s_{ek}(x,t) = \sum_{k=0}^{\infty} \sum_{n=0}^{\infty} \{\tilde{p}_n(x) u_k^{(n)}(t) + \tilde{p}_n(1-x) v_k^{(n)}(t)\} \tag{2.8b}$$

Substitution of the first subseries of (2.8b) in (2.8a), yielding:

$$\sum_{k=0}^{\infty} \sum_{n=0}^{\infty} \tilde{p}_n(x) u_k^{(n+2)}(t) - \sum_{k=0}^{\infty} \sum_{n=0}^{\infty} \tilde{p}_n^{(2)}(x) u_k^{(n)}(t) = g_k(x) u_k(t) \tag{2.9a}$$

results in the set of recurrent equations:

$$\tilde{p}_0^{(2)}(x) = -g_k(x); \quad \tilde{p}_{2m}^{(2)}(x) = \tilde{p}_{2m-2}(x); \quad \tilde{p}_{2m-1}^{(2)}(x) = 0, \quad m \ge 1 \tag{2.9b}$$

This together with the condition that $z_h s_{ek}(0,t) = z_h s_{ek}(1,t) = 0$ yields:

$$\tilde{p}_{2m}(x) = -g_{k+m+1}(x), \; \tilde{p}_{2m+1}(x) = 0, \quad m \ge 0 \tag{2.10}$$

In a similar way the general term of the other subseries in (2.8b) can be handled. Hence it appears:

$$z_h s_e(x,t) = -\sum_{k=0}^{\infty} \sum_{n=0}^{\infty} \{g_{k+n+1}(x) \overset{(2k,2n)}{f}(0,t) + g_{k+n+1}(1-x) \overset{(2k,2n)}{f}(1,t)\} \tag{2.11}$$

2.2 Boundary-input generalized steady-state

1D diffusion system

For the diffusion system (2.1) the boundary-input generalized steady-state $z_d b_e(x,t)$ is the gss, forced by $\varphi_0(t)$ and $\varphi_1(t)$ in the case $f(x,t) = 0$. Hence, the gss $z_d b_e(x,t)$ can be formulated as an infinite linear combination of the time-derivatives of $\varphi_0(t)$ and $\varphi_1(t)$ with space-dependent coefficients. This two-point expansion of $z_d b_e(x,t)$ can be derived in a similar way as has been done for the gss $z_d s_e(x,t)$ by substitution of this series expansion in the state equation and the boundary conditions of (2.1) and further evaluation of the resulting identities. But, because at each time the time-derivatives of order k in the boundary points are equal to the space-derivatives of order $2k$ in these points, the gss $z_d b_e(x,t)$ can also be found straightforward by replacing in (2.3a) the space derivatives of $f(x)$ of order $2k$ in the points $x=0$ and $x=1$ by the time-derivatives of $\varphi_0(t)$ and $\varphi_1(t)$ of order k. This results in the two-point expansion:

$$z_d b_e(x,t) = \sum_{k=0}^{\infty} \{g_k(x) \varphi_0^{(k)}(t) + g_k(1-x) \varphi_1^{(k)}(t)\} \tag{2.12a}$$

The eigensolutions $\phi_k(x,t)$ of the diffusion system are:

$$\phi_k(x,t) = \exp(-k^2 \pi^2 t) \sin k\pi x \quad k = 0, 1, \mathbf{L} \tag{2.12b}$$

1D hyperbolic system

22

For the hyperbolic system the boundary-input generalized steady-state $z_h b_e(x,t)$ is the gss $z_h b_e(x,t)$, forced by $\varphi_0(t)$ and $\varphi_1(t)$ in the case $f(x,t) = 0$. Hence, the gss $z_h b_e(x,t)$ can be formulated as an infinite linear combination of the time-derivatives of $\varphi_0(t)$ and $\varphi_1(t)$ with space-dependent coefficients. From the homogeneous system equation it follows that at each time the time-derivatives of order $2k$ in the boundary points are equal to the space-derivatives of order $2k$ in these points. Hence, replacing in (2.3a) the space-derivatives of $f(x)$ of order $2k$ in the points $x=0$, $x=1$ by the time-derivatives of $\varphi_0(t)$ and $\varphi_1(t)$ of order $2k$ results in the two-point expansion:

$$z_h b_e(x,t) = \sum_{k=0}^{\infty} \{g_k(x)\varphi_0^{(2k)}(t) + g_k(1-x)\varphi_1^{(2k)}(t)\} \tag{2.13a}$$

The eigensolutions $\phi_{1k}(x,t)$ and $\phi_{2k}(x,t)$ of the hyperbolic system are:

$$\phi_{1k}(x,t) = \sin k\pi t \sin k\pi x \text{ and } \phi_{2k}(x,t) = \cos k\pi t \sin k\pi x \quad k = 0,1,\mathbf{L} \tag{2.13b}$$

Notice that a characteristic of a generalized steady-state solution is that it does not contain eigensolutions. Dependent on the initial conditions there arises a transient solutions, composed of eigensolutions.

2.3 Some analytic examples
Source-input generalized steady-state of the 1D diffusion system
As a first example the case is considered that the source-input is only space-dependent: $f(x,t) = h(x)$. In that case the expression (2.7) reduces to:

$$z_d s_e(x,t) = -\sum_{k=0}^{\infty} \{g_{k+1}(x)h^{(2k)}(0) + g_{k+1}(1-x)h^{(2k)}(1)\} \tag{2.14a}$$

Taking for example $h(x) = 1$ and applying the property $g_1(1-x) = \frac{1}{6}(x - x)$ yields:

$$z_d s_e(x,t) = -g_k(x) - g_1(1-x) = \frac{1}{2}x(1-x) \tag{2.14b}$$

In the second example the source-input is taken as a function of time only: $f(x,t) = e$ '. The space-independency of $f(x,t)$ then implies:

$$f^{(2k,n)}(x,t) = 0, k > 0 \text{ and } f^{(0,n)}(x,t) = (-1)^n e^{-t} \tag{2.15a}$$

Substitution this result in equation (2.7) yields:

$$z_d s_e(x,t) = \sum_{n=0}^{\infty} \{g_{n+1}(x)(-1)^n + g_{n+1}(1-x)(-1)^n\}e^{-t} \tag{2.15b}$$

Applying further the expression of the generating function for the polynomials $g_k(x)$:

$$\frac{\sinh \sqrt{b}(1-x)}{\sinh \sqrt{b}} = \sum_{k=0}^{\infty} g_k(x) b^k \tag{2.15c}$$

results in:

$$z_d s_e(x,t) = \left\{\frac{\sin(1-x) + \sin(x)}{\sin(1)} - 1\right\}e^{-t} \tag{2.15d}$$

Notice that $z_d s_e(x,t) = 0$ for both $x = 0$ and $x = 1$.

As a last example of the source-input generalized steady-state of the diffusion system a source-input with separated variables is considered:

$$f(x,t) = \sin \pi x f(t) \text{ with } f(t) = \pi^2 \sin at + a\cos at \tag{2.16a}$$

23

It then holds that also the derivatives of the source term with respect to x of order $2k$ are functions with separated variables:

$$f^{(2k,0)}(x,t) = (-\pi^2)^k \sin \pi x \, f(t) \tag{2.16b}$$

In this case it is obvious to assume that:

$$z_d s_e(x,t) = \sin \pi x \, g(t). \tag{2.16c}$$

Substitution of this in the system equation and further evaluation results in:

$$z_d s_e(x,t) = \sin \pi x \sin at \tag{2.16d}$$

Source-input generalized steady-state of the 1D hyperbolic system
As a first example again the input-source is taken space-dependent only: $f(x,t) = h(x)$. The equation (2.11) then reduces to:

$$z_h s_e(x,t) = -\sum_{k=0}^{\infty} \{ g_{k+1}(x) h^{(2k)}(0) + g_{k+1}(1-x) h^{(2k)}(1) \} \tag{2.17}$$

Notice that for $f(x,t) = h(x)$ the generalized steady-states $z_d s_e(x,t)$ of equation (2.14a) and $z_h s_e(x,t)$ of equation (2.17) are equal. This is not amazing because then both system equations become equal to the same time-independent system equation: $\partial^2 z / \partial x^2 = h(x)$.

In the second example again a source-input with separated variables is considered:

$$f(x,t) = (\pi^2 - a^2) \sin \pi x \sin at \tag{2.18}$$

Following the same procedure as in the last example of the source-input generalized steady state of the 1D diffusion system yields:

$$z_h s_e(x,t) = \sin \pi x \sin at \tag{2.19}$$

which is the same result as for in equation (2.16d).

Boundary/total-input generalized steady-state of the 1D hyperbolic system
For boundary conditions $\varphi_0(t) = \varphi_1(t) = \sin at$ and zero source-input ($f(x,t) = 0$) the two-point expansion (2.13a) can be used the find the boundary steady-state solution:

$$z_h b_e(x,t) = \frac{\sin ax + \sin a(1-x)}{\sin a} \sin at \tag{2.20}$$

The total generalized steady-state $z_{he}(x,t)$ is the sum of $z_h s_e(x,t)$ in expression (2.19) and $z_h b_e(x,t)$ in expression (2.20). Notice that the introduction of boundary-inputs opens the possiblity to apply boundary-control of the combined effect of the source-input gss and the transient state solution.

3. EIGENFUNCTION EXPANSION METHOD
Besides the two-point expansion method for constructing the gss solution of distributed parameter systems other approaches are possible. One approach is the so-called eigenfunction method. In the paper this method will be demonstrated for the 1D diffusion system (2.1) with homogeneous boundary conditions $\varphi_0(t) = \varphi_1(t) = 0$ and zero initial condition $\gamma(x) = 0$. But the method can easily be extended to hyperbolic systems. Applying the method of separation of variables it is wellknown that

$$\phi_k(x,t) = \exp(-k^2 \pi^2 t) \sin k \pi x \quad k = 0,1,L \tag{3.1}$$

are solutions (eigenfunctions with eigenvalues $k\pi$) that satisfies the system equation with homogeneous boundary conditions. Because $\{\sin k\pi x\}_{k=0}$ forms a complete set of orthogonal functions on the interval $[0,1]$ any other arbitrary function can be expanded into an infinite series of these orthogonal functions, the so-called eigenfunction expansion.
Considering the source term $f(x,t)$ of the 1D diffusion suystem (2.1) then $f(x,t)$ can be expanded into an infinite series of eigenfunctions:

$$f(x,t) = \sum_{k=0}^{\infty} f_k(t) \sin k\pi x \quad \text{with} \quad f_k(t) = 2\int_0^1 f(x,t) \sin k\pi x \, dx \tag{3.2}$$

We now try to find a solution $z_d(x,t)$ of the form:

$$z_d(x,t) = \sum_{k=0}^{\infty} c_k(t) \sin k\pi x \tag{3.3}$$

Substitution in the 1D diffusion equation (2.1) taking into account that $\gamma(x) = 0$ results into an analytical expression for $c_k(t)$. Substitution into (3.3) leads to the solution:

$$z_d(x,t) = \sum_{k=0}^{\infty} \int_0^t \exp\{-k^2\pi^2(t-\tau)\} f_k(\tau) d\tau \sin k\pi x \tag{3.4}$$

which, as easily can be verified, fulfils the boundary conditions and the initial condition.
The easiest way to extract the gss solution $z_d s_e(x,t)$ from this expression is by means of the Laplace transformation of $z_d(x,t)$ with respect to the time t. For simplicity only the result is given. For a more detailed description see [7]. The gss solution $z_d s_e(x,t)$ is given by:

$$z_d s_e(x,t) = -\sum_{k=0}^{\infty} \left[\int_t^{\infty} \exp(\zeta - k^2\pi^2)(t-\tau) f_k(\tau) d\tau \right]_{\zeta=0} \sin k\pi x \tag{3.5}$$

in which the parameter ζ can be chosen in such a way that the integral in this expression will be bounded. In case that the source term $f(x,t)$ is a function of x only, i.c. $f(x,t) = f(x)$ and $f_k(t) = f_k$, the integral can be evaluated which then leads to the following expression:

$$z_d s_e(x,t) = -\sum_{k=0}^{\infty} f_k \frac{\sin k\pi x}{k^2\pi^2} \tag{3.6}$$

4. MPP MODELLING ASPECTS

For both the source-input and the boundary-input generalized steady-states, i.c. $z_d s_e(x,t)$, $z_h s_e(x,t)$, $z_d b_e(x,t)$, and $z_h b_e(x,t)$, function evaluations of an array of points have to be carried out. In particular it concerns function evaluations of the space-dependent coefficients $g_k(x)$ and $g_k(1-x)$ as well as the function $f(x,t)$ and its derivatives with respect to x and t.
Evaluation of a given function of one or more arguments for an array of N argument values have the common property that the function evaluation for different argument values is identical as well as mutually independent. Hence, parallel evaluation of a function for an array of argument values can be based upon SPMD programming without any communication between computing nodes. It means that a speed up of N with a 100% efficiency is possible for a distributed-memory parallel computer of N computing nodes by allocating to each computing node as computing task the function evaluation. For D computing nodes (D a divider of N) and an enlarged task per computing node equal to the function evaluation for N/D argument values again the speed up is D with an efficiency of 100%. In other cases one or more computing nodes are idle for some time, resulting in a speed up smaller than the number of computing nodes and an efficiency smaller than 100%.
This makes it possible to perform the data processing and the data exchange in an efficient way by combining SPMD programming with a step-lock synchronization procedure for the data exchange. This combination will be denoted by Single Program Step Lock (SPSL) programming. Step lock synchronization means that after the data processing of a (number of) function evaluation(s) step is ready the data processing is locked simultaneously in all computing nodes concerned and the exchange of the current values of the state variables is started. This continues until the new values of all state components have been distributed to the computing nodes concerned. Then the computing nodes start simultaneously the data processing of the next (number of) function evaluation(s) step. A parallel SPSL program can be handled in principle by any type of interconnect system because the only demand is that speed and bandwidth of the interconnect system are large enough to perform the total data exchange in an acceptable or even a neglectable time compared to the time needed for the total data processing. Or in other words, the presented method is a scalable one. That's to say, enlarging the problem size does not have much influence on

the processing time, as long as the parallellism of a (scalable) parallel computer remains sufficiently large.

Notice that the usual numerical way of modelling through discretization of the distributed parameter system results in the necessity to solve a large set of linear equations.

The above way of creation of parallellism by means of problem splitting is schematically represented in figure 1.
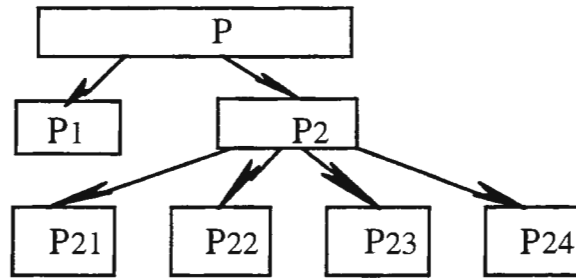
Figure 1. Creation of parallellism through problem splitting.

In general the same holds for the eigenfunction expansion method. But in this case the calculation is a little bit more complicated because the time-dependent coefficients in the infinite series (3.5) and the time-dependent coefficients $f_k(t)$ are now integral expression which has to be solved by conventional numerical methods. However, when the source term $f(x,t)$ and all its derivatives with respect to x and t vanish at the boundaries of the interval $[0,1]$ the eigenfunction method is the only method left.

5. CONCLUSIONS

In the paper it is elucidated for linear, time-independent inhomogeneous distributed parameter systems that the generalized steady-state is a useful system concept in modelling. It is also exposed that an important consequence of massively parallel simulation is a quite different approach in system analysis and system modelling, aimed at creating and disclosing the parallelism of a system in the analysis phase and to preservation of this parallelism as much as possible in both the modelling and implementation phase.

The results until now look promising that future parallel modelling of problems from the very beginning may often lead in practice to highly parallel models with much unbounded parallelism, i.e. the resulting parallel models are largely of the SPMD programming and SL synchronization type. This outcome is important for future portability of these parallel models to a wide range of parallel computers.

A further conclusion is that very likely a great change looks imminent in system modelling and simulation.

6. REFERENCES

[1] L. Dekker; Numerical aspects of the one-dimensional diffusion equation; PhD thesis, Delft University of Technology, 1964

[2] J.M. Whittaker; On Lidstone series and two-point expansions of analytical functions; Proc. London Math. Soc. (2) vol 36, 1934, pp. 451-469.

[3] L. Dekker; 4-Point series expansion, a massively parallel formulation of a function of two variables; Report 93-122, ISSN 0922-5641, Fac. of Technical Mathematics & Informatics, Delft University of Technology, 1993.

[4] L. Dekker, S.W. Brok, F.J. Lingen; Semi-analytical method for MPP simulation of distributed parameter systems; Proc. EUROSIM Conference Massively Parallel Processing Applications and Developments, Delft, 1994, pp.579-590.

[5] L. Dekker, S.W. Brok; A massively parallel simulation method for parabolic and hyperbolic systems; Proc. 1995 EUROSIM Conference, Vienna, 1995, pp. 249-254.

[6] L. Dekker, S.W. Brok; Parallel distributed parameter system-adapted simulation and inter-polation; Proc. EUROSIM Conference HCPN Challeng es in Telecomp and Telecom, Delft, 1996, pp. 93-101.

[7] S.W. Brok; Systems, Modelling and Simulation (in Dutch), Graduate Course (Chapter 8 and 9), Delft University of Technology, August 1994.

Using Cray T3E for the parallel calculation of cellular radio coverage

Pentti Huttunen[1], Jari Porras[1], Jouni Ikonen[1] and Kari Sipilä[2]

[1]Lappeenranta University of Technology, P.O.Box 20, FIN-53851 Lappeenranta, Finland
{Pentti.Huttunen, Jari.Porras, Jouni.Ikonen}□lut.fi
[2]Nokia Research Center, P.O. Box 407, FIN-00045 NOKIA GROUP, Finland
Kari.Sipila□research.nokia.com

Abstract

A ray tracing based cellular coverage simulator has been parallelized in order to speed up excecution. The simulator has been developed for accurate prediction of radio wave propagation in urban microcells. It combines a vertical plane model simulating the wave propagation over the buildings and a horizontal plane model simulating the wave propagation in the street canyons between the buildings. The horizontal plane calculation is made by ray tracing using multiple diffractions around the corners of building. The parallel version has been implemented for a distributed memory machine, CRAY T3E. Parallelization of the vertical and horizontal plane calculations reduces the execution times considerably. Results indicate that only a small number of processors are required for efficient parallel execution. The load balancing algorithm has an important role in obtaining good speedups.

Introduction

Calculation of radio wave propagation and field strength is one of the most critical tasks in any computer based cellular network planning system or simulator. Because of increasing demand for capasity in cities the cells are getting smaller and smaller and there is a need for an accurate prediction of coverage in a complex urban geometrical setting. During past years several methods have been developed for this purpose. Some of the models are empirical, or statistical based on field strength measurements and experience, see e.g. [14], and some of them are deterministic using accurate maps of the service area as input. Most popular deterministic methdos are based on ray tracing because of its accuracy. Examples of ray tracing based methods can be found from [5, 9, 10, 13]. In [7] there is a good overview.

The ray tracing method enables accurate results but it also demands a huge amount of calculations. Due to the high amount of calculations it requires a powerful computer or a lot of time. As several simulations need to be run before optimal parameters for the cellular network can be found, the execution time of the simulation must be kept reasonable. Improved algorithms have been developed [8, 13] but the simulation time remains too long for the iterative process of radio network planning.

Parallel computing offers a solution for time consuming field strength calculations. Calculation times can be significantly reduced by partitioning the problem into independent parts and by executing the parts in parallel. Coverage calculation is a suitable application for parallel execution as the independent parts can be easily found. However, the size of different parts may vary which may result in uneven execution times. In order to achieve the best possible speedup a work balancing algorithm need to be implemented for the application.

In this paper, the use of Cray T3E as a parallel environment for the coverage calculation simulator has been studied. Since Cray T3E is a distributed memory machine, message passing is used for the communication between processors. The parallelization of coverage calculation is implemented by using MPI (Message Passing Interface) and SHMEM (Shared Memory) library calls. The following chapters present the coverage calculation, parallelization of the calculation and the results.

Coverage calculation

Coverage calculation means determining those places in the service area of the radio network where a certain base station (BTS) can be used for a two way connection between the mobile station (MS). This is done by calculating the attenuation between the BTS location and other locations on a large enough area around the BTS. In simulators it is sufficient to create a uniform square grid of receiver points (i.e. possible locations of the MS) over the simulation area where each point represents a small square. Density of the grid may vary according to the desired accuracy. A 4*4 meter square is usually considered accurate enough. Figure 1 presents the creation of receiving points by using a grid. The closed polygons visible in Figure 1 represent buildings seen from above.
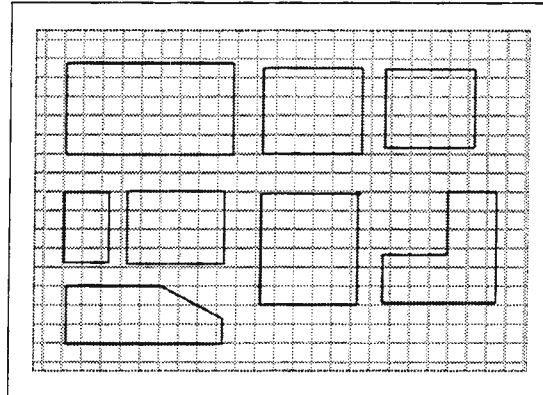
Figure 1. Creation of the receiving grid, i.e. receiving points.

In this work the coverage calculation has been implemented for microcell environments. Microcells are used for an efficient utilization of the available frequency in dense city environments with high capasity requirements. In microcells the base station antennas are typically located on the walls clearly below the buildings' roof and therefore the size of the cell is usually less than 1km^2 [4,11].

The software uses as input BTS parameters, MS parameters and an accurate map of the area where the buildings has been presented as closed polylines with building height information. As output it produces the average field strength to every receiving point outside buildings, i.e. indoor coverage calculation is not implemented in the current version.

Two different kinds of calculations are needed for accurate results, i.e. vertical and horizontal plane. In vertical plane calculation the field strength is computed by checking the vertical plane connecting transmitting and receiving point and by calculating losses due to the obstacles and free space. Only rough field strength estimations can be achieved by using vertical plane calculation alone. The basic idea of the vertical plane calculation is presented in the Figures 2 and 3.

In the horizontal plane the field strength calculation is based on the use of reflections and diffractions which are modeled by using ray tracing. Basic types of rays are depicted in Figure 4. In earlier ray tracing implementations a large number of rays have been launched from the transmitter to every direction and followed until their power falls below a threshold. Every time a ray goes close enough to a receiving point the received power is updated. A propagating ray can reflect from the obstacles or in the case of diffraction the propagating ray divides into multiple rays diffracting to every direction from the diffraction corner. As diffractions generate multiple new rays the number of rays increases exponentially as a function of diffraction order requiring a huge amount of calculations.
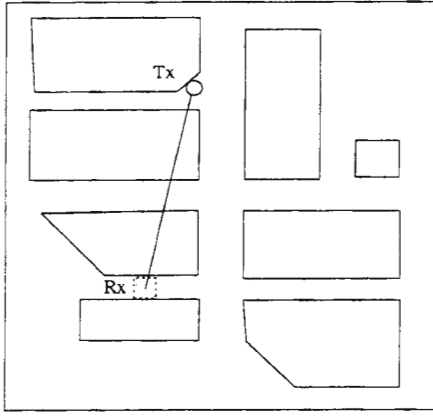
Figure 2. Vertical plane calculation seen from above.
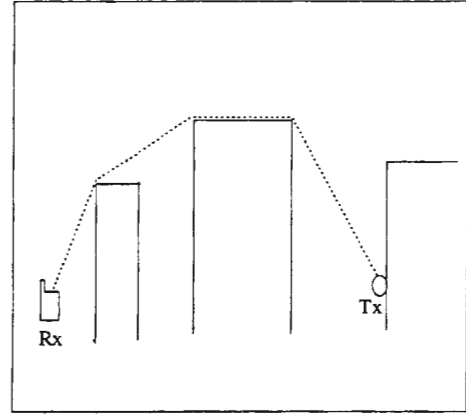
Figure 3. Vertical plane calculation seen from the ground level.
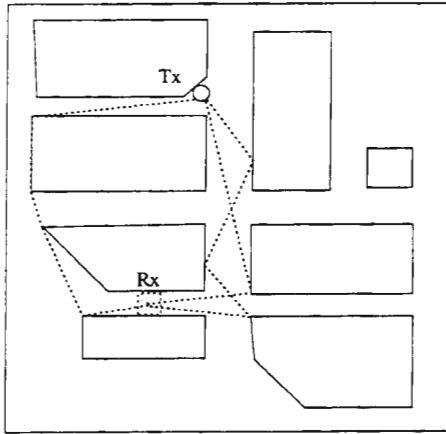
Figure 4. Basic types of ray propagation.
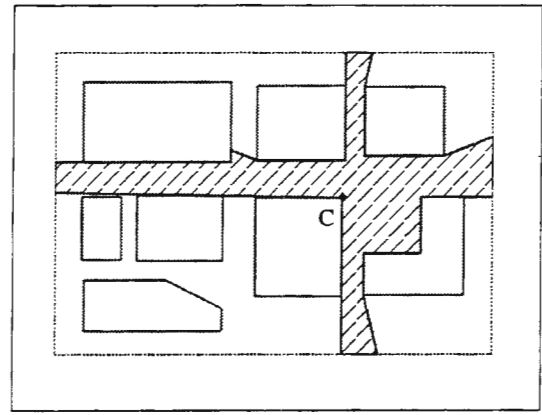
Figure 5. Line-of-sight polygon of corner C.

Due to the amount of computation in the horizontal plane calculation a more sophisticated and accurate method has been developed. The new ray tracing method is based on the use of transmitting points' line-of-sight (LOS) polygons [8]. A transmitting point's LOS polygon represents the area that can be seen from the transmitting point. LOS polygons are created for the BTS as well as to all diffracting corners. Figure 5 presents the LOS polygon of a diffracting corner C. When a ray arrives to the corner C the polygon illustrates those receiving points that can be reached by the ray when it diffracts from corner C. By using this method no single ray have to be traced but ray tracing merely reduces to filling of polygons. The field strength values of the receiving points in LOS polygons are updated according to the power value of the ray arriving to the transmitting point, the angle between the in-coming and out-going rays and the distance from the transmitting point to the receiving point. In this paper reflected rays are ignored because of their insignificant contribution to the local mean received power [12] in case of microcells. Multiple diffractions are implemented so that after a ray has diffracted from a corner it can diffract from another corner and so on up to a predetermined attenuation threshold.

Parallel approach to the coverage calculation

In coverage calculation most of the time is consumed in updating the field strength values of the receiving points. Both vertical and horizontal plane calculations are suitable for the parallelization. In the vertical plane calculation the division between processors is trivial, since each receiving point will take approximately the same time to process. Receiving points are divided to the processors by using a simple division, i.e. each processor calculates an equal sized simulation area. It should be noticed that the actual

work differs as only outdoor receiving points are considered. This difference has no significance as the vertical plane calculation takes only a negligible time when compared to the horizontal plane calculation.

In horizontal plane calculation the field strength values are updated by filling the LOS polygons. Since the polygons are independent objects they can be processed in parallel. However, the sizes of LOS polygons are not equal as the area seen from corners is different. The computation time of a single LOS polygon depends both on the number of its receiving points and the number of times the polygon is filled. A LOS polygon is filled as many times as there are rays arriving to the corner. This makes the work balancing more challenging. The simple division method used in vertical plane calculation is not sufficient in the horizontal plane calculation. Therefore, the WorkPool algorithm was developed to handle the work balancing between the processors.

WorkPool algorithm

The WorkPool algorithm makes the division of LOS polygons among processors according to the real amount of work. In order to realize the real amount of work the total number of receiving points per LOS polygon is first calculated and then multiplied by the number of rays arriving to the base corner of the LOS polygon. The real work of a LOS polygon is stored with the number of corner to a table. The table is then sorted in descending order according to the real amount of work. LOS polygons are processed starting from the largest, i.e. from the top of the table. The table acts like a pool from where processors fetch a new LOS polygon to process until all polygons are filled. A free processor will always process the largest free polygon. The WorkPool algorithm proved to be a suitable solution for coverage calculation.

Implementation

Table 1 presents the MPI [1,6] and SHMEM [2, 3] library calls which are used for parallel approach of the coverage calculation. MPI_Init() and MPI_Finalize() calls are necessary for the initialization and termination of the parallel environment. Communication between processors is implemented by using MPI_Bcast() and MPI_Reduce() library calls. MPI_Bcast() call is used to transfer data from one processor to the predefined set of processors. At the end of the simulation the computed field strength values are collected from processors to a single processor by using MPI_Reduce() call. The SHMEM call is used for implementing the WorkPool algorithm.

Table 1. MPI and SHMEM library calls used in the parallelization.

MPI library call	SHMEM library calls
MPI_Init() MPI_Finalize() MPI_Bcast() MPI_Reduce()	fast_mem_short_finc()

In parallel version of coverage calculation several aspects affect to the achieved performance, i.e. generation of LOS polygons, their distribution, communication between processors, etc. Different approaches were considered for the work and communication optimization. According to the performed studies the best performance was achieved by using pre-calculated LOS polygons, sequential LOS polygon read and one to all broadcast.

Cray's own SHMEM call is used to speed up the WorkPool algorithm. Fetching of the new index to the table of LOS polygons is implemented with the fast_mem_short_finc() call. This call takes care of the synchronization between processors and returns the next free LOS polygon to the calling processor. The use

of SHMEM call speeds up the communication as SHMEM calls are processed by the Cray T3E's support circuitry. A component called backmap [3] is used to maintain the cache coherency.

Test cases and results

Five test cases are used for the evaluation of the parallel coverage calculation. Test cases are from two kind of cities, European (maps 1-3) and Asian (maps 4 and 5). In all cases the simulation area represents an area covered by a microcellular network. Maps 1-3 present a typical European city with a lot of small buildings and empty space, i.e. parks and streets. The streets are wide, long and straight. The amount of free space is relatively high compared to the whole area. Maps 4 and 5, are from an Asian city. They have large buildings with little free space. The streets are narrower and shorter than in maps 1-3. Therefore, the number of receiving points outside buildings is much smaller. Test cases and their parameters are presented in Table 2.

Table 2. Test cases used in the simulator.

Test case	Size of the calculation area (m)	Number of receiving points to update	Number of corners
Map1	1 640 x 1 470	84140	1436
Map2	1500 x 1380	71258	1251
Map3	1250 x 1730	96278	999
Map4	1270 x 840	46656	511
Map5	1270 x 840	46656	511

In each simulation experiment the coverage is calculated only for a single BTS. Spacing of the receiving grid in all test cases is 4 meters, i.e. each receiving point represents a square of 4*4 meters. Number of receiving points to update -column tells the amount of outdoor receiving points which will have the field strength information at the end of the simulation. Actual number of updates is bigger because each receiving point is likely to be updated several times. Number of corners -column indicates the actual number of diffracting corners in the simulation area. Each map has been chosen so that it represents different characteristics of cities.
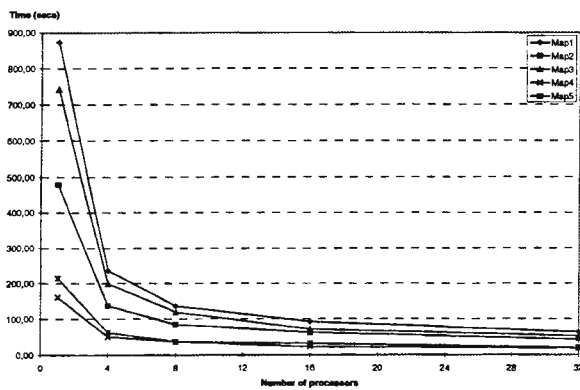
Figure 6. Execution times in Cray T3E.
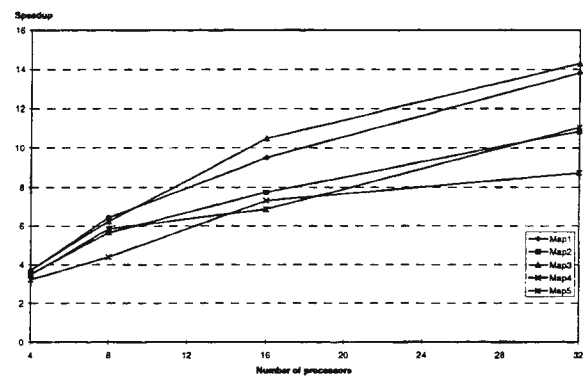
Figure 7. Achieved speedups in Cray T3E.

Figure 6 presents the execution times of all test cases. Parallel simulations were run using 4, 8, 16 and 32 processors. The achieved results are compared to the sequential version of the same simulator. It can be observed that parallel simulation performs clearly faster than the sequential simulation. However, the minimum execution time is achieved with relatively small number of processors. Even though the application is highly parallel the scalability of the problem in distributed memory environment remains low. Figure 7 illustrates the achieved speedups when parallel simulation is compared to the sequential. The low

scalability of the problem can be noticed as low speedups. With 4 processors speedups between 3-3,8 are achieved whereas with 32 processors the speedups remain between 9-14.

Low scalability of the problem in distributed memory environment is due to the sequential part and the communication overhead. Eventually the amount of work in each processor is so small that the sequential part (disk I/O) and the communication overhead generated by the parallel part will dominate the execution time. As far as the microcell simulation is concerned disk I/O takes about 20% of the total execution time. Transfer of LOS polygons to the processors consumes about 15% of the execution time. With larger simulation areas there would be more work for processors and the problem would be more scalable.

Conclusions

Execution times of the cellular radio coverage calculation simulator were reduced significantly by using parallel computing approach. Both vertical and horizontal plane calculations of ray tracing based model were parallelized. A simple division of receiving points was found to be the fastest solution for the vertical plane calculation. WorkPool algorithm proved to be a solid way to divide the work load uniformly in the horizontal plane calculation. The results indicate that good speedups can be achieved for the parallel radio coverage calculation. However, the distributed memory environment, e.g. Cray T3E, is not the best possible platform for the application. Especially, the low scalability of the application prevents the use of large number of processors. In shared memory environment the achieved speedups and scalability are expected to be much higher due to the non-existing communication overhead.

References

[1] Cray Research: "Message Passing Toolkit: MPI programmer's manual", Cray Research Inc., 1996.

[2] Cray Research: "CRAY T3E Applications Programming", Cray Research Inc., Oct. 1996.

[3] Cray Research: "Cray T3E Optimization", Cray Research Inc., Jan. 1997.

[4] Feuerstein, M. et. al. "Path loss, Delay spread, and Outage Models as Functions of Antenna Height for Microcellular System Design", IEEE Transactions on vehicular technology, Vol 43, NO. 3, August 1994

[5] Fritsch, T., Tutschku, K., Leibnitz, K.: "Field strength prediction by ray-tracing for adaptive base station positioning in mobile communication networks", University of Wurzburg, Research report No. 122, Aug. 1995.

[6] Haataja J., Mustikkamäki K.: "Parallel programming with MPI" (in Finnish), CSC-Tieteellinen laskenta Oy, 1997.

[7] Liang G., Bertoni H. L., "Review of ray modeling techniques for site specific propagation prediction" in Wireless Communications, TDMA versus CDMA, ed. S.G. Glisic, Kluwer London, 1997, pp.323-343.

[8] Heiska, K., Kangas, A.: "Microcell propagation model for network planning", IEEE PIMRC, 1996

[9] Inanoglu, H., Topuz, E.: "A ray based indoor propagation model for DECT applications", European Simulation Symposium 1994, pp. 267-271.

[10] Salmi M.: "Parallel ray tracing in propagation modelling of indoor mobile radio communication", Research Report 51, Lappeenranta University of Technology, ISBN 951-763-924-4.

[11] Sarnecki, J., et. al.: "Microcell design principles", IEEE Communications Magazine, April 1993, pp. 76-82.

[12] Sipilä K., Heiska K.: "Can ray tracing be used as a fading generator in simulating micro cellular mobile radio system?", The 8th International Conference on Wireless Communications, July 96, Calgary, Canada.

[13] Tutschku, K., Leibnitz, K.: "Fast Ray-Tracing for Field Strength Prediction in Cellular Mobile Network Planning", University of Wurzburg, Research report No. 134, Jan 1996.

[14] Parsons, J.D, "The Mobile Propagation Channel", Wiley & Sons, Inc. New York, 1992, ch. 4.

Analyzing distributed simulation

Jouni Ikonen[1], Jari Porras[1] and Jarmo Harju[2]

1 Lappeenranta University of Technology, P.O. BOX 20, FIN-53851 Lappeenranta, {Jouni.Ikonen, Jari.Porras}@lut.fi
2 Tampere University of Technology, P.O. BOX 553, FIN-33101 Tampere, Jarmo.Harju@cs.tut.fi

Keywords: Distributed simulation, cluster of workstations, performance analysis

Abstract

Simulation is an important tool in planning. Large simulations require a lot of memory and processing time. In many cases the long simulation times reduce the usability of simulation. In this paper, aspects affecting distributed simulation have been studied, including hardware, software and network related issues. Experiments have been carried out with a distributed simulator, Diworse. Simulation experiments are conducted to show the importance of the used network parameters and the importance of consideration of the used operating system. The achieved results indicate that distributed simulation can be an effective method for reducing the simulation time.

Introduction

Simulation has become an important modeling tool in many areas. It has been used efficiently in meteorology, telecommunications, physics, etc. Unfortunately, the simulation models are getting larger and, as the result, the time required by the simulation is growing. Multiprocessor computers and distributed processing environments have successfully been used to reduce the growing simulation time [4]. In this paper distributed simulation in a workstation environment has been studied. Special emphasis has been focused on finding the elements that affect to the distributed simulation, and the effects of each element.
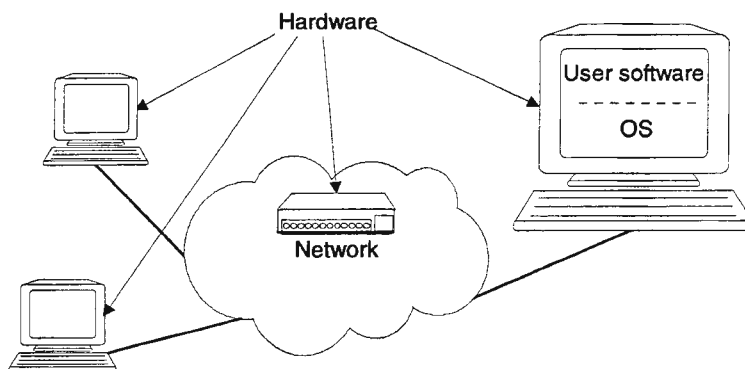
Figure 1. Elements of distributed simulation

Distributed simulation is affected by all elements of the distributed system, e.g., software, hardware and communication network. The software can be divided into the user and the system specific parts. The user specific part consists mainly of the distribution algorithm and other user controllable software elements. The distribution algorithm takes care of the correct simulation, synchronization and deadlocks. Several distributed algorithms have been proposed in literature, and their comprehensive examination can be found from [4]. The system specific part contains operating system, device drivers, internal buffers, etc. Regular users can not normally access the system specific parts. The hardware part consists of the used processing and transmission equipment. The used hardware affects everything, but is often a compromise as simulation may not be the main application run in the system. In this paper the simulation is performed in a cluster of workstations. The communication network is used for interworkstation communication and is usually considered the main weakness of the distributed environments [2]. The communication network part

contains all the elements that take part in transmitting a message from one workstation to another. Parameters of the network part could be the network type, connection elements, e.g. hubs and switches, etc.

In this paper the analysis of distributed simulation is approached by using a previously implemented distributed workstation environment, Di*worse*, and a GSM network application [3]. Simulation experiments are performed by using Sun IPX workstations with a Solaris 2.4 operating system. Workstations are connected through an ordinary ethernet network using a hub or a switch. Communication between simulation processes is implemented with sockets and the TCP/IP protocol. A modified Chandy-Misra algorithm [7] is used as a distributed simulation algorithm.

The effect of different network equipment is tested with a switch and a hub in the Ethernet network. Various TCP parameters and their effect have been tested in order to minimize the simulation time. These parameters include send and receive buffer sizes and the Nagle algorithm [5]. The number of used workstations has been varied between 1 and 9. Results from the simulations and network tests are presented. The results indicate that good speedups can be achieved with distributed simulation but some improvements can still be gained by exploiting the used environment to its full extent.

Distributed simulation of a GSM application

Distributed simulation has been researched at the Lappeenranta University of Technology since 1994. During the research a distributed simulation environment, Di*worse*, for workstation networks has been developed. The simulation environment is based on a conservative Chandy-Misra algorithm [1], which has been developed further [7] to be more suitable for simulation of cellular telecommunication systems. Distributed simulation requires that a problem is divided into logical parts. In the Chandy-Misra -algorithm these parts are called logical processes (LPs). They communicate with each other through predefined channels.

The distributed workstation environment, Diworse, is based on the Manager-Agent concept. This is illustrated in Figure 2. Agent processes contain the logical parts of the simulation model and take care of the distributed simulation. The Manager process is used for controlling the start, the end and the error situations. It does not have an effect on the simulation, but it synchronizes the logical processes at the beginning of the simulation. All the Agent processes are executed in separate workstations whereas the Manager process can be executed in a separate workstation or in the same workstation as one of the Agents. Both Manager and Agent processes contain a scheduler which takes care of the communication between physical processes in different workstations and simulation order of logical processes inside the workstation. Communication between workstations has been implemented by using the TCP protocol, because reliable message delivery has to be guaranteed.
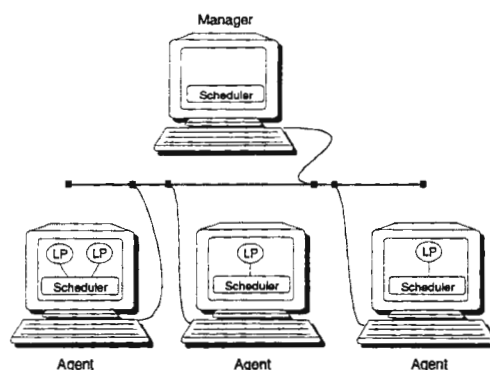
Figure 2. Di*worse*

Logical processes that are located on different workstations can be executed independently, but they may have an effect on each other. Therefore the logical processes must make sure that events are simulated where and when they should. This problem is solved by using the Siml∞p algorithm [7], which is a modification of the Chandy-Misra algorithm [1]. This algorithm improves the performance of distributed simulation.

The distributed simulation environment has been analyzed with a GSM application. In the results presented in this paper the GSM system has been simulated for 900 seconds. Call load per transmitter has been varied from 100 to 800 calls per transmitter per hour. The average length of a call is 60 seconds. The simulation area consists of 27 transmitters. Each transmitter is represented by an LP. Mobiles within the service area of a transmitter are represented by an LP. Therefore, the simulation area of 27 transmitters consist of 54 LPs. Simulator implements both handover and power control procedures. Both of these procedures are realized by carrier per interference calculation. These calculations generate most of the computational demands of the application.

Analysis of the distributed simulation

Distributed simulation is affected by all the elements of the distributed system. It is hard to clearly divide those elements into separate categories as they affect to and overlap each other. In this paper a crude division of distributed environment into software, network and hardware parts has been used.

Software

In this paper software is divided into user and system specific parts. Distributed simulation algorithm and some TCP parameters are clearly in the user specific part whereas operating system belongs to the system specific part. Some parts of the software, e.g. load balancing algorithm, the effect of other users and applications, can not be explicitly placed into either category.

Figure 3 presents simulation times of the GSM application on Diw*orse* software. These results show that distributed simulation performs much faster than the sequential simulation. Sequential simulation is presented as a dotted line in the figure. Distributed simulations are marked with number of workstations used in the simulation. Results gained with distributed simulation software are also dependent on the problem to be solved. Problems to be solved have different degrees of parallel potential [8], which means that only some parts of the problem can be solved in parallel and that some serial parts of the problem can not be parallelized. This affects to the speedups that can be achieved with the parallel simulation.
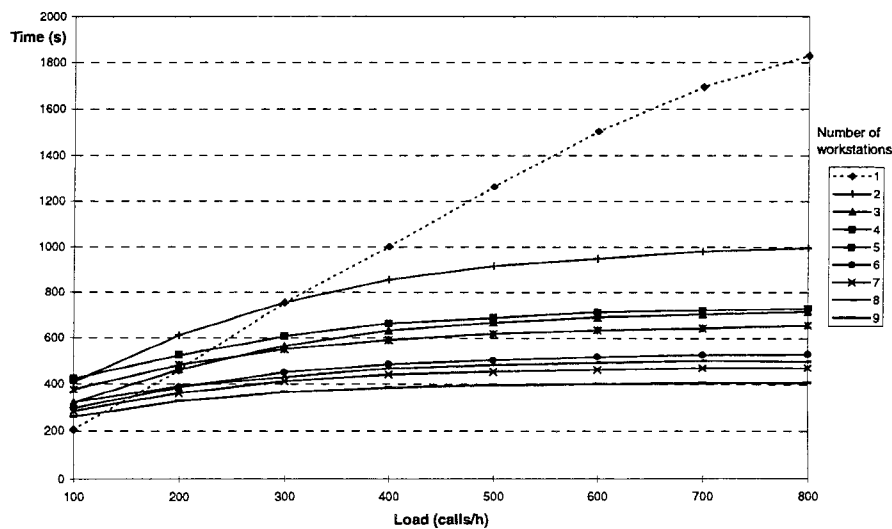
Figure 3. Simulation times of a GSM application

From Figure 3 it can be observed that simulation times are gathered into groups, e.g. simulation times with 7 and 8 processors are quite close to each other. This is due to the fact that LPs can not be divided equally depending on the number of workstations used. Therefore, the workstation with most work dictates the simulation speed. Load balancing between workstations has to be taken into consideration in order to get good results. In Diworse a static load balancing is used as the user has to predefine which LPs are executed in which workstation. Even if load balancing is used, the load can not always be divided perfectly as previous results show.

The effect of send and receive buffers was also studied. Tests were performed by transmitting different sized packets between two Sun IPX workstations and changing buffer sizes used in Diworse. The change in buffer sizes did not affect the execution times in these tests. No long queues are generated into any of the connections between hosts in simulation, so buffer size does not really affect the performance.

Normally TCP protocol uses the Nagle algorithm [5] to collect small packets together. This improves network usage as combined packets have only one set of headers instead of many. Unfortunately, there is a time period during which these packets are collected and this delay is not desirable in distributed simulation. In Diworse distributed processes have to send messages to each other and it is important to send all messages as soon as possible, as a process may need messages from other processes to proceed. The effect of the NODELAY parameter in the TCP/IP, which controls whether Nagle algorithm is used or not, was studied. Figure 4 presents a comparison of simulation times with and without the Nagle algorithm. It can be easily seen that the use of Nagle algorithm degrades the simulation times and should not be used in distributed simulation.
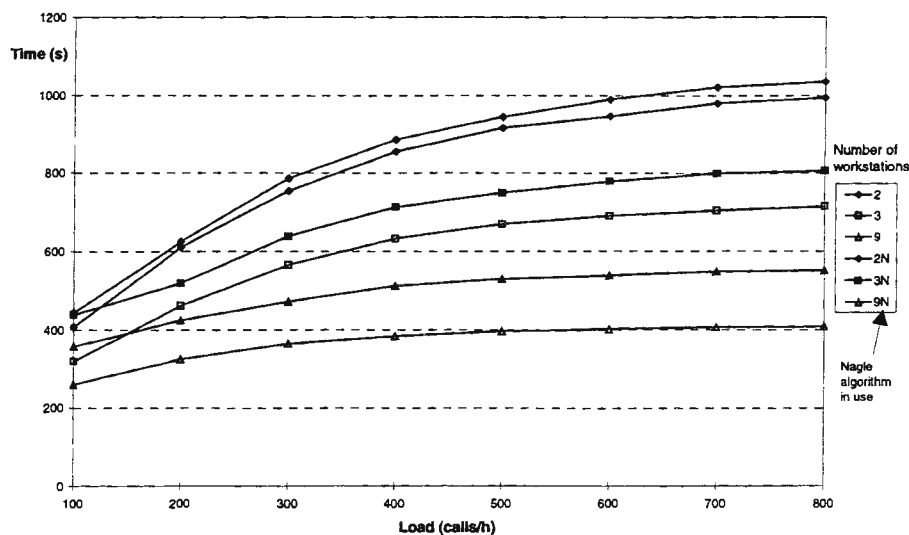
Figure 4. Effect of Nagle algorithm in simulations

The operating system has many parameters, which have an effect on the simulation, e.g. other users, tasks, interrupts, device drivers and internal buffers. User can not usually directly affect many aspects of the operating system, but there are some indirect ways. In early versions of Diworse each logical process (LP) was executed in its own process. This caused a number of problems as a large number of LPs would run in a single workstation. Communication between LPs in one workstation was almost as complicated as communicating trough the network as synchronization and simulation messages had to be exchanged between processes. The large number of processes also caused a lot of context switches and disk I/O. These problems were overcome by combining separate processes into a single process. LPs in a workstation are collected into one process which schedules the execution of LPs and performs the communication to other workstations (see Figure 2). Similar results of combined processes are also reported in [9].

Network equipment that connects workstations has its own impact. Experiments were performed with an Ethernet hub and a switch. The used hub is 8 port CentreCOM MR820TR and the switch is 3Com 12 port SuperStack II switch 3000. The hub is a relatively inexpensive 10 Mbps Ethernet network device, whereas the used switch is a more expensive device with capabilities to handle 100 Mbps connections. The switch uses a store-and-forward method for transferring packets from one port to another.

Transmission tests of different sized packets between two Sun IPX workstations resulted in graphs presented in the Figure 5. It can be noted that the hub gets faster transmission times in these experiments. However, when a hub is used multiple workstations transmitting concurrently might cause collisions in the network and retransmission is required. The use of switch prevents collisions, but suffers from the buffering of packets. It should be noted that there few different type of switches, which use different policy in transferring a packet from a port to another [6]. Closer look to transmission time figure reveals an interesting result. As the packet size exceeds 64 bytes a clear step in the transmission time can be observed. This is assumed to be due to internal buffers of the network interface cards or due to the operating system's internal buffers. This kind of result was not detected in newer Sun workstations.
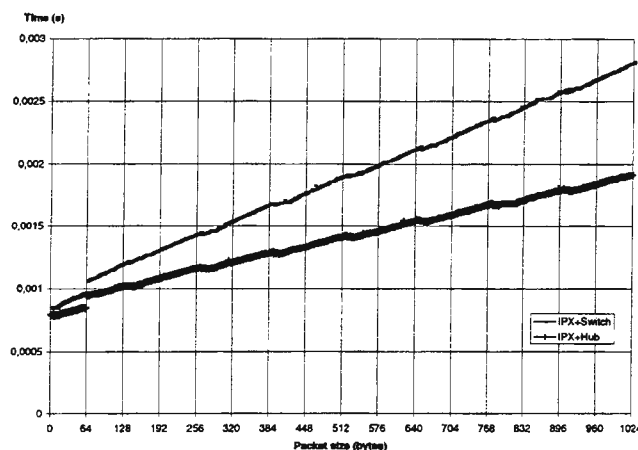
Figure 5. Hub and switch compared

Experiments with the hub and the switch were not impressive when they were performed in Diw*orse* - no difference in simulation times could be detected. One reason for this is that packets used in simulation are relatively small and the difference in transmission times between the hub and the switch is small. Also the retransmissions required in the hub were avoided in the switch. These results lead to the conclusion that the selected network equipment has less effect on the simulation time of Diw*orse* than initially was expected.

Improvements in processing power have a direct impact on the performance of distributed simulation. If the used equipment is heterogeneous the load must be divided according to performance of the equipment. The used hardware can have tremendous effects, but often users have to make compromises as they can not always upgrade their equipment to the best available. No results with different workstations are presented, but experiments with Sun Ultra 1 workstations indicate that Diw*orse* scales nicely to faster computing equipment.

Conclusions

Distributed simulation is a promising way to improve simulation performance. Unfortunately, writing a simulation application for a distributed environment tends to be much harder than for a non-distributed environment. This is due to the fact that there are more aspects to be considered in distributed simulation than in a sequential. Optimization of the used transmission parameters was found to be important. The

results indicate that simulation times can be improved considerably. Consideration of the operating system behavior was found to be significant and should not be overlooked in the planning phase.

References

[1]   Chandy K and Misra J.: *Distributed Simulation: A Case Study in Design and Verification of Distributed Programs,* IEEE Transactions on Software Engineering, 1979, pp. 440-452.

[2]   Dhabaleswar K. O., Lionel M. N.: *Special Issue on Workstation Clusters ands Network-Based Computing - Guest Editor's Introduction,* Journal of Parallel and Distributed Computing 40, 1997, pp. 1-3.

[3]   Hara V., Harju J., Ikonen J. and Porras J.: *Application of Distributed Workstation Environment to the Parallel Simulation of Mobile Networks,* Proceedings of the 2nd Workshop on Personal Wireless Communications, Frankfurt am Main, Germany, 1996, pp. 127-136.

[4]   Fujimoto R. M.: *Parallel Discrete Event Simulation,* Communications of the ACM, Vol. 33, No. 19, October 1990, pp. 30-53.

[5]   Nagle, J.: *Congestion Control in IP/TCP Internetworks,* RFC 896.

[6]   Parnell, T.: *LAN Times Guide to Building High-Speed Networks,* McGraw-Hill, 1996.

[7]   Porras J., Hara V., Harju J., Ikonen J.: *Improving the Performance of the Chandy-Misra Parallel Simulation Algorithm in a Distributed Workstation Environment,* The 1997 Summer Computer Simulation Conference, Arlington, 1997, pp. 657-662.

[8]   Salmi, M., Harju J. and Porras J.: *Computing the parallel potential of event-oriented simulation applications,* Proceedings of the ESS'94, pp. 153-157.

[9]   Simonovich, D.: *Merging Processes in Parallel Discrete-Event Simulation,* 7th European Simulation Symposium, October 1995, pp. 61-64.

Simulating Details on Demand Using Variable-Resolution Modeling

Miroslav Liška, Peter Meòhart, Jiøí Šafaøík
Slovak University of Technology, 812 19 Bratislava, Slovakia
email: liska@dcs.elf.stuba.sk

Abstract

This paper addresses the possibility of engaging the variable-resolution modeling paradigm for solving the complexity problem of simulating large systems. The proposed approach exploits a frequent phenomenon that the level of required details varies in space and time during a particular simulation run. Very often, only a selected part of the model needs to be simulated at the high-resolution level, whereas the remainder can be simulated with lower resolution, because its influence over the focused submodel is minimal. The paper also covers implementation issues for the simulator with a user-driven resolution zooming based on the variable-resolution modeling.

Introduction

Along with growing need to simulate large and complex systems consisting of thousands of mutually interacting entities, the lack of available simulation tools capable to simulate these systems (in acceptable time) is being increased, too. Most often, simulation tools and languages try to cope with this problem by distributing the model among multiple simulation processors [2] or by running clones of the same model in parallel or distributed environment [6]. However, none of those methods is satisfactory for simulating huge systems. Communication overhead caused by interactions among thousands of simulation entities in distributed environment will considerably cripple performance of the whole simulation run. On the other hand, simulation of the model clones will not be usable when the aim of the simulation is not to collect statistical data, but rather to interact with the user.

The simulation models consisting of vast amounts of entities owe their complexity to the modeling process, in which they are designed at a very precise, high-resolution level. However - only very seldom the high-resolution entities (HREs) must be involved throughout the whole model, especially if they are located in a submodel that is currently not of a great interest. Instead, the parts of the model, which are not required to run with the top-precision during the given simulation run, or the entities currently not being observed by the user (or by reporting module) can be simulated at a lower-level resolution.

This paper proposes a *variable-resolution modeling* (VRM) paradigm as a solution to the problem of simulating large and complex system. The simulation processor initially executes only a simulation of a coarse model – designed at a low-resolution level. Nevertheless, the observation stations joined to the simulation can freely browse through the resolution level hierarchy, as if the simulation were running at a multiple levels of resolution. When an observation at a precise level is required, the appropriate low-resolution entities (LREs) are disaggregated into the constituent HREs, working at a lower level of abstraction.[1] When the HREs are not needed anymore, they are aggregated back to the LRE.

A good example of a system where this kind of simulation appears to be a better solution than distributed or parallel simulation is the battlefield simulation, allowing multiple observation clients to join the battle. The coarse model consists of entities representing armed troops, which are acting as a single objects – statistically counting number of soldiers that were killed or injured depending on the state of troops mutually fighting. However, if an observer station requires inspecting a troop in details, in order to see the members of the troop individually, the focused entity is disaggregated into HREs representing soldiers.

To enable usage of such an approach, the model must be designed at several levels of resolution, so the user can change readily the resolution of a desired part of the simulation model. We cannot aspire to continuously variable resolution analogous to the zooming of a camera, but we can aspire to making discrete changes in resolution that are easy to follow and, within limits, valid [1].

This paper tries to answer the problem of aggregation-disaggregation of the entities by making LREs and HREs co-existing in the time. When a view on the high-resolution simulation is required, a *hook* entity (a touple of HREs intended to replace functionality of a LRE(s) at a lower level of abstraction) is involved into the simulation. The involvement is accomplished by redirecting transaction channels used by LRE to HREs, so they can process the incoming transactions at a more detailed level and to produce output with a higher precision. Moreover, during the

---

[1] Levels of resolution and levels of abstraction are inversely related: high resolution means low level of abstraction, and low-resolution means high level of abstraction

duration of HREs involvement, the low-level state variables of a LRE are reflecting the precise values computed by HREs. The techniques of redirecting transaction channels – *hooking*, and attribute reflection - *ghosting* are described in the following sections. Also, the mapping consistency across levels of resolution is discussed and hook design issues are addressed.

Hooking transactions

In the approach proposed by this paper, the simulation entities represent logical processes, able to mutually communicate via transaction channels. The channels are unidirectional and are created by connecting input and output

Figure 1 Communicating entities
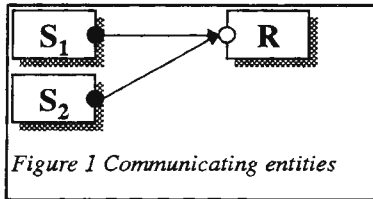
ports of entities supposed to communicate. The interaction[1] between entities is realized as sending a chunk of information (that we call the *transaction*) using the output port of the sender-entity, what causes receipt of the transaction in the receiver-entity's input port. This concept implies that an output port can be attached only to one input port, but any number of transaction channels can be bound with a single input port.

When a LRE is to be changed into HRE during simulation run, the usual approach is to redirect the transaction channels feeding the LRE to the HRE input ports. Thus, the HRE is able to process transactions instead LRE, at a more detailed level (e.g. distribute them among its disaggregated parts, calculate the result with higher precision, or process the transaction with respect to information the LRE does not have). Moreover, the output from the HRE must be usually sent to the entity attached to the output ports of the disaggregated LRE, to

supply the receiving entity with the more detailed data. The LRE (marked X on all following figures) can still remain active, and continue sending some transactions, which are not supplied by HRE. The HRE therefore **joins** the transaction channel by simply reusing the channel referenced by the given output port of the LRE. (The HRE is marked in the picture as **join** – according to name of the technique used. This convention is used to denote HREs in all following pictures and serves also as a keyword in simulation language we use to describe variable-resolution models).
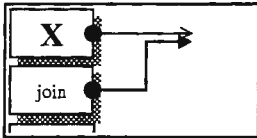
If the LRE is not allowed to continue talking on its output port anymore (after involving HRE), another technique comes to use – the HRE joins the channel, but **strips** off the LRE. As the LRE becomes dumb, all its responsibilities on the stripped channel must be taken over by the HRE. The decision whether the LRE should remain active on the channel or not must be made by the model designer. If the LRE should remain only partly active (i.e. some transactions must not be sent anymore, but some are still valid), then another redirection technique can come to help – the output port of the LRE is redirected to the newly-created transaction channel. An auxiliary entity, listening this channel, distinguishes which transactions sent by X can be re-transmitted further, and which not. This **redirect** technique can be of course applied to another set of problems, too.

Another problem is the input ports redirection – i.e. redirection of the transactions incoming to the LRE. It would be easy, if only one transaction channel could be attached to one input port. However, the situation described above requires more complex handling. All transactions incoming onto the input port of the LRE are to be re-routed to HRE, so all channels joined to the LRE input port must be redirected. This task can be basically accomplished in the following ways:

- All the output ports in the model are checked. If the examined port is connected to the LRE input port, which is to be 'stolen' by HRE, then the channel is redirected.

- The input port keeps a list of the attached output ports. When being redirected, all output ports from the list are changed, and the list is moved to the port accepting the redirected channels.

- The output ports are connected to the receiving input port indirectly – via so-called *port redirector*. When a re-routing is required, only the redirector's input port reference is changed, but the output ports still continue to use the same redirector.

Figure 2 Port redirector usage

---
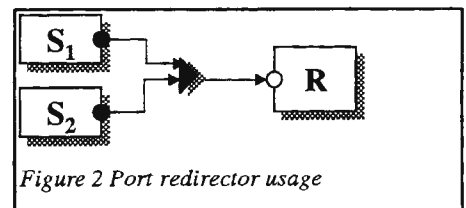
[1] The *interaction* term used in this paper does not describe the same concept as interaction used in HLA. It always refers to communication between two entities performed by sending a transaction one to another.

For the implementation we chose the last method as the most suitable, considering mainly the speed of redirection, which is much higher than in other approaches. Moreover, this method does not cause the memory fragmentation in such extent as the alternative method using lists of the connected output ports. The disadvantage of the chosen method is probably only in the time overhead caused by indirect port addressing used, but the speed is still comparable with the approaches not using the port redirector.
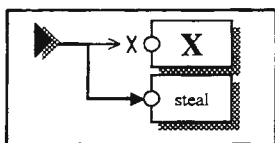
If the input data for the entity **X** are not to be stolen (and processed by the HRE), but rather to be enriched by additional data from the HRE, then the **join** technique comes again to use. Although the name and the basic principle is the same as the **join** technique described above, but parameters passed to the join function reference the transaction channel to be joined rather by the input port connected to the mouth of the channel.

Having port redirector, many other different redirection techniques may be easily applied. The one, already described above, where the input data are 'stolen' from the LRE and redirected to the input port of the HRE is called **steal**.
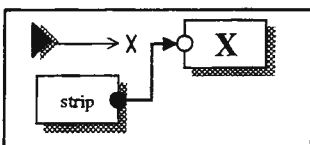
Likewise, the already described **strip** technique can be taken from the 'opposite' end, too. If all input data to the entity are to be abandoned and replaced by new input data generated by a HRE – then the old channel leading to **X** is stripped (making its mouth blind), and a new channel is created between the given entities. Responsibility for the correct behavior in the case of sending a transaction over a blind channel is up to the simulator engine.

Finally, the **redirect** technique can be extended, so that not only transactions outgoing from the given output port are re-routed, but instead it is possible to **redirect all** transactions entering the input port attached to the output port via port redirector. Thus, the re-routing takes place after the redirector, not before.
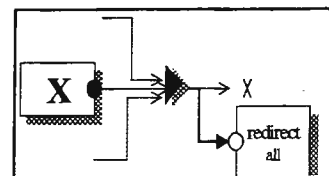
The logic of all transaction redirection techniques can be described more formally:

```
OP = <O, &PR>

PR = <R, &IP>

IP = <I, &PR>

(PR.IP ≠ null) ⟹ (PR.IP.PR ≡ PR)
```

An output port OP is described by a couple of values, one containing application-specific attributes of the port (O), and the second containing the reference to the port-redirector used (PR). The same way, description of a port redirector includes a set of auxiliary attributes of the redirector, and the reference to the input port fed by the channel. Input port contains a backward reference to the port-redirector, so some redirection techniques (*redirect, join(2), steal*) can be easily implemented. The input port referenced by a port-redirector must contain reference to the same redirector it is referenced by.

The implementation itself is given in the following table (using a simplified C++ language notation). The letter **X** denotes a LRE entity, originally working in the coarse model, and **Y** is a HRE entity applying a redirection technique. Input and output ports of the entities are named **I** and **O**, respectively.

Simulation language expression	*C++ code translation*
Y.O **join** X.O	Y.O.PR = X.O.PR;
Y.O **strip** X.O	Y.O.PR = X.O.PR; X.O.PR = null;
Y.I **redirect** X.O	Y.I.PR = new PR; Y.I.PR.IP = Y.I; X.O.PR = Y.I.PR;
Y.O **join** X.I	Y.O.PR = X.I.PR;

```
Y.O strip X.I              X.I.PR.IP = null;
                           Y.O.PR = new PR;
                           Y.O.PR.IP = X.I;
                           X.I.PR = Y.O.PR;

Y.I steal X.I              X.I.PR.IP = Y.I;
                           Y.I.PR = X.I.PR;
                           X.I.PR = null;

Y.I redirect all X.O       X.O.PR.IP.PR = null;
                           X.O.PR.IP = Y.I;
                           Y.I.PR = X.O.PR;
```

Whatever straightforward the transaction redirection concept might look like, it hides potential drawbacks. Consider the following situation: The LRE entity **A**, with input port A.I can be hooked by a HRE entity **B**, stealing input data from A.I and redirecting them to its own input port: B.I:

<div align="center">
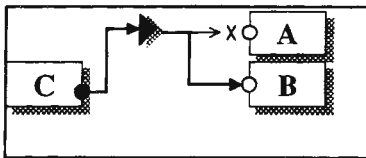
B.I **steal** A.I

</div>

However, the model contains also another HRE - **C**, whose intended function is to enrich the data input for the A.I by additional information flowing from its output port C.O:

<div align="center">

C.O **join** A.I

</div>

Involvement and the existence of both hooks are independent. The user may demand involving only one of the HRE entities, or can make them run simultaneously. If **C** entity is involved first, it bears no problem, but if **B** entity first 'steals' the input and afterwards the user requires start of **C** entity, he or she founds that no channel leading to A.I port can be joined anymore - it has been stolen.

The simulator can cope with such a situation in the following ways:

- The request of the **C** entity to join the channel leading to A.I simply fails. However, this solution makes the simulation user confused - if the hook **C** is started first and then the entity **B** is involved, the simulation works correctly, so the user expects the same (or nearly the same) behavior as would be produced if the hooks were started in reverse order.

- A new channel is created between ports C.O and A.I, so **C** can send its additional data to A.I, as it is supposed to. The flaw is again in the behavior, which is different when compared with the reverse situation described above.

- Additional data from the C.O are sent to the B.I instead of A.I, taking into consideration that B.I works now as a replacement of A.I. The simulation model achieves the same state, as it would be when starting the hooks in the reverse-order.

The last approach sounds the most reasonable, and it was taken in the implementation of the hook technique as the only suitable. However, there still remains a problem of obtaining the port, that should be joined instead of A.I - i.e. the problem of finding out that B.I should be joined instead of A.I. The ports are not designed so they could remember who has 'stolen' the channels attached to them - such approach implies cascade searching of all ports stealing input from each other. The *ghosting* technique, described in the following section solves this problem more elegantly.

Ghosting attributes

The state of each entity is represented by a set of its attributes (e.g. the *NetworkServer* entity can have attributes like *Utilization, UsersPeek, OpenConectionsCnt*, etc). A subset of the attributes can be observed by a simulation user in order to create reports, charts, animations and for other statistical and visualization purposes. Part of the attributes remains hidden for the user– it covers internal implementation of the entity. Some attributes can be set by the user, in order to parameterize the entity. Usually, the attributes are stored in the object representing the entity, although in several situations they are not stored physically in the memory, but must be computed from other data instead.

An approach to the generic description of simulation entities and their attributes described in [4] and [5] helps to access arbitrary entity/attribute even from external applications, without the need to write specific code for each attribute. This technique was successfully used to create a simulator with an open interface [7], and to design a tool for constructing user interfaces for simulation [8].

Name2Offset		ParamPtr	ParameterTable(*simplified*)		
Name	**Index**	**Ptr**	**Type**	**Flags**	**Location**
			without hook		
SoldCnt			**ptInt**	**pfRegular**	**&SoldCnt**

			Type	**Flags**	**Location**
	added by hook				
			ptInt	**ptFunction**	**&GetSoldCnt**

Figure 3 Attribute ghosting mechanism
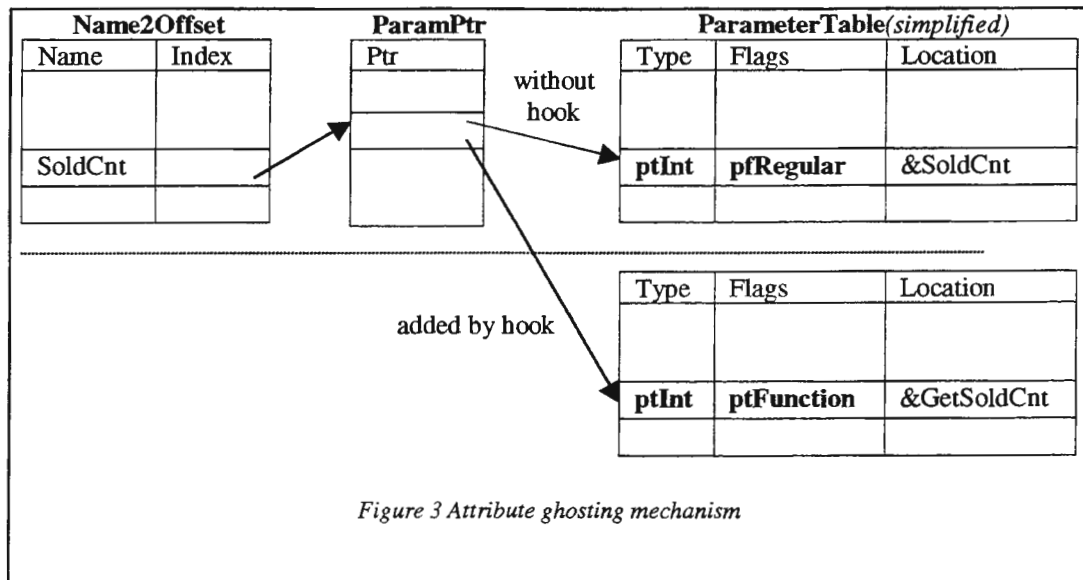
The principle of the entity/attribute description technique is based on creating a table (called **parameter table**) for each entity class, containing name, type and location of every attribute of the entity. The location of the attribute is given in terms of relative distance (offset) of the attribute from the start of the entity object envelope. In the case the attribute is a static member of the entity class, the location value points to the absolute address of the described attribute. For computed attributes, the location field contains the address of the function used to calculate the value of the attribute.

When the hooks paradigm (or another approach to the VRM) is engaged, the original parameter tables technique fails. When a LRE is disaggregated into its constituent HREs, the low-resolution attributes must be changed into high-resolution ones, which must be usually computed from its multiple parts, located in HREs. For example, a LRE representing a division has an attribute *SoldiersCnt*, which is stored in the entity as an integral-type class member. After disaggregation of the division into separate HREs representing battalions, the count of soldiers must be computed rather as a summing of the soldiers in each battalion.

Thus, the disaggregation on the entities brings the need of 'ghosting' attributes – the technique, where the attribute of the entity is owned (or also calculated) by another entity. The *SoldiersCnt* attribute of the division must be computed by the hook, whenever its value is needed. The low-resolution attribute cannot be just 'forgotten' during the hook duration, primarily because the lifetime of the hook is (usually) unlimited. Moreover, the user (or the external application) observing the attribute must be correctly informed of the state of the examined entity, and the existence of the hook taking ownership of the attribute must be for them transparent.

The parameter tables are class-based, i.e. there is only one table per entity class, because all entities based on the same class are expected to have equal set of attributes. However, if a LRE entity is disaggregated – it changes some of its attributes (its type, location or owner) so the description stored in its parameter table is not valid anymore. The parameter table approach was hence extended in order to allow description of dynamic changes in individual entities.

The simplest solution is to create separate parameter table for each entity instance, but such an approach would lead to waste of memory resources, mainly in the case of large and complex systems, where the simulation model consists of thousands of entities. More elegant answer to this problem is alike to the approach used in transaction hooking.

The description record in the parameter table is not to be accessed directly, but rather via **ParamPtr** table, containing pointers (attribute redirectors) to the current description of the given attribute. Names of the attributes are not stored in the **ParameterTable** anymore, but are moved into a hash table **Name2Offset**, which contains associations between attribute names and the references to the attribute redirectors. The reference is given as an index to the to the **ParamPtr** table[1]. A hook that supports attribute ghosting for a LRE changes the appropriate attribute redirector to reference the high-resolution attribute during its initialization procedure.

Using this approach, there is only one instance of the **ParameterTable** and **Name2Offset** tables per entity class. Along them, there is one separate instance of **ParamPtr** table for each hookable entity, and one instance reused by all non-hookable entities. The entity is said to be *hookable*, if there is at least one hook in the model supposed to hook

---

[1] In real implementation, however, the technique is accelerated by using offset of the given attribute redirector, instead of its index.

43

transactions or ghost attributes of the given entity. Decision, whether the entity is hookable or not is made after the simulator reads the simulation model description, or can be determined during the run-time, when an attempt to hook an entity is made. The latter approach spares the memory resources, but it is slower, because it requires heap operations and various run-time checks and its usage is discouraged.

Attribute ghosting technique gives a solution to the problem posed in the section describing transaction hooking – to obtain the HRE port that steals transactions from a LRE port. As the ports have their own record in the **ParameterTable**, the ports can be ghosted too. When the port is stolen, the appropriate pointer in the **ParamPtr** table is changed to point to a HRE port instead of original LRE port. When a request comes to obtain a LRE port description, the HRE port description is retrieved instead.

Hooks lifetime

As was already said before, HRE entities are involved into the simulation only 'when needed'. The need may be declared directly, by requiring the involvement of the specified HRE(s) via specific interface functions provided by simulator. Indirect demand, however, is more often used – if the user (ext. application, reporting module, etc.) requires observing an attribute of a HRE, the HRE is automatically joined to the simulation model, using hooking and ghosting techniques described above. This process is transparent to the user - he does not need to know that the entity he observes works only temporarily, and will be again removed when not necessary anymore. The decision of the hook join/disjoin is made according to the reference counters, assigned to every hook existing in the model.

The hooks often require for their proper functionality other hooks to be involved into the simulation together with them. For example, the HRE entity processing transactions at a very-precise level needs another HRE entity working in a distant part of the simulation model, to enrich particular transaction channels by specific data. The simple grouping of both entities into one hook will not do, because the latter can work also standalone. Such hooks therefore declare their demand in the description of the simulation model (or in the hook definition). Hooks can get started indirectly also if the hook requires values of high-resolution attributes of another hook, not yet involved into the simulation.

Multiple hooks involvement may occur also in the case the **hooksets** (sets of hook) are used. These allow simulation user (or model designer) to create sets of otherwise independent hooks, that need to be started as a group very often. The hooksets can be involved into the simulation by one command, simplifying (and speeding-up) this way the communication between user and the simulator.

Mapping consistency

When a LRE is going to be disaggregated into HREs, the hook containing the HREs must properly initialize the state of all HREs. Their state must be *consistent* with that one of LRE being disaggregated. In other words, the state of HREs should be the same as it would be if the high-resolution simulation were running since the start of the simulation. The so-called *mapping consistency* is supposed work also in the reverse direction – i.e. when the HREs are going to be aggregated back to the LRE, the hook must update the LRE into consistent state.

More formally, let

$S_H(t)$ denotes the state of a hook working at a high-resolution level in the time t

$S_L(t)$ the state of a LRE in the time t

$d(S_L(t))$ disaggregation function

$a(S_H(t))$ aggregation function

$\varphi_L(S_L(t), \Delta t)$ state-transition function at low-resolution

$\varphi_H(S_H(t), \Delta t)$ state-transition function at high-resolution

where

$$d(S_L(t)) = S_H(t) \text{ and} \tag{1}$$

$$a(S_H(t)) = S_L(t) \tag{2}$$

By [1] the models are considered to be *consistent in the aggregate* if

44

$$a(\varphi_H(S_H(t),\Delta t)) = \varphi_L(a(S_H(t)),\Delta t) \tag{3}$$

and to have a *complete consistency* if

$$\varphi_H(S_H(t),\Delta t) = d(\varphi_L(a(S_H(t)),\Delta t)) \tag{4}$$

In the real implementation, though, even the aggregation/disaggregation operations performed immediately one after another could cause problems. By aggregating HREs into LRE, some information not available in a low-resolution simulation will be obviously lost. Thus, by disaggregating LRE again, part of the high-resolution state variables must be determined stochastically. We therefore say the hook to be *AD consistent* if

$$d(a(S_H(t))) = S_H(t) \tag{5}$$

The *AD consistency* (5) together with *consistency in aggregate* (3) assures the hook to have *complete consistency* (4). The reverse case of disaggregation/aggregation operations pair usually does not cause implementation problems, because the state of LRE can be deterministically computed from the state of the HREs. Most of hooks are hence *DA consistent*, i.e.

$$a(d(S_L(t))) = S_L(t) \tag{6}$$

For the aggregation and disaggregation functions accomplishing the mapping consistency across levels of resolution cannot be (in most cases) simple described at a simulation language level, the appropriate functionality must be described in the language closer to the machine level instructions, e.g. in C/C++ languages. This fact narrows the potential set of the hook designers – not every user is able to program in C++ or similar language [9]. Moreover, for the security reasons, the simulation users (and even the model designers) should not be allowed to create their own simulation components in the machine level language. This task is still left for the simulation component designer.

We managed to develop a hook description language, which simplifies creating hook component by using C++ syntax extended by new keywords denoting transaction hooking types (strip, redirect, etc.) and supporting description of attribute ghosting. The aggregation/disaggregation functionality is implemented in virtual methods *OnJoin* and *OnDisjoin*, where each hook implements the code guarding mapping consistency.

Conclusions and future work

In this paper we proposed to solve the problem of simulating large and complex systems by using *variable-resolution modeling* paradigm. The VRM approach is commonly used to connect dissimilar simulations (e.g. simulations differing in language, time step, and the representation of the common elements) and is concerned mainly with resolving conceptual and representational differences that arise from multiple levels of resolution in simulations that are joined for a common objective [10].

The hook technique described in this paper can be exploited in many areas of simulation science. In fact, the original idea leading to development of this approach was an issue of research in the field of simulation animation, where the intention was to create a general-purpose animation module suitable for wide set of simulations. Using hooks, the model can be simulated at the resolution it was originally designed, but the parts intended to be animated are simulated at the level that meets animation requirements. For example, in battlefield simulation there is no need to simulate exact movements of each particular soldier, the battalion level is enough. However, when the soldiers are to be animated, the given battalion is disaggregated into its constituent soldiers and simulated thus at the level that can be immediately and precisely animated.

In the future, we expect to make the proposed approach ready for being used in distributed simulations. Also, the process of designing hooks and the VRM model will be revisited, in order to find more general, sophisticated and user-friendly solutions.

Acknowledgement

References

[1]   Davis,P.K., Hillestad,R.(1993), "Families of Models That Cross Levels of Resolution: Issues for Design, Calibration and Management", In *Proceedings of the 1993 Winter Simulation Conference*, ACM, New York, pp.1003-1012.

[2]   Hanáèek,P.(1996), "Problems of Implementation of Virtual Time for Parallel Simulation", In: 18th Int. Workshop *"Advanced Simulation of Systems" ASS'96*, Zábøeh, September 1996, pp.298-303.

[3]   Hluchý,L., Dobrucký,M., Astaloš,J.(1997), "Hybrid solving method for task allocation in distributed systems", In Proceedings of the Seventh International Conference on Artificial Intelligence and Information - *Control Systems of Robots*, ed. by I. Plander, World Scientific 1997, pp.189-201.

[4]   Liška,M., Meòhart,P.(1996a), "Parameter Tables - The Transparent Approach to Accessing the Simulation Model Entities". Proc. *Information Technology Interfaces ITI'96*, Pula, June 1996, pp.439-444.

[5]   Liška,M., Meòhart,P.(1996b): "A Methodology for Describing the Simulation Model Entities", In: Proceedings of Scientific Conference with International Participation *"Electronic Computers and Informatics"*, Kosice-Herlany 26.-27.9.96. Kosice, TU 1996, 294-299.

[6]   Liška,M., Adámy,M., Lališ,I.(1996a): "Goal Seeking in the Parallel and Distributed Simulation", In: 18th Int. Workshop *"Advanced Simulation of Systems" ASS'96*, Zábøeh, September 1996, pp.261-266.

[7]   Liška,M., Adámy,M., Lališ,I.(1996b): "Binding the Simulator with the Spreadsheet", In: Simulation in Industry, 8th *European Simulation Symposium ESS'96*, October 24-26, 1996. Ghent, SCS 1996. , pp.334-338.

[8]   Liška,M., Adámy,M., Lališ,I.(1997): "Object Oriented Interface Builder For Simulator Applications", In: *Information Technology Interfaces ITI'97*, Pula, June 1997, pp.561-566.

[9]   Peringer,P.(1996), "Methods of Simulation Systems Testing", In: 30th Spring International Conference *"Modelling and System Simulation" MOSIS'96*, Krnov, April 1996, pp.175-180.

[10]   Reynolds,P.F.JR., Anand.,N., Sudhir,S.(1997): "Consistency Maintenance in Multiresolution Simulations", In: ACM Transactions on Modelling and Computer Simulation, Vol. 7, No. 3, July 1997, pp.368-392.

[11]   Weatherly,R.M.,Wilson,A.L., Griffin,S. P.(1993), "ALSP-Theory, experience and future directions", In *Proceedings of the 1993 Winter Simulation Conference*, ACM, New York, pp.1068-1072

Trade-Offs Between Parallelism and Convergence Rate of Parallel Solvers

H.X. Lin, E.A.H. Vollebregt [a], T. Yang [b]

[a]Department of Applied Mathematics, Delft University of Technology, P.O. Box 356, 2600 AJ Delft, The Netherlands
Tel: +31-15 78 7229, Fax: +31-15 78 72 09, e-mail: h.x.lin@twi.tudelft.nl

[b]Department of Computer Science, Linköping University
S-581 83, Linköping, Sweden, email: tiaya@portofix.ida.liu.se

Keywords: parallelism, convergence, and scalability

1. Introduction

From supercomputers to small computers multiprocessors and parallel (or distributed) processing are becoming more and more a common practice incorporated at all levels of every computing system. This raises the problem: how to distribute the work among the processors? In other words how to detect and/or create parallelism in a program? This paper discusses the issue of the parallelization of sevral known iterative methods for the solution of large sparse systems of linear equations.

There are many legacy software which have been developed and currently used for simulation of scientific and engineering problems. Therefore parellelization of existing software has been received much attention, and it is important for parallel computing in order to achieve the gerenal acceptance. Many efforts in research of automatic compilation and generation of parallel programs have been carried out and now some results have already been obtained. Besides the research on parallel programming languages and automatic compilation, an equally if not more important issue is designing parallel methods and algorithms.

Sequential methods and algorithms usually optimize the number of operations and/or memmory usuages. However, while many solution methods used in existing software are efficient in terms of convergence (thus operation count) on sequential computers, they are often not very suited for parallel computing because they exhibit few parallelism. In Section 2, we will discuss two such examples and motivate that in the search of efficient parallel algorithms trade-offs between parallelism and convergence rate must be considered. Section 3, will analyze the parallelizability of a number of known iterative solution methods. Then in Section 4, we consider parallel iterative methods based on domain decomposition approaches. We compare the scalability of different methods and discuss the preconditioned AMS-CG method which is a promising scalable method. Finally, some concluding remarks are drawn in Section 5.

2. Examples of existing software

In this section we discuss two examples of solution methods used in existing CFD software packages TRIWAQ and WISH3D respectively.

TRIWAQ is a general program that is used for the simulation of many different shallow water problems [5,6]. It is used by RIKZ, the National Institute for Coast and Sea Management of the Netherlands. The simulation of the flow is governed by the 3D shallow water euations, which are derived from the incompressible Navier Stokes equations by assuming a hydrostatic pressure distribution.

With repect to time, the PDEs are integrated with an Alternating Direction Implicit (ADI) type scheme. For this each time step is split in two halves that each consists of two phases. After the discretization, in each time step a system of equations, $A\Phi = b$ must be solved. In the case that first order upwind discretizations are used for the advective terms, matrix A is penta-diagonal.

An important part of computation in TRIWAQ at each time integration step is the so-called CUE-solver, which is a type of flow-following Gauss-Seidel methods. The CUE method resembles the GS method in the sense that unknowns are updated one by one. However, the order in which unknowns are updated is different. In the GS method the unknowns are updated cyclically in a fixed order, until the approxiamtion is accuarate enough. In the CUE method a simple heuristic is used for adapting the ordering of updates of the unkowns to the flow field; this rule consists of sweeping through grid rows in positive and negative directions and doing the same for sweeps over rows. The algorithm CUE is depicted in pseudo-code Figure 1. In this algorithm i_1 and i_2 count the sweeps over all rows and within rows, and q counts the iterations for each unknown. In the calculation of a new approximation for a grid point in step (1.k) the latest available approxiamtions are used for the unknowns of neighboring grid points. The precise form of the update formula (1.k), hidden in function f, therefore depends on the order in which unknowns are updated. I ffor certain i_1, n and i_2 both M_{inc} and N_{inc} are positive, then the calculation of $\phi_{m,n+1}^{(q)}$ uses value $\phi_{m,n-1}^{i_1 I_{max}}$ of the previous row, $\phi_{m-1,n}^{(q)}$, $\phi_{m+1,n}^{(q-1)}$ from row n and $\phi((i_1 - 1)I_{max})_{m,n+1}$ of the next row. This data dependence is illustrated in Figure 2. It shows the iteration levels of the unknowns that are required for the calculation of $\phi_{m,n}$ at iteration level $q = i_2 + (i_1 - 1)I_{max}$.

Figure 3 compares the number of iterations reguired by different iterative methods for the solution of $A\Phi = b$. The CUE method (with the number of sweeps I_{max} set to 1 per row of the grid) is compared to the Jacobi, Gauss-Seidel, Block-Jacobi (with 16 subgrids arranged in a 4 by 4 subdomains), and the Krylov-subspace methods BiCG, BiCGStab and GMRES. The test problem consists of a finite difference grid of 65×65 points. It can be concluded that Gauss-Seidel type methods can be very effective for advection dominated problems, but only if the advective terms are discretized by an upwind method and if the numbering of the unknowns is adapted to the flow field[1]. More details about the comparisons can be found in [12,13]. The CUE is very efficient (and robust) for both cases with a strong uniform advection flow and with a circulating flow, whereas the GS method has difficulties with the test problem with a circulating flow.

With respect to the parallelizability of these methods, it is another story. The sequence of updating an unknown in the GS and CUE method is fixed to a certain order (e.g. first the unknowns with a lower index number then the unknowns with a higher index number). This implies for the CUE method at the beginning only 1 unknown can be updated, followed by two unknowns of the direct neighboring grid points, and so on. The update of unknowns constrained by the data dependence can be described a wavefront starting a corner point propagating at a speed of one neighboring grid point each step. Therefore, the maximum parallelism is limited to $\sqrt{n}$ on a $\sqrt{n} \times \sqrt{n}$ grid. Moreover, parallelization along the wavefronts leads to fine-grain paral-

---

[1]The convergence of Krylov subspace methods can usually be improved by preconditioning. We have ignored preconditioning here. In a later section we do consider preconditioning for computing the vertical water levels.

$$
\begin{array}{ll}
\text{set initial estimate } \phi^{(0)} & \text{(1.a)} \\
\text{determine dominant } v \text{ direction: } V = \sum_{m,n} v_{m,n}, & \text{(1.b)} \\
\text{determine dominant } u \text{ direction in all rows } n\text{: } U_n = \sum_m u_{m,n} & \text{(1.c)} \\
\text{set } N_{inc} = sign(V), \text{ set } N_{first}, N_{last} & \text{(1.d)} \\
\textbf{for } i_1 = 1, 2, \ldots \textbf{ do} & \text{(1.e)} \\
\quad \textbf{for } n = N_{first} \textbf{ to } N_{last} \textbf{ step } N_{inc} \textbf{ do } (\text{sweep over rows}) & \text{(1.f)} \\
\quad\quad \text{set } M_{inc} = sign(U_n), \text{ set } M_{first}, M_{last} & \text{(1.g)} \\
\quad\quad \textbf{for } i_2 = 1 \textbf{ to } I_{max} \textbf{ do} & \text{(1.h)} \\
\quad\quad\quad q = i_2 + (i_1 - 1)I_{max} & \text{(1.i)} \\
\quad\quad\quad \textbf{for } m = M_{first} \textbf{ to } M_{last} \textbf{ step } M_{inc} \textbf{ do } (\text{sweep in row } n) & \text{(1.j)} \\
\quad\quad\quad\quad \text{update grid point } (m,n): \phi_{mn}^{(q)} = f(\phi^{(q)}, \phi^{(q-1)}, \ldots) & \text{(1.k)} \\
\quad\quad\quad \textbf{end for} & \\
\quad\quad\quad \text{reverse } m \text{ direction } (M_{inc} = -M_{inc}, swap(M_{first}, M_{last})) & \text{(1.l)} \\
\quad\quad \textbf{end for} & \\
\quad \textbf{end for} & \\
\quad \text{reverse } n \text{ direction } (N_{inc} = -N_{inc}, swap(N_{first}, N_{last})) & \text{(1.m)} \\
\textbf{end for} &
\end{array}
$$

Figure 1. Sequential iteration process CUE for the 2D problem.
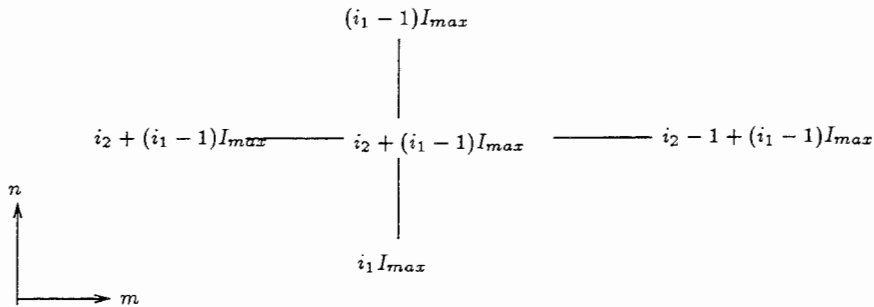
Figure 2. Update stencil for the calculation of $\phi_{m,n}^{(q)}$, $q = i_2 + (i_1 - 1)I_{max}$ (central point). for positive M_{inc} and N_{inc}.
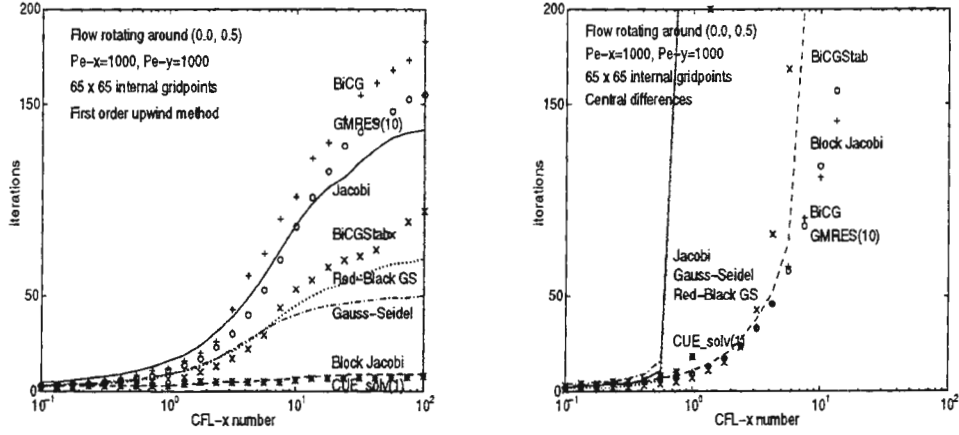
Figure 3. (*left*) Convergence for the test problem with rotating flow, first order upwind, with a flow pattern of a half circle. (*right*) The same test problem, now with derivatives approxiamted by central differences.

lelism and causes a low data locality (for every update operation, which comprises only several basic arithmetic operations, several values of the neighboring grid points must be communicated from other processors). Therefore, the CUE method cannot be parallelized in this way. The Jacobi method has a higher degree of parallelism, it can be easily parallelized, however, its very slow convergence makes even the use of a large number of processors ineffective. The major operations in the Krylov subspace type methods are matrix-vector multiplications. Thier parallelizability is less than the Jacobi method, but better than the GS and CUE methods. However, it is known the calculation of the inner products required by a Krylov method imposes global communication, and this global communication is a bottleneck for the scalability (see e.g. [9,15]. Another difficulty in parallelizing the software package TRIWAQ is that its data structures use much indirect references (e.g. [7,8]), but here we omit the discussion about this aspect.

Next, we consider the CFD software package WISH3D, which is a general program for simulation of heat flow in furnaces, etc. WISH3D is developed by the TNO-TPD Applied Physics Research Institute. It can be used for siumlation of 2D and 3D problems. Characteristics of the solution of the algebraic systems of equations after the discretization of the physical domain are the use of the Gauss-Seidel type methods along the lines (of one coordinate diemension) in a 3D problem and the use of the so-called line-TDMA (Tri-Diagonal Matrix Algorithm) for solving the unknowns along one coordinate direction (line) whithin a plane. For instance, the update of unknowns is performed by first sweeping along the z-planes (i.e. plane with constant z coordinates) and then along the y-planes. Within each plane the variables are updated one line by one line along the x-direction. This corresponds to solving a tridiagonal system of equations of the unknowns on the line by a Gaussian elimination method. The solution of a tridiagonal system by Gaussian elimination is strictly sequential, therefore parallelism of the computation must be sought elsewhere. To update the unknowns of a number lines in parallel corresponds to changing a Gauss-Seidel update sequence to a Jacobi update, consequently the convergence is influenced by that. A very popular parallelization approach is to divide the domain into a number of subdomains and use the original solution method for the global domain (in our case

the line-TDMA) for each subdomain. Again we observe a deteriation of convergence by this approach which is getting worse as the number of subdomains increases. This can be explained by the fact that a line is now cut into several pieces and the boundary values used by a line Gaussian elimination for line-segement of an internal subdomain is less accurate and therefore requiring more iterations to achieve the same accuracy (in other words, in one iteration the information propagtion is limited by the boundaries of a subdomain).

The discussions in this section illustrate that although many solution methods implemented in existing software programs are very efficient on sequential computers, they are less suitable for parallel simulation. Therefore, we need to modify these methods or use other type (new) methods in order to obtain an efficient parallel implementation.

3. Parallelization of some known iterative methods

We compare the parallel speedups and convergence of several known methods in this section. The following methods are considered: Jacobi, SOR (generalized GS), CG and PCG with MILU as preconditioning. We choose as the test problem a simplified form of the shallow water equations.

$$\frac{\partial u}{\partial t} = fv - g\frac{\partial \zeta}{\partial x} + \frac{1}{h^2}\frac{\partial}{\partial \sigma}\left(\mu\frac{\partial u}{\partial \sigma}\right) \tag{1}$$

$$\frac{\partial v}{\partial t} = -fu - g\frac{\partial \zeta}{\partial y} + \frac{1}{h^2}\frac{\partial}{\partial \sigma}\left(\mu\frac{\partial v}{\partial \sigma}\right) \tag{2}$$

$$\frac{\partial \zeta}{\partial t} = -\frac{\partial}{\partial x}\left(h_0\int^1 ud\sigma\right) - \frac{\partial}{\partial y}\left(h_0\int^1 vd\sigma\right) \tag{3}$$

The equations have been transformed in the vertical direction into depth-following coordinates (σ) by the so-called sigma transformation: $\sigma = \frac{\zeta-z}{d+\zeta}$. The boundary conditions at the sea surface ($\sigma=0$) are given by: $\left(\mu\frac{\partial u}{\partial \sigma}\right)_{\sigma=0} = -\frac{h}{\rho}W_f\cos(\phi)$, $\left(v\frac{\partial v}{\partial \sigma}\right)_{\sigma=0} = -\frac{h}{\rho}W_f\sin(\phi)$ with W=windforce and ϕ=direction of windforce. The boundary conditions at the bottom ($\sigma=1$) are given by: $\left(\mu\frac{\partial u}{\partial \sigma}\right)_{\sigma=1} = \frac{gu_d}{C^2}$, $\left(v\frac{\partial v}{\partial \sigma}\right)_{\sigma=1} = \frac{gv_d}{C^2}$ with u_d and v_d velocities near the bottom. In the hor izontal direction a staggered grid is used , see e.g. [14] for more information abou t the advantages of staggered grids. In the vertical direction the grid is divided into a few equidistant layers.

The variable ζ is discretized as Z, which is the water level compared to a plane of reference. The spatial derivatives are discretized using second-order central differences. The time integration is performed by a two-stage time splitting method [2]. The time step is splitted in two stages: $t = n$ to $t = n + 1/2$ and $t = n + 1/2$ to $t = n + 1$. At the first stage the equations describing the ve rtical diffusion are treated implicitly. Each vertical line of grid points corresponds to a separate tridiagonal system of equations.

At the second stage the equations describing the propagation of the surface waves are treated implicitly. This pentadiagonal system of equations describes the water level related to the water level in 4 neighbouring gridpoints:

$$(1 + c_1 + c_2 + c_3 + c_4)Z_{i,j}^{n+1} - c_1 Z_{i+1,j}^{n+1} - c_2 Z_{i-1,j}^{n+1} - c_3 Z_{i,j+1}^{n+1} - c_4 Z_{i,j-1}^{n+1} = Z_{i,j}^{n+1/2}$$

$Z_{i,j}^{n+1}$ is the water level in gridpoint (i,j) at time $t=n+1$. The coefficents c_i are functions depending on the water level Z. This equation can be put in the form:

$$A\left(Z^{n+1}\right)Z^{n+1} = Z^{n+1/2}$$

The equations are first linearized before applying an iterative solution method by:

$$A\left(X^m\right) X^{m+1} = Z^{n+1/2}$$

Here $X^0 = Z^{n+1/2}$ and A is symmetric positiv e definite. These sytems are linear in X^{m+1}. The iterative methods will be used to compute X^{m+1}. Thus there are two iteration processes: The outer iteratio n process updates $A(X^m)$ and continues until $\|X^{m+1} - X^m\| < \varepsilon$ and the inner iteration process solves each X^{m+1}. Both processes are executed until convergence. In our experiments we compare the results of the iterative methods for the entire two iteration processes in the second stage.

For the parallelization of the computation in an iteration, we partition the domain into a number of subdomains (a subcube for a 3D problem) of equal size. The update of the unknowns within a subdomain is assigned to the same processor, in this way data exchange is only required for the grid point at the boundaries between the subdomains (except for the evaluation of the inner products which require global communication). Table 1 shows the speedups and execution times of the different methods when they are used to compute a solution to the same accuracy. It can be seen that the Jacobi method has the highest relative speedup (i.e. the ratio of the execution time the Jacobi method executed at 1 processor and at p processors). This is as we have expected. The fact that the speedup is not equal to p with p processors is mainly caused by the global communication overhead of the inner product required for the convergence check[2]. The SOR method has the second highest relative speedup. The relative speedups of the CG and PCG methods are very poor, they soon drop when the number of processors is increased beyond a certain number. This is beacuse two inner products required in each iteration and the global communication overhead increases quickly with an increasing number of processors.

On the other hand, Table 1 also shows that the total execution time is the lowest with the PCG method, whereas the Jocobi method has the largest execution time. Since eventually the total execution time of a program is the only thing we are interested when using parallel computers to accelerate the simulation, we must not only look at the speedup figure, but also at the convergence rate. This means that the trade-off between parallelism and convergence rate must be considered.

Table 1
Comparisons of sppedups for different methods on the Cray T3D.

Method	Processor grid					Exec. time
	2x1	3x2	4x3	5x4	6x5	(1 processor)
Jacobi	1.94	5.13	8.36	11.0	12.75	15.31 s
SOR	1.92	4.90	7.65	9.67	10.61	2.52 s
CG	1.97	3.57	3.50	3.27	2.84	4.61 s
CG+MILU	1.31	2.72	3.05	2.77	2.36	1.31 s

4. Scalable parallel iterative methods

In this section we first describe the so-called preconditioned AMS-CG method (Approximate Multi-Subdomain Conjugate Gradient method [11,10]), followed by discussion about the exper-

---

[2]In order to reduce this overhead, we already reduce the frequency of convergence to once in every 10 iterations for all the methods.

imental results and scalalibity of the method. The AMS-CG is a domain decomposition based CG method, it is specially designed for (massively) parallel computation. From the previous section we have already observed that the main performance bottleneck of the CG type methods is the global communication required by the two inner products in each iteration. Therefore the main idea in the AMS-CG is to reduce or eliminate the global inner products from the CG method.

4.1. Approximate Multi-Subdomain CG

Starting with a basic form of CG without any optimizations, we can derive the Multi Subdomain CG (MS-CG) method by splitting the global search direction into local search directions, one for each subdomain. The local search direction p_d for subdomain d is a vector with zero entries for variables corresponding to positions outside subdomain d. Each iteration a p_d for each subdomain must be determined. The p_d's are put in the columns of a matrix P. Instead of the scalars α and β in the CG method, α and β are vectors with a separate entry for each subdomain. We require that $p_d^k = z_d^{k-1} + \beta^k P^{k-1}$ is conjugate to all previous p_d^j with $j<k$. This is satisfied by computing β from $C^{k-1}\beta^k = -(Q^{k-1})^T z^{k-1}$. In order to minimize $\left\| x - x^k \right\|_A = \left(x - x^k, b - Ax^k \right)$, α is computed by $C^k\alpha^k = (P^k)^T r^{k-1}$, where $C^k = (Q^k)^T P^k$. It can be proven that the errors $\left\| x - x^k \right\|_A$ are non-increasing [11]. The preconditioned MS-CG method can be derived from PCG.

Algorithm: preconditioned Multi-Subdomain CG

```
x⁰=initial guess, k=0, r⁰=b-Ax⁰
while 'not converged'
   solve Mz_d^k=r_d^k
   k=k+1
   if (k=1)
      for each subdomain d: p_d¹=z_d⁰

   else
      C^{k-1}β^k=-(Q^{k-1})^T z^{k-1}
      for each subdomain d: p_d^k=z_d^{k-1}+β^k p^{k-1}    end
   Q^k=AP^k
   C^k=(Q^k)^T P^k
   C^k α^k=(P^k)^T r^{k-1}
   x^k=x^{k-1}+P^k α^k
   r^k=r^{k-1}-Q^k α^k
end
x=x^k
```

The matrix M has to be a symmetric positive definite preconditioning matrix. $p_d^i=0$ outside subdomain i. The matrix C is symmetric positive definite if the matrix A is. The CG method has the disadvantage that each iteration 2 dot products are computed. Instead of these dot products the MS-CG method has 2 matrix equations with the matrix C which should be solved. The matrix C is symmetrical positive definite in our problem. The idea of the Approximate MS-CG (AMS-CG) method is to approximate the 2 equations with the matrix C by a few Jacobi iterations. The matrix C is relatively small: $l \times l$, where $l = $ number of subdomains. The Jacobi method only requires local communication. The convergence of the Jacobi iterations is not tested, because we don't want to introduce global communication. Usually a small fixed

number of Jacobi iterations (2 to 4) is sufficient. The AMS-CG method requires no global communication, except to check convergence. This convergence check can be done once after a number of iterations.

4.2. Preconditioning

In this section we will describe several preconditioners for the CG-method and the AMS-CG method. ILU preconditioning [4] is well known. It is derived from the LU-decomposition. The LU-decomposition generates full matrices. The ILU(0) algorithm allows no fill in, ie: An entry in L and U is zero whenever the corresponding entry in the matrix A is zero.

Gustaffson [3]proposed the MILU (Modified ILU) method. The compone nts which are disregarded in ILU are added to the main diagonal. This way the row sum of LU is equal to the row sum of A. In our experiments the MILU preconditioner performed much better than the ILU preconditioner. For this reason most experiments with preconditioning were done with MILU.

The ILU and MILU preconditioners have to be uncoupled in order to compute different subdomains in parallel. There are a few possibilities: We can simply disregard the coupling. We can impose boundary conditions on the internal boundaries. Or we can use a Schur complement method. However, we did not use a Schur complement method. This would have resulted in more communication, while we tried to precondition without communication or with very little communication. Consequently we found an increasing number of iterations for an increasing number of subdomains. With a Schur complement method the number of iterations will almost certainly be roughly the same for different numbers of subdomains.

5. Numerical results

The test problem used is that of de Goede [2, p. 63]. We simulated a rectangular basin of 400 km $\times$ 800 km. The bottom is inclined with a depth of 20 m at one end and a depth of 340 m at the other end in x-direction. The grid used is 100×100 points with 5 vertical layers. Simulations are carried out with timesteps equal to: 600 s, 1200 s and 2400s. We simulated a period of 4 hours and used a wind force of 1.0 N/m^2 in x-direction. For the AMS-CG method we took a number of 3 Jacobi iterations.

Convergence

A good indication of the convergence properties of an iterative method is the average number of iterations required per time step. The convergence properties of AMS-CG are just as good as the convergence properties of CG in our experiments. This holds with and without preconditioning. As expected the convergence properties of Jacobi are bad, especialy for larger time steps. The convergence of the SOR method is better than the convergence of CG and AMS-CG if the right relaxation parameter is chosen. The convergence of the MILU-preconditioned CG and AMS-CG are best of all. To give an indication: With a time step of 1200 s and 6×5 subdomains the number of iterations for Jacobi is 1008, for CG/AMS-CG: 202, for SOR: 125, for CG+MILU/AMS-CG+MILU: 55.

Convergence of MILU-preconditioned CG or AMS-CG becomes worse as domain numbers increase. The largest difference is between 1 global domain (1×1) and 2 subdomains (2×1) . We tried to improve this situation, however the number of iterations remained almost the same with overlapping or non-overlapping subdomains and with different internal boundary conditions. The convergence of FSAI-preconditioned AMS-CG was worse than the convergence of MILU-preconditioned AMS-CG. In case of preconditioning we will only present results with MILU using non-overlapping subdomains.

Execution time
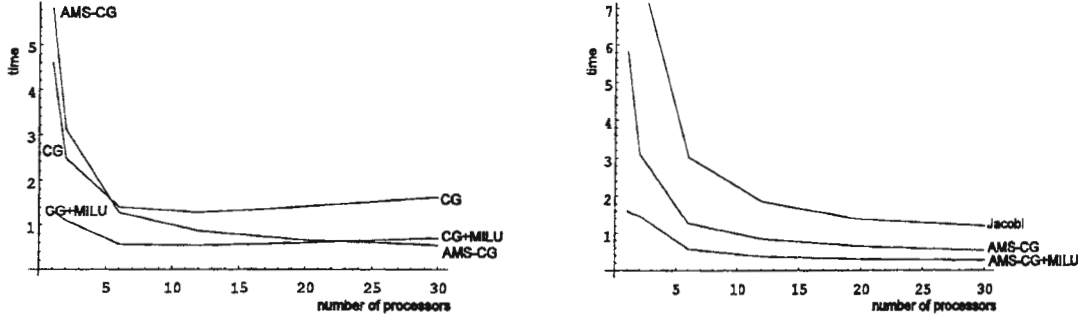
Figure 4. Average execution time per time step

The CG and AMS-CG methods require more computations per iteration, with or without preconditioning. However, the number of iterations is less for CG and AMS-CG. This results in a better overall performance of CG and AMS-CG for a complete time step. Figure 4 show the average execution time per time step against the number of processors. The left graph in Figure 4 shows that AMS-CG is always better than Jacobi but worse than CG and CG+MILU on a small number of processors. If the number of processors increases AMS-CG becomes better than CG and CG+MILU. The right graph in Figure 4 shows that the average execution time of AMS-CG+MILU is better then the execution time of AMS-CG. The execution time of AMS-CG+MILU is also better than the execution time of SOR.

Speedup

The convergence check implies that every iteration a global dot procuct must be computed. Therefore it is advantageous not to check the convergence every iteration. In our experiments we choose to check the convergence every 10 iterat ions. This reduces global communication considerably. It is not relevant to do this for CG, because one of the 2 dot products per iteration in CG is the same as for the convergence check. Figure 5 shows the relative speedup compa red to the execution time on 2 processors with 2×1 subdomains. The figure is clear: The global communication of CG and CG+MILU results in a very bad speedup. AMS-CG and AMS-CG+MILU have roughly the same speedup as Jacobi and SOR, which are known to be well parallelizable.

Scalability

To evaluate the scalability we kept the problem size for each subdomain constant. Each subdomain consisted of 50×50 gridpoints. The execution times are compared with the execution time for 1 global domain. Table 2 provides the execution time for 6×5 subdomains divided by the execution time for 1 global domain. Note that the problem size grows proportionally with the number of subdomains. This results in increasing execution times with increasing number of subdomains. Also the communication overhead results in larger execution times with more subdomains. On the other hand there is a speedup when the number of processors increases.

The execution time of the Jacobi method with increasing number of subdomains grows fast. This is caused by the slow convergence of this method when the problem size increases. The SOR method does quite well in this case but its performance soon worsens as the time step increases
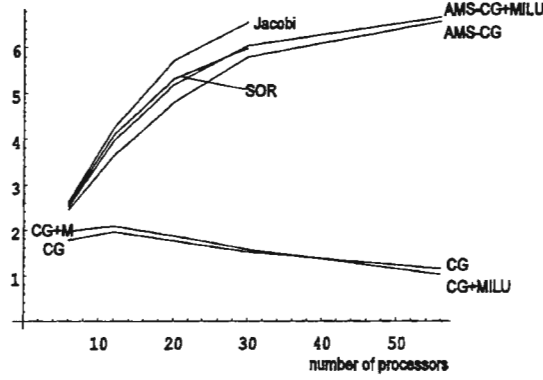
Figure 5. Scaled speedups compared to relative to the execution time of 2x1 domains (on 2 processors).

Table 2
Execution time 6 × 5/execution time 1 × 1

method	time step 600 s	1200 s	2400 s
Jacobi	20.0	-	-
SOR	6.3	11.4	11.3
CG	16.3	16.6	16.9
CG+MILU	8.5	11.0	15.7
AMS-CG	10.0	9.9	8.4
AMS-CG+MILU	4.4	7.3	15.8

(e.g. with 1200 s the execution time ratio becomes 11.4). The CG and CG+MILU methods are not good scalable because of the fast increase of the overhead of global communication. The AMS-CG+MILU has the best scalability for small time steps, but it becomes worse for large time steps. The reason is that the preconditioning is then less effective. The performance of the AMS-CG method is in between and its convergence is insensitive to the time step.

6. Conclusions

We compared the iterative methods Jacobi, SOR, CG and AMS-CG on the solution of a system of equations which describes wave propagation. It clearly shows that there is a trade-off between parallelism and convergence rate in choosing a solution method for parallel computation. The (preconditioned) AMS-CG method is well suited for a parallel solution of this application problem in combination with domain decomposition. In this case the method has convergence properties comparable to (preconditioned) CG. The speedup of AMS-CG is much better than the speedup of CG. This is due to the fact that AMS-CG requires only local communication. MILU preconditioning results in better convergence.

It is remarkable that the number of iterations makes a big jump from 1 subdomain to 2 subdomains, while this is not observed for AMS-CG. The number of iterations increases only slightly from 2 subdomains to 3 subdomains and more. Future research should analyze the

preconditioner and improve it. An interesting topic is to implement a Schur complement preconditioner and compare the convergence and scalability. The Schur complement preconditioner generally has a faster convergence [1]. However, it requires more communication, for instance, we must be careful to avoid global communication in computing a Schur complement in a parallel implementation.

REFERENCES

1. T.F. Chan and D. Goovaerts, *A note on the efficiency of domain decomposed incomplete factorizations*, SIAM J. Sci. Stat. Comput., Vol. 11, No. 4, pp. 794-802, July 1990.
2. E. de Goede, *Numerical methods for the three-dimensional shallow water equations on supercomputers*, Ph.D. Thesis, University of Amsterdam, The Netherlands, 1992.
3. I. Gustafsson, *Modified Incomplete Choleski Methods*, In: *Preconditioning Methods. Theory and Applications*, Edited by David J. Evans, Gordon and Breach, Science Publishers, New York, 1983, pp. 265-293.
4. J.A. Meijerink and H.A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148-162.
5. J.W.M. Lander, P.A. Blokland and J.M. de Kok, *The three-dimensional shallow water model TRIWAQ with a flexible vertical grid definition*, Technical Report RIKZ/94-0..., National Institute for Coastal and Marine Management, The Hague, the Netherlands, 1994.
6. J.J. Leendertse, *A new approach to three-dimensional free-surface flow modelling*, Technical Report R-3712-Neth/RC, Rand Corporation, Santa Monica, Cal., 1989.
7. H.X. Lin, et al., *Parallel Simulation of 3-D flow and transport models within the NOWESP project*, Simulation Practice and Theory, Vol. 3, 1995. pp. 257-271.
8. M.R.T. Roest, *Partitioning for Parallel Finite Difference Computations in Coastal Water Simulation*, Ph.D. Thesis, Delft University of Technology, 1997.
9. E. de Sturler and H.A. van der Vorst, *Reducing the effect of the global communication in GMRES(m) and CG on parallel distributed memory computers*, Technical Report 832, Mathematical Institute, University of Utrecht, The Netherlands, 1994.
10. A.H. Veenstra, H.X. Lin and E.A.H. Vollebregt, *A comparison of scalability of parallel iterative methods for shallow water equations*, in Proceedings of the tenth conference on Domain Decomposition Methods, Contemporary Mathematics, AMS, 1998.
11. E.A.H. Vollebregt, *Multi-Subdomain Conjugate Gradients: A global inner product free CG-type method*, Techn. Report, Department of Applied Math.&Informatics, TU Delft, 1997.
12. E.A.H. Vollebregt, *Parallelisation of algorithms for advection diffusion problems*, Simulation Practice and Theory, Vol. 3, 1995, pp. 273-286.
13. E.A.H. Vollebregt, *Parallel Software Development Techniques for Shallow Water Models*, Ph.D. Thesis, Delft University of Technology, 1997.
14. C.B. Vreugdenhil, *Numerical methods for Shallow-Water Flow*, volume 13 of *Water Science and Technology Library*, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1994.
15. T. Yang and H.X. Lin, *The Improved Quasi-Minimal Residual Method on Massively Parallel Distributed Memory Computers*, IEICE Trans. Information and Systems, Vol. E80-D, No. 9, 1997. pp. 919-924.

EFFICIENT MODELING AND SIMULATION OF A VIRTUALLY SHARED MEMORY ARCHITECTURE

Andreas Pipis[1], Georgios K. Theodoropoulos [2*], Michael Stefanidakis[1], Dimitris Lioupis [1,2]

[1] University of Patras, Dept. of Computer Engineering & Informatics, Patras, Greece

[2] Computer Technology Institute, Papandreou 70, Zografou, 157 73 Athens, Greece

ABSTRACT

Modeling and simulation have been assigned crucial roles in the design, development, analysis and evaluation of computer architectures. The design of parallel architectures in particular is a complex and difficult endeavor that makes modeling and simulation essential tools. In this case, high simulation performance is a prerequisite since, large workloads need to be simulated for a realistic analysis of the parallel system. This paper presents the approach that has been followed for the modeling and simulation of PiSMA, a virtual shared memory parallel architecture, scaleable up to 100s of processors.

1. INTRODUCTION

Technological and architectural advances have dramatically increased the size and complexity of computer system designs. The need to cope with this complexity and the requirement for shorter development times and reduced design costs have assigned key roles to discrete event simulation modeling [15] [19] in computer architecture research. Discrete event simulation modeling lies today at the heart of digital system design and is the approach mainly employed for the design, testing and evaluation of computer architectures; all levels above the silicon and switch levels are typically modeled using this approach [10]. Simulation modeling is an essential tool for experimenting with alternative ways of using the available silicon area, verifying the timing behavior and functional correctness and measuring the performance of alternative architectural designs. This is particularly true for parallel, multiprocessor architectures whose size, complexity and asynchronous, non-deterministic behavior make their design, testing and evaluation a difficult endeavour. The simulation of digital systems in general, and computer architectures in particular has long been categorized among the highly computation intensive applications. In the case of parallel architectures, high simulation performance1 is a crucial issue for, the simulation of the parallel execution of even a small workload can consume large amounts of CPU time, while in order to obtain a reliable evaluation of the architecture, large workloads need to be simulated.

This paper presents the approach that has been followed for the simulation of PiSMA, a virtually shared memory parallel architecture which is under development at the University of Patras, Greece, within the PENED research program funded by the Hellenic Ministry of Development, General Secretariat of Research and Development (PiSMA-1819 project).

---

* Dr. Georgios K. Theodoropoulos is currently with the School of Computer Science, University of Birmingham, U.K.

[1] Various approaches have been followed to speed up the simulation of digital systems. Special purpose hardware accelerators achieve impressive simulation performance, however, they are expensive and have limited flexibility in terms of simulated element types and delay models [1]. Another technique is partial simulation, wherein only parts of the system are tested through simulation; this can accelerate the overall design process but entails the danger of overlooking costly design errors. A third approach, which the availability of parallel computers has made increasingly popular, is the use of distributed simulation techniques [8].

1

2. THE PiSMA ARCHITECTURE
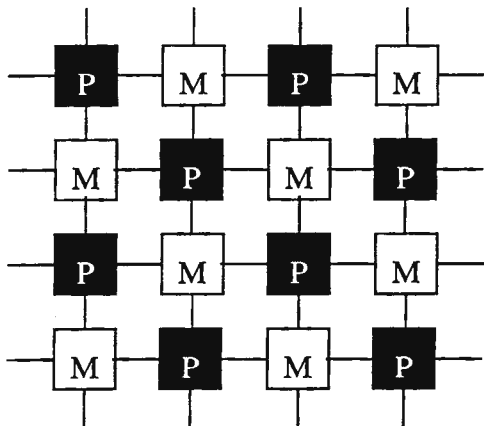
Figure 1: The PiSMA Architecture
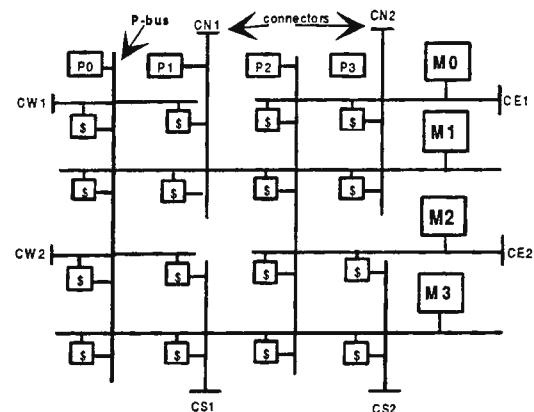
Figure 2: The PiSMA Basic Building Block

The PiSMA virtually shared, distributed memory architecture [11] forms an expandable toroidal grid with alternating processors and memories: each processor is connected to four memories and each memory to four processors (**Figure 1**). This structure enables each processor to communicate directly (through a common memory) with up to eight neighboring processors. Communication to any other processor beyond these eight processors on the grid is performed by message passing which is transparent to the user. Message passing is achieved through memory-to-memory copy. The four ports to memory are implemented as four caches connected on the same bus with the memory module as depicted in **Figure 2**. The processor - memory cluster consists of two basic boards; namely, the *processor board* and the *memory board*. The processor board has two versions, to implement the desired connectivity and allow easy expansion. The memory board contains four memory modules. The lines at the end of the split buses indicate connectors (CN1, CE1, etc.). By connecting together the north connector with the south and the east with the west, the smallest configuration possible which is 4 processors and 4 memories, is formed [12] . The operating System of PiSMA utilizes a diffusion algorithm which ensures that the workload is evenly balanced among the processors [13].

3. MODELLING PiSMA

Several tools have been developed for the modeling and simulation of parallel computer structures; SIMON [7], ADAS [18], PAWS [16], Network II.5 [5], FRIGG [3], Talisman [2] are some examples of such tools. However, although these generic tools exhibit a number of useful features, they do not often provide the functionality required to match the specific needs and objectives of the designer of a particular architecture. Thus, it is often a typical case in computer architecture research, to build custom discrete event simulators. This was the case with PiSMA too; in order to model the unique features of PiSMA and obtain an experimentation platform with the necessary functionality and of high performance, a custom discrete event simulator has been developed. The simulator is written in C and executes on a SUN SPARCstation 10 platform. It is execution driven, executing machine language code produced by a standard C compiler.

A distinct feature of the simulator of PiSMA is that it allows part of the simulated code to execute directly on the host platform. Typically, in computer architecture simulations, the simulated code (e.g. benchmarks) is meta-executed by the simulator which acts as an interpreter and which is in turn executed on the host computer. In the PiSMA simulator however, part of the simulated code may be executed directly on the host computer. The simulator itself does not perform the execution of the simulated code; it just encapsulates and describes the structural characteristics and operational rules of PiSMA. An efficient approach has been followed for the realization of this idea, whereby the simulated assembly code is preprocessed and special *swap commands* are inserted after certain instructions. Swap commands transfer the execution control between the Operating System of the simulated architecture PiSMA and the host

2

platform respectively, thus enabling both architectures to work on the same simulated code. The simulated code is compiled and linked with the simulator code to form a single piece of object code which is then executed on the host computer. This technique is rendered feasible by using the same processor (i.e. the SUN SPARC 10) as both the host and the target processor of the simulation, i.e. it is assumed that the processors of the simulated PiSMA are SPARC10s. By executing the simulated code directly on the host computer instead of using the simulator as an intermediate interpreter, high simulation performance is achieved, an issue which is considered crucial and of great importance.
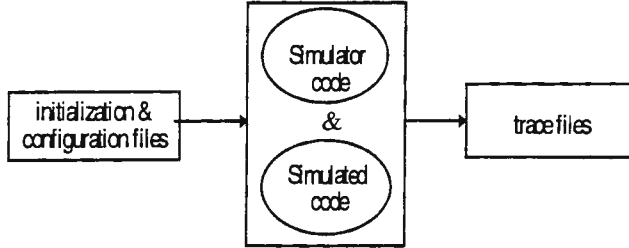
Figure 3: The Simulator Interface

A high level view of the simulator is depicted in Figure 3. Since, after compilation, the simulated code becomes part of the simulator, the input provided to the simulator includes only:

• configuration files with information regarding the number of processors a nd memories of the PiSMA grid, memory size, cache size etc.

• initialization files with data used as input to the simulated code.

The output generated by the processor is directed to a number of trace files and provides information regarding values such as the total execution time in cycles, the number of OS-application instructions executed, the time spend in specific parts of code, information regarding the operation of memories and caches etc. The following sections provide a short presentation of the key elements of the simulator; for a detailed description of the structure and operation of the PiSMA simulator the reader is referred to [14].

4. THE STRUCTURE OF THE SIMULATOR

```
bootstrap()        PiSMA grid initialization
init()             System variable initialization
repeat
  for id=0 to DIM  Single step all
    pe(id)         processors
    me(id)         memory modules.
  update()         Update module
  clock++          Advance System clock
until shutdown
```

where **DIM** is the number of processors and memories.
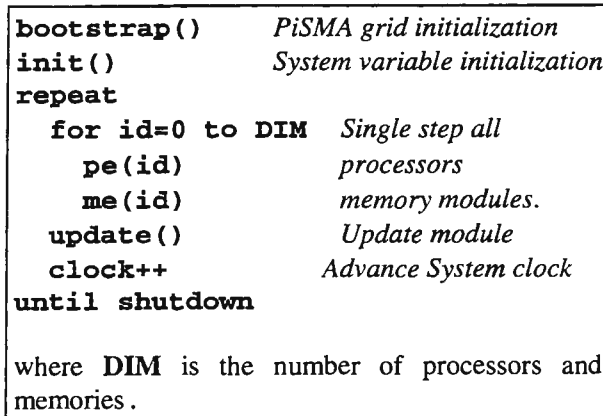
Figure 4: The Simulator Main Loop.
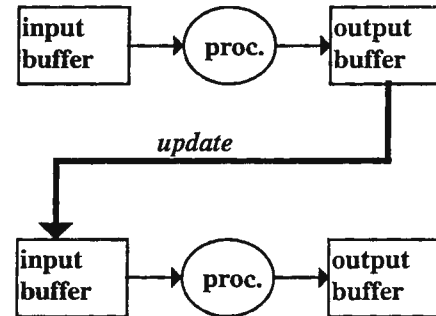
Figure 5: Inter-process Communication in the Simulator

The simulator is organized as a set of logical processes, with each process modeling a functional module of PiSMA. Two types of logical processes are used, namely *pe()* and *me()*, for the modeling of processors and memories respectively. The memory process also incorporates the functionality of the four caches. Instances of these processes are connected together to form the PiSMA toroidal grid.

Figure 4 illustrates the main loop of the simulator. The simulator is time-driven. It utilizes a global clock variable which is the heart of the simulation operating as an autonomous unit. The simulated time advances every instruction cycle. The time-driven approach was adopted for the PiSMA simulator because it provides for the precise and detailed control of the simulator while the space required for storing run-time monitoring information is small. The simulation of parallelism is achieved by single stepping each processor and memory process for one cycle in a round robin fashion. The processes of the simulator exchange values via buffers, as depicted in Figure 5. A process executing at time step t_n consumes data that were produced at time step t_{n-1} and produces data that will be consumed by its peer processes at time step t_{n+1}. When a single-stepping cycle for all processes is completed, an *"update"* routine is called which

3

moves the produced output values to the input buffers, to provide the input data for the next cycle. The termination of the simulator takes place upon the encounter of the special *"shutdown"* instruction in the simulated code.

4.1 THE PROCESSOR LOGICAL PROCESS

A high level view of the logical process pe() is depicted in Figure 6. This process consists of two interacting levels, namely the control loop and the simulated code.
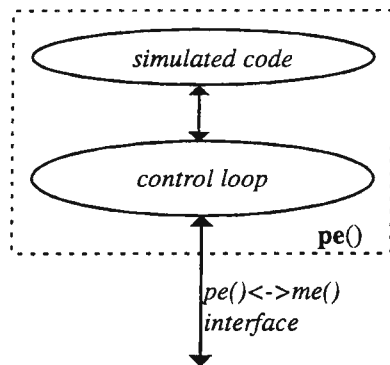
Figure 6: The Simulator Logical Processor

4.1.1 THE CONTROL LOOP

Essentially, this level corresponds to the simulated processor hardware. The control loop of the simulator is implemented as a finite state machine which simulates the functions of a typical RISC processor. These include the communication with memory and the execution of the simulated code. Furthermore, the control loop controls the flow of execution between the simulator and the simulated code (Operating system or application). The operation of the control loop is illustrated in Figure 7.

4.1.2 THE SIMULATED CODE

The simulated programs, including both user applications and Operating System routines, are written in an extended version of C. This supports parallel programming structures, such as fork/join and synchronization control of multiple processes (barriers). A preprocessor is used to convert this code into conventional C, which is then compiled and linked together with the simulator.

In order to achieve the required functionality, the assembly code generated by the C compiler from the simulated source code, is fed into special filters which insert extra instructions of special type between the normal, assembly language instructions as depicted in **Figure 8**. These instructions are referred to as *"swap instructions"* and act as traps that transfer the control of program execution to the control loop of the corresponding processor logical process.

The execution of simulated code assembly instructions and swap instructions alternates; after each individual assembly instruction of the simulated code, the appropriate swap instruction is executed. The execution of a swap instruction has the following effects:

1. The necessary host machine registers (processor state) are saved.
2. A parameter table with information based on the type of the swap instruction is updated.

Figure 7: The Control Loop

3. The simulator control is transferred to the corresponding control loop.

4

The control loop uses the parameter table to execute the corresponding assembly instruction. Instructions that do not access the memory are directly executed while instructions that access the memory are simulated by the control loop. This is because, although the processors of PiSMA are assumed to be of type similar to the processor that hosts the simulation, the references to the host memory must be mapped to the multiple memory modules and caches of PiSMA.
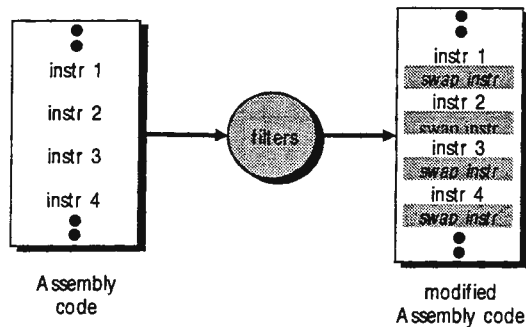
Figure 8: The Insertion of Swap Instructions

Thus, within the context of the simulator, memory modules and caches are simulated structures. One of the main objectives of the simulation is to produce realistic memory address traces of the PiSMA architecture under consideration. These traces drive the cache and memory simulated modules to produce realistic measurements. When the execution of the swap instruction is completed, the control returns to the simulated code for the execution of the next instruction.

4.1.2.1 SWAP INSTRUCTIONS

The processor may operate in one of two modes, namely, OS_MODE, executing OS code and USER_MODE, executing user applications. In OS_MODE absolute addressing is permitted while in USER_MODE access to memory is performed only relatively to each application's data segment. Depending on the mode of operation, different swap instructions may be executed which are handled in a slightly different manner. There are also swap instructions that are common to both modes of operation.

1. **Common Swap Instructions.** These are common to both OS_MODE and USER_MODE.

   a) *SwapNormal.* Simulated instructions that do not access the memory are followed by this swap instruction.

   b) *SwapNewPC.* This swap instruction is inserted after each label in the simulated assembly code to save the simulated Program Counter (e.g. in the case of branch instructions). The new PC value for the instruction is produced by a special filter which keeps track of the relative positions of the simulated instructions before the insertion of the swap instructions.

   c) *SwapTerminate.* This instruction declares the termination of the execution of a simulated function. Upon the encounter with this instruction, the control loop updates the Program Counter to point to the return address in the calling function.

2. **OS_MODE swap instructions.** These are executed only in OS_MODE of operation.

   a) *SwapWriteAbs.* This instruction is used by the operating system to perform a memory write using absolute addressing. The operating system has access to the whole address space, including the applications' data segments.

   b) *SwapReadAbs.* This instruction is used by the operating system to perform a memory read using absolute addressing.

   c) *SwapOsCall.* This instruction enables an operating system routine to call another OS routine. When executing this instruction, the operating system: a) stores the PC into the stack b) creates a PCB for the new routine c) loads the PC with the starting address of the new routine and d) makes all the necessary arrangements so that upon the commencement of the new clock cycle the new routine is executed.

   d) *SwapLDST.* These instructions are used to implement simulated semaphores. Since PiSMA is a shared memory architecture, semaphores are required to achieve mutual exclusion when accessing common memory. *SwapLDST* implements the atomic instruction 'Load from Memory and Store 1'. The release of the semaphore is achieved in the application source code by calling the special function unlock(); this is actually converted to a *SwapApWrite* instruction with data=0.

5

e) *SwapDemand.* After a demand request has been served by the operating system, this instruction is used to transfer control to the corresponding application. The operating system of PiSMA is preemptive and multitasking. Thus, it utilizes structures such as a) a ready queue, with the PCBs of the ready simulated processes, and b) a blocked queue, with the PCBs of the blocked and waiting processes. Processes are executed in a round-robin fashion. A time slice is assigned to each user process (OS processes are non-preemptable), upon the completion of which the process is preempted, its PCB is placed at the tail of the ready queue and the process at the head of the ready queue is assigned the CPU. *SwapDemand* is also executed when a PiSMA processor receives the messages a) FORK, for the creation of a new process, and b) REMOTE READ/WRITE from/to local memory request from a remote processor.

3. **USER_MODE swap instructions**. These are similar to the swap instructions of the OS_MODE, but memory is accessed via a segmentation scheme instead of using absolute addressing. The actual memory address is produced via a combination of a relative offset and the corresponding code or data segment. USER_MODE swap instructions include also operating system traps which activate the corresponding service routines (e.g. P4 [4] and PVM [9] primitives).

   a) *SwapApRead & SwapApWrite.* These are similar to the corresponding OS_MODE instructions but absolute addressing is not supported.

   b) *SwapApPRead & SwapApPWrite.* These are memory Read and Write instructions respectively which support dynamic pointer usage. These instructions refer to a memory location where the descriptor of the pointer is stored. The descriptor is initialized during the execution of the program and provides the information required to access the data referred to by the pointer.

   c) *SwapOSTrap.* These instruction activate an OS trap forcing the control of the execution to be transferred temporarily to the operating system. The traps include:
      - *SwapApFork.* Creates a new parallel process.
      - *SwapApCall.* Calls a function from the application program.
      - *SwapApMalloc.* Dynamic memory allocation
      - *SwapApFreemem.* Dynamic memory release.

   d) *SwapApShutdown.* This instruction declares the end of program execution and terminated the simulator.

4.2 THE MEMORY LOGICAL PROCESS
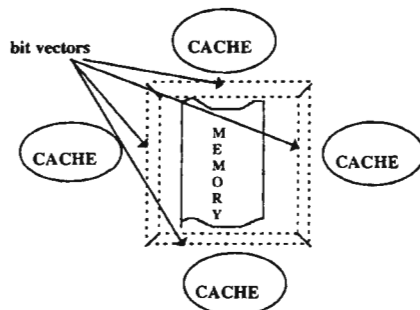
Figure 9: The memory Logical Process

The memory logical process models one main memory module and the four cache memories that are physically connected to it. Memory is modeled as a word table. Cache memories access this table using bit vectors as depicted in Figure 9. A bit of value 0 corresponds to a cache miss, in which case the processor is forced to wait until the data block is fetched from memory, while a bit of value 1 corresponds to a cache hit, in which case the processor can access the requested data. The block replacement algorithm utilizes the LRU policy.

6

5. PERFORMANCE RESULTS

Benchmark	Number of Processors				
	4	**16**	**36**	**64**	**100**
stereo	493	912	1603	2034	3276
water	442	418	634	1025	1373
fft	153	238	452	652	1162
lu	47	144	397	640	1044
qsort	103	126	180	293	414
dmult	10	12	15	20	32

Table 1: Elapsed Execution Time (in minutes)

above benchmarks.

Benchmark	Number of Processors				
	4	**16**	**36**	**64**	**100**
stereo	10096	10223	10192	10263	10127
water	12324	14672	15603	15537	15564
fft	12631	15738	16392	17608	16547
lu	11496	14604	12939	12445	11891
qsort	10854	10539	9336	9454	9110
dmult	13668	10408	10688	10714	10553

Table 2: Simulated Clock cycles/sec

Several classes of benchmarks have been used to study the behavior of the simulator and verify its performance including the Toy benchmarks **Quicksort** (N=10000) and **Dmult** (N=64), the **LU** (n=4 and B=12 and N=48), **FFT** (N=128) and **WATER-Nsquared** (N=512) benchmarks from the SPLASH-2 suite [17], and the **Stereo** benchmark from the CMU suite [6]. Tables 1 and 2 present some indicative results regarding the performance of the simulator when executing the

Table 1 shows the real time (in minutes) taken by the simulator to execute each benchmark for several PiSMA grid sizes, while Table 2 shows the number of simulated clock cycles executed by the simulator per second.

It can be seen from Table 1 that the simulation time required for small grid sizes is in the order of few hours for small and large workloads alike. When large grid sizes and large workloads are simulated the simulation time is increased by an order of magnitude. These values are substantially smaller than those typically encountered in the simulation of parallel architectures; such heavy simulations would typically require several weeks.

Table 2 shows that the PiSMA simulator maintains a constant execution rate of about 10.000 instructions per second. This performance is also considered particularly high for simulators. The slight variations observed in the execution rates of the simulator between benchmarks are attributed to the different usage of the OS and memory management primitives by each benchmark.

6. CONCLUSIONS

This paper has described the methodology followed for the construction of a discrete event simulation model for the PiSMA virtually shared memory architecture. High performance was one of the main objectives of this methodology. To meet this objective, it was decided to adopt a hybrid technique, whereby a significant part of the simulated code is executed directly on the host computer while the rest is interpreted by the simulator. The achieved performance results have justified this decision proving the efficiency of this technique. The high performance of the simulator has enabled the PiSMA design group to conduct experiments using large workloads on PiSMA configurations consisting of hundreds of processors, thus achieving a realistic analysis of the architecture within reasonable time boundaries.

REFERENCES

[1]   Agrawal, P., et al., "MARS: A Multiprocessor Based Multiprogrammable Accelerator", IEEE Design and Test of Computers, 4, 10, October 1987, pp. 28-36.

[2]   Bedichek, R., "Talisman: Fast and Accurate Multicomputer Simulation", SIGMETRICS 1995.

[3]   Bier, J. C., Lee, E. A., "FRIGG: A Simulation Environonment for Multiple Processor DSP System Development", Proceedings of 1989 IEEE International Conference on Computer Design, October 1989, pp. 280-283.

[4]   Butler R. Lusk E. "Users Guide to the P4 Parallel Programming System" Tech. Report ANL-92/17, Argonne National Lab. Oct. 1992.

[5]   Network II5, CACI Products Inc., La Jolla, California.

[6]   Dinda, P., et. al. "The CMU Task Parallel Program Suite", Tech. Report CMU-CS-94-131, Dept. of Computer Science, Carnegie-Mellon University, March 1994.

7

[7]   Fujimoto R., "SIMON: A Simulator of Multicomputer Networks" Tech. Report UCB/CSD83/140, Computer Science Division, U.C. Berkeley, Sept. 1983.

[8]   Fujimoto R., "Parallel Discrete Event Simulation", Communications of the ACM, 33, 10, October 1990, pp. 31-53.

[9]   Geist A. et al "PVM3 Users Guide and Reference Manual" ORNL/TM-12187 Sept. 1984.

[10]   "Hardware Description Languages", Advances in CAD for VLSI, Volume 7, Hartenstein, R. W., (Editor), North Holland, 1987.

[11]   Lioupis, D., Kanellopoulos, N., "CHESS Multiprocessor: A Processor-Memory Grid for Parallel Programming", Cache and Interconnect Architectures in Multiprocessors, edited by Michel Dubois and Shreekant Thakkar, Kluwer Academic Publishers, 1990

[12]   Lioupis, D., Kanelopoulos, N., Stefanidakis, M., "The Memory Hierarchy of the CHESS Computer", Microprocessing and Microprogramming 38, pp 99-107, 1993

[13]   Lioupis, D., Stefanidakis, M., "Dynamic Load Balancing on a Virtually-Shared Memory Parallel Computer System", PARLE 94, Athens, 1994

[14]   Lioupis, D., Theodoropoulos, G., Pipis, A., Smirli, M., "Design of the Parallel Architecture PiSMA: The PiSMA Simulator", PENED/PiSMA/2/96 Internal Progress Report, Computer Technology Institute, Athens,Greece, December 1996.

[15]   "Discrete Event Simulation Modelling: Directions for the 90s", Panel Session, Chair Radiya, A., Proceedings of the 1992 Winter Simulation Conference, December 1992, pp. 773-782.

[16]   Performance Analysts Workbench System, Scientific and Engineering Software, Austin, Texas

[17]   Woo S.C., Ohara M., Torrie E., Singh J.P. & Gupta A., "The SPLASH-2 Programs: Characterization and Methodological Considerations", Proc. of ISCA'95, June, 1995.

[18]   Yalamanchili, S., Carpenter, T., "Modelling and Analysis of Multiprocessor Architectures", Proceedings of the 7th Computers in Aerospace Conference, October 1989, pp. 344-350.

[19]   Zeigler, B. P., "Theory of Modelling and Simulation", John Wiley and Sons, 1976.

8

A heuristic for the analysis of particle simulation problems

Kees Lemmens (C.W.J.Lemmens@Twi.TUDelft.nl),
Fac. of Information Technology and Systems,
Delft University of Technology (The Netherlands).

Version 1.0: 14th February 1998

Abstract

In the case of large scale particle physics simulation the world currently seems to follow two distinctly different tracks: Tree-code approaches like Barnes-Hut or Fast-Multipole and on the other side Particle-Mesh methods and derivatives. Recently I have compared both methods with software that was able to use both methods on the same problem. I then decided to propose a heuristic that could make it possible to make the choice between the two different families on something else than only intuition.

The most important features of a dataset to be examined are the following :

1. The number of particles (N)

2. Total intended simulation time (T)

3. The variation in the distance between each particle and it's *closest* neighbor

4. The individual and total potential and kinetic energies and their ratio

5. The mass (density) distribution and it's autocorrelation function

6. The average displacement and its variation during the simulation time T

7. Ratio between average shortest interparticle distance and average displacement (indication for the number of collisions)

Several of the features mentioned above require $O(N^2)$ operations, which makes them difficult to compute, unless an approximation technique is used. Examples are the potential energy computation, the minimum distances and of course the density autocorrelation.

I used a combination of approximation techniques, such as a statistical method (randomizer) for the average of the minimum distances, a Tree-Code like algorithm to compute the potential energy for each particle and a Particle-Mesh like algorithm using FFT's to compute the density auto-correlation.

1 Introduction

Many different physical phenomena can be described as interactions between pairs of particles. Examples are the motion of planets and satellites, colliding star clusters, but also molecular dynamics and electron movement in conductors and semi-conductors.

In all these cases there exists a *pairwise* interaction (force) between the particles in these systems. These forces can be either a gravitational, electrical or another fundamental force or even combinations with both attractive and repulsive components.

If we want to simulate such a system in time we should calculate for every timestep the *total* force acting on each particle (which is composed of all pairwise interactions with the other particles and thus requires $O(N^2)$operations) and then calculate for each particle the change of speed and the displacement within this interval. This is basically what we do in almost all types of particle simulations.

Virtually all used algorithms are refinements based on 2 fundamentally different methods :

- Hierarchical Tree-Code Algorithms like Barnes -Hut and Fast-Multipole (TC).

- Particle-Mesh algorithms like PM and P^3M, based on solving Poisson equations on a mesh (PM).

How is the choice for either TC or PM made ? Both have their advantages and disadvantages, but in the literature about particle simulations there was surprisingly little I could find about this topic. There is a lot being said about collisionless and collisional systems and uniform and non-uniform distributions, but how to choose for one of these two types given a certain problem is a question that is left largely unanswered. I became interested in this subject myself and tried to do the following :

- Determine which parameters/properties of a given dataset could be used to make a reasonable decision, not mainly based upon intuition.

- Design a simple heuristic that could be used to make the choice without the need of many complicated computations. Using all basic methods and see which one gives the best results is not the best option in my opinion ...

- Implement the code for the heuristic, which should be fast enough to be able to give a clear indication for even very large datasets within a reasonable time.

2   Parameters and properties

If we need to distinguish between parametersets to be able to make the right choice, what parameters could then be used to make the best decision ? I'll discuss the ones I considered to be the most interesting briefly hereafter :

1. The **number of particles** (N). It may sound a trivial, but of course if the number of particles is lower than a certain threshold it is useless to use either a Tree-code or a Particle-Mesh method, as Particle-Particle is definitely the most accurate and in that case undoubtedly also the fastest choice.

2. The **distances between each particle and it's closest neighbor** and the variation in these distances. For each particle we can compute the distance to the closest neighbor: this is the minimum interparticle distance (MID) for *that single particle*. The shortest value for all these MID values is the *shortest* minimum interparticle distance ($SMID$) and the average of all these values is the *average* minimum interparticle distance ($AMID$).

   Computing the **standard deviation in the minimum distances** (MID) can supply even more information. A large standard deviation implies a large variation in

67

these minimum interparticle distances, while a small one implies a more or less uniform distribution.[1]

3. **The potential and kinetic energy and the ratio between them.** The potential energy is a measure for the amount of acceleration that would occur during a simulation. The kinetic energy on the other hand says something about the initial velocity of the particles. If the kinetic energy dominates, the approximated displacement during the whole simulation can be computed using *only* the kinetic energy and the intended duration of the simulation.

   If on the other hand the potential energy dominates we could give an indication about the displacement by *only* using the initial acceleration and assuming it would be constant during the whole simulation.

4. **The mass (density) distribution and its autocorrelation** (DAC). The density distribution can be obtained in a similar way as with the Particle-Mesh method. If we use this result to compute the *autocorrelation* function of the density distribution (see section 3) we can obtain information about the density uniformity.

   If the system appears to be more or less uniform according to this observation, the Particle-Mesh method could be used, provided the ratio between the average minimum interparticle distance and the average displacement $(AMID/AD)$ is large enough to guarantee that many collisions are unlikely (see next point).

5. **The ratio between the average minimum interparticle distance** $(AMID)$ **and the average displacement** (AD). The combination between the average minimum distance and the average displacement can give useful information about the number of collisions that may occur during the simulation.

   This can make the distinction between collisional and collisionless systems and thus between the choice for either Tree-Code or Particle-Mesh, but of course this only makes sense if the variation in either the average minimum distance and the average displacement is not too large.

   In a distribution that was already found to be reasonably uniform, this could be useful extra information to improve the quality of the decision.

6. **Number of detected collisions** during one simulation step. What we actually do is compute one timestep using a Tree-code like algorithm. In this simulation collisions can occur. If we find a significant number of collisions within this single timestep Particle-Mesh is certainly not the best option. Note that it may also be wise to reconsider the timestep in this case ....

Many of the observations mentioned above were combined into a piece of software that is able to give an indication about the properties of a certain dataset and at the same time give a hint what method would be best suited according to these observations. The complete heuristic as was implemented in my analyzer program is shown in figure 1.

---

[1]Note that for a lattice of particles - which could be considered as very uniform - the standard deviation should be always 0 ! Also note that this value doesn't say anything about the *density* distribution, unless all particles have the same weight !

3 Autocorrelation

An important part of the program is the computation of the autocorrelation function. This can give us valuable information about the uniformity of the interparticle distances or about the density distribution and that is why I want to give it a little more attention here.
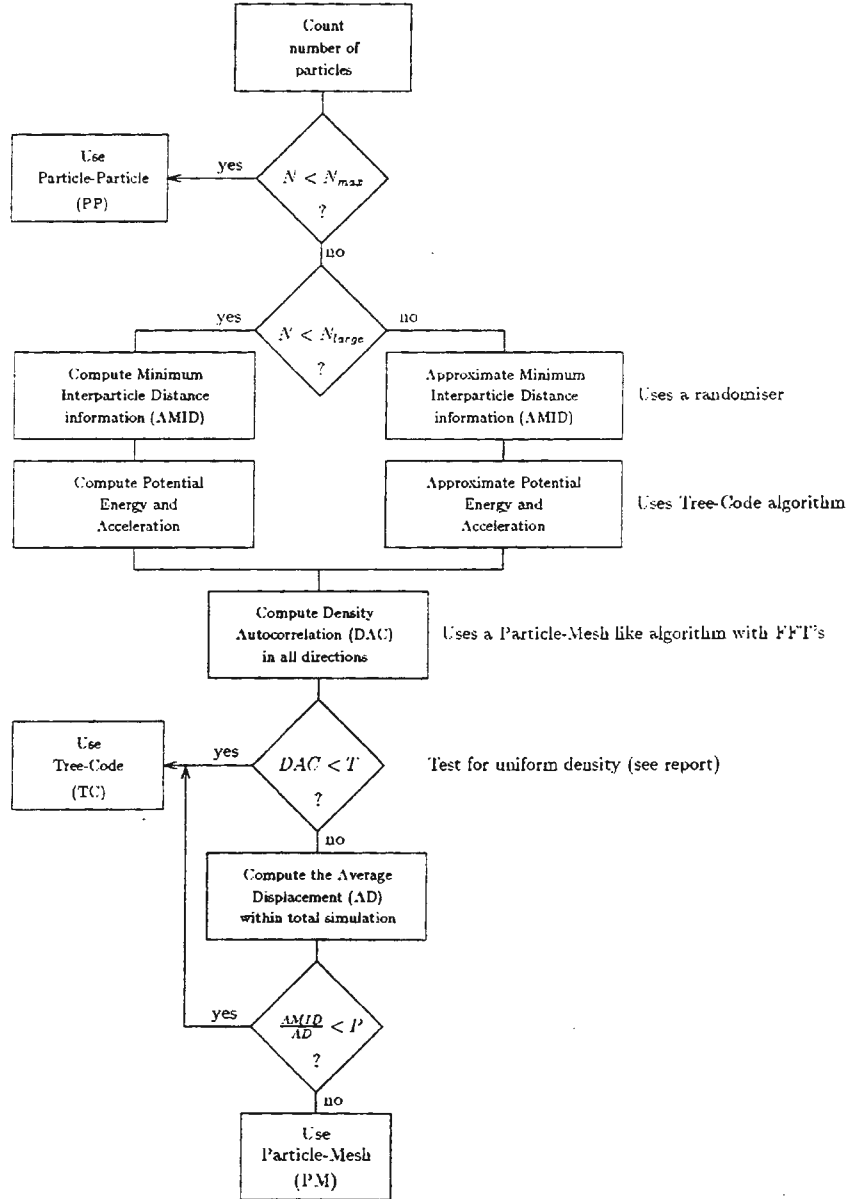
Figure 1: Flowchart for analyzer program.

The autocorrelation of a function $f(x)$ can say something about how regular this function $f(x)$ is over a certain interval Δx : it is the amount of "overlap" we find if we shift the function over a distance Δx to the left or the right. We just measure how large the overlapping area is for a certain value to find the autocorrelation for that distance.

A large overlap indicates a large auto-correlation for this value of Δx and a small overlap on

the other hand means a small auto-correlation for Δx.

Of course there exists for every value of Δx a corresponding value for the "overlap" and all these values together gives us a new function : "the autocorrelation function" $A(f(x))$ for the initial function $f(x)$.

An autocorrelation function can be computed in the following way :[2]

$$f(x,y) \Rightarrow A(f(x,y)) = \int_l \int_k f(x,y) \cdot f(x+k, y+l) \, dk \, dl \tag{1}$$

If we define a helper function $g(x,y) = f(-x,-y)$ we can rewrite this equation into a (2D) convolution of the functions $f(x,y)$ and $g(x,y)$. This yields us for equation 1:

$$A(f(x,y)) = \int_l \int_k f(x,y) \cdot g(k-x, l-y) \, dk \, dl \tag{2}$$

Agreed, this is still not really pleasant, but this will change if we transform the equation above into the (2D) Fourier domain, just as is done with the Poisson equation in Particle-Mesh simulations :

$$\widetilde{A}(f(x,y)) = \widetilde{A}(k,l) = \widetilde{f}(k,l) \cdot \widetilde{g}(k,l) \tag{3}$$

Equation 2 then changes into a straightforward multiplication of the two (transformed) functions $\widetilde{f}$ and $\widetilde{g}$.

As with the Poisson equations before, the functions $f(x,y)$ and $g(x,y)$ could be replaced with a set of numbers : the values of $f[x,y]$ and $g[x,y]$ at the gridpoints that specify the density distribution. That gives us the possibility to use FFT's to obtain the autocorrelation distribution: transform the distributions $f[x,y]$ and $g[x,y]$ into the distributions $\widetilde{f}[k,l]$ and $\widetilde{g}[k,l]$, then multiply all corresponding values of $\widetilde{f}$ and $\widetilde{g}$ with each other and finally go back to the spatial domain again using an inverse FFT.

Special attention needs to be paid to the **spacing** and the **dimension** of the grid. :

The **gridspacing** should be of approximately the same order as the mean interparticle distance or else we either wash out information or introduce high fluctuations in the autocorrelation function. I choose to make it about twice the size of the average minimum interparticle distance to smoothen the function somewhat and also to reduce the computation time. The gridspacing is also limited by the number of particles: it is useless to have a number of gridcells that is several magnitudes higher than the total number of particles.

As for the **dimension** of the grid : to avoid aliasing it is important that the grid is more than large enough to contain all particles. I calculate the maximum interparticle distance in both directions and set the grid size at about 4 times the size of the particle space.

4   Analysis of some practical problems

To test the software, I used a number of predefined datasets. For the larger problems ($> 10^3$) I used 2 small helper programs to generate the set at runtime: A dataset generator for monotonous grid patterns and another one for pseudo-random patterns.

Let's take a closer look now at the differences and similarities for some generated patterns.

---

[2] I write functions of 2 variables now as I studied only 2D because they were much easier to visualize
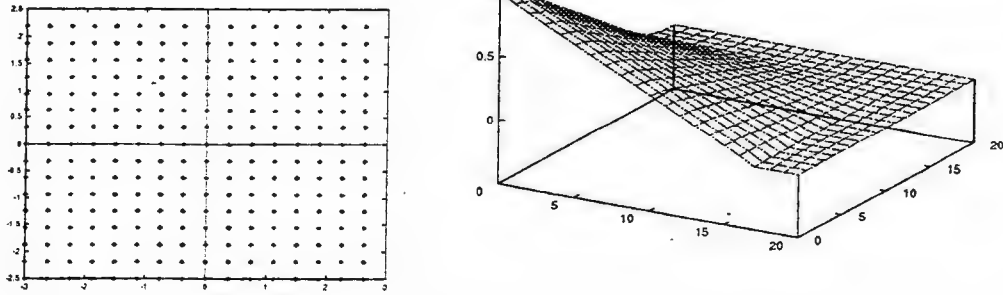
Figure 2: A small gridpattern and it's autocorrelation function (256 particles)

4.1 Uniform problems
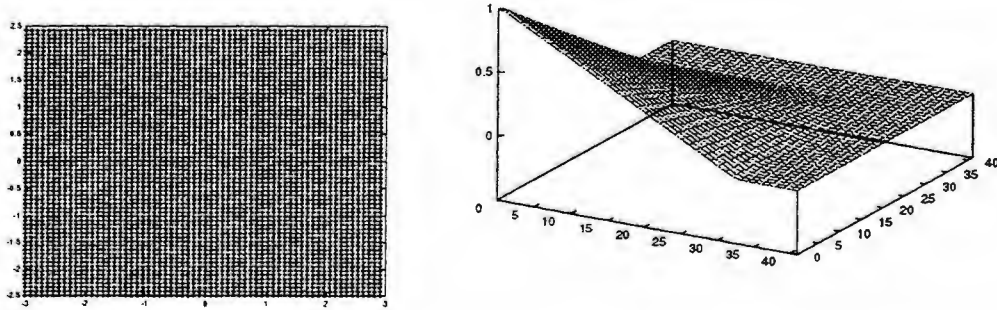
Figure 3: A larger gridpattern and it's autocorrelation function (4096 particles)

For the gridpatterns we can say that they are always uniform, no matter how large the problem is, provided of course that all masses are identical. To see what this means for the autocorrelation function of these patterns, I have plotted a relatively small one and a larger one, together with their autocorrelation functions (see figures 2 and 3). Apart from a few scale factors one can see that they are virtually the same.

For the random patterns (datasets where the positions are generated at random), we'll see that the results are a little different. Examples can be found in figure 4 and 5. Now the smoothness of the autocorrelation function also depends on the gridsize that is used to determine the density distribution !

This is caused by the following effect: as soon as the size of a gridcell is large compared with the average interparticle distance, we'll find approximately the same amount of particles in each cell, no matter at which cell we are looking. But if the size of a gridcell is too small, we'll see large variations as the number of particles per cell is going to vary much more in

that case than with a monotone grid.
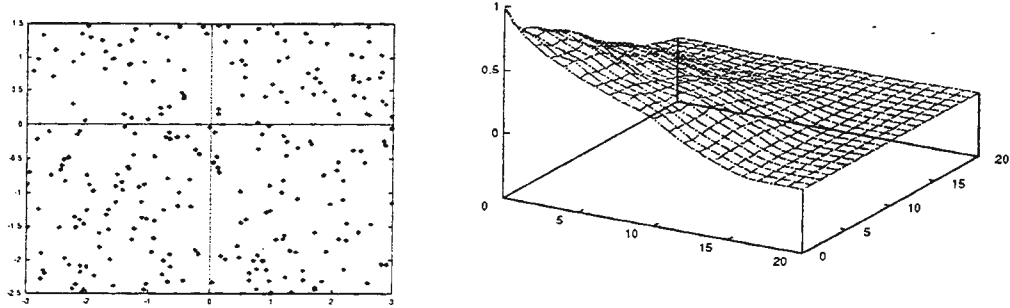
Figure 4: A small random generated pattern and it's autocorrelation function (256 particles)

Nevertheless we can see that the random pictures also have something in common with the monotonous grid pictures : especially for large randomly generated sets, the autocorrelation function is almost identical with the sets that consist of monotonous grids, as the variation in the density is only noticeable on a very small scale. For random sets larger than approximately 10^5 the difference with monotonous grids of the same size is negligible, as long as we are not interested in events that take place on a very small scale and of course if the average particle displacement during the simulation is not too large.
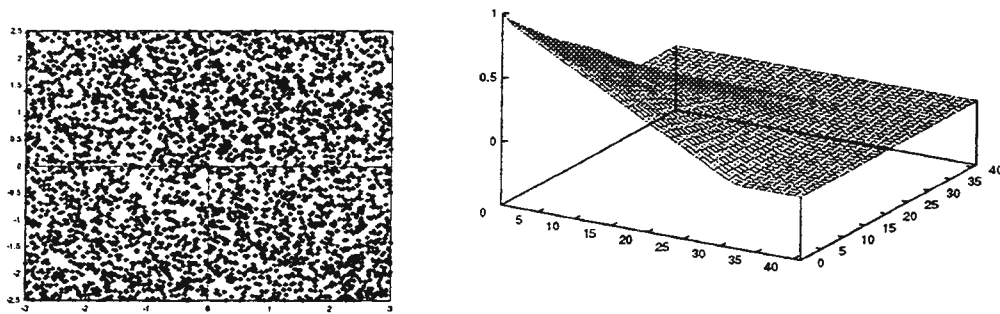
Figure 5: A larger random generated pattern and it's autocorrelation function (4096 particles)

A special situation occurs for the random set when the mean particle displacement lays *between* the shortest minimum interparticle distance $(SMID)$ and the average minimum interparticle $(AMID)$ distance. In that case there will be a noticeable difference in the number of collisions between random sets and monotonous sets : for the monotonous sets we may say $SMID = AMID$ so no collisions would be found, while for the random sets $SMID < AMID$, possibly resulting in a certain number of collisions.

Conclusion : On a large scale both monotonous grids and random sets can be considered to be **uniform** as long as the displacement is small enough.

4.2 Non-uniform distributions
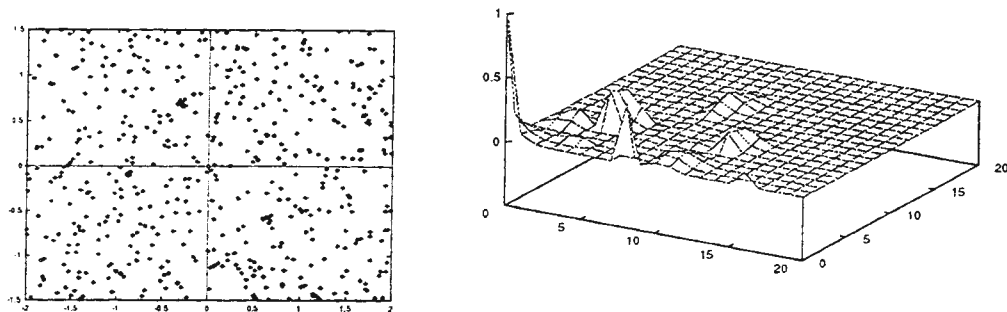
Figure 6: 500 particle random set with a few extra heavy masses causing non-uniformity.

Now let's see what happens with the autocorrelation if we have clearly non-uniform distributions. These distributions can be created by adding some heavy masses on a random set or a grid, by generating a few random sets at different locations in space and merge them into one new set (clusters) or simply by using a set with only a few particles.
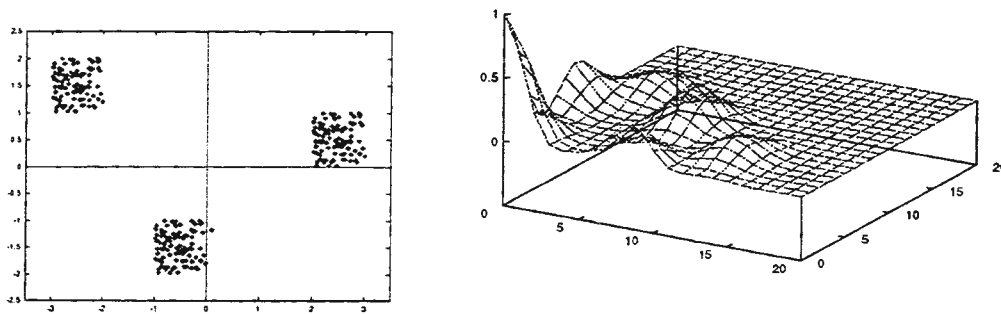
Figure 7: 300 particle set built by clustering 3 smaller random sets of 100 particles each.

Some examples are given in figures 6 and 7. Note that in all cases the autocorrelation function drops very fast as soon we drift away from the center. This leads to the conclusion that the average value of the autocorrelation function, especially around the center, can provide us with the information we are searching for.

I decided to use the average value of the autocorrelation function in both directions but only from 0 to $\frac{1}{4}$ of the grid spacing, to make the distinction clear enough. Far away from the

center all pictures show that the autocorrelation is small, so using these values only makes the distinction less clear. This property was given the symbol T in the flowchart in figure 1. In formula my rather simple definition for the discriminator T looks like :

$$T = \frac{2}{N} \sum_{x=1}^{N/4} A(x,0) + \frac{2}{N} \sum_{y=1}^{N/4} A(0,y) \tag{4}$$

Where the autocorrelation function $A(x,y)$ has to be rescaled to be always between 0 and 1. Let's now summarize some values for T for true random and grid patterns and compare them with the values found for the non-uniform examples. First we'll look at some generated gridpatterns (where $SMID$ and $AMID$ should be essentially the same) :

Particles	T	SMID	AMID
1024	0.512	0.1563	0.1563
4096	0.506	0.0781	0.0781
16384	0.500	0.0391	0.0391
65536	0.490	0.0195	0.0195
131072	0.497	0.0138	0.0138

And for some generated random sets (where $SMID$ should be of course less than $AMID$) :

Particles	T	SMID	AMID
1024	0.2623	0.0226	0.0860
4096	0.4027	0.0041	0.0355
16384	0.4691	0.0037	0.0207
65536	0.4894	0.0004	0.0111
131072	0.4926	0.0013	0.0079

It is clear from the table that the larger the number of particles, the closer the value of T comes to the value found in the gridpattern examples (about $T \sim 0.5$).

Finally we also need the T values for the non-uniform examples, to see if the discrimination made by using the value of T would be sufficient :

Particles	T	SMID	AMID
300	0.0437	0.0025	0.0562
500	0.0032	0.0111	0.0761

As shown in the table, the values of T are much smaller now. I decided to call every set with $T > 0.4$ a *uniform* distribution, every set with $T < 0.1$ a *non-uniform* distribution and all sets with $0.4 > T > 0.1$ sets that need closer inspection. Of course these values are just more or less arbitrary, but nevertheless they are good enough to serve the purpose: a *rough* discrimination between uniform and non-uniform sets.

The sets belonging to the "borderland" could be done using Particle-Mesh as long as the required accuracy is not too high, in the other case it would be better to use Tree-Code, although this may take significantly longer as shown in [Lemmens].

5 Implementation aspects

As usual there were several practical problems to be solved while implementing the heuristic. I will summon the most interesting ones hereafter :

- It should be possible to compare the factor T between all different types of sets that should be analyzed, no matter if they are large or small, uniform or non-uniform. For that reason it was necessary to scale the autocorrelation function back to alllow only values between 0 and 1 (vertical axis). The same applies for the horizontal axis : we need the average value of the summation, so it has to be scaled back with the number of particles N. (see equation 4)

- The interparticle distance information ($SMID$ and $AMID$) requires $O(N^2)$ operations. As this would take too much time for the larger sets, I decided to use a randomizer to draw a subset M of particles in this case and do the computation only for this subset, reducing the number of computations to $O(N \cdot M)$. The number of samples M that is used to determine the interparticle distance for large sets is currently set to be $R \cdot \log N$, where the factor R has a default of 10.

- A comparable problem as with the interparticle distance information occurs for the computation of the initial acceleration (read potential energy) to be able to determine the average displacement. This was solved by using a Tree-Code like algorithm to approximate this acceleration as soon as the number of particles exceeded a certain threshold. The Θ factor for the Tree-Code part is currently set to 1 by default.[3]

- The average displacement contains important information about the system being collisionless or not. But suppose the system as a whole would be translating with a high speed. As this could make the information totally useless, I decided to always transpose the problem to the Center-Of-Mass system. This of course sets the total momentum to 0 in both directions.

- The meshspacing for the autocorrelation function is important : if it is too large, the function may contain large fluctuations and if it is too small we loose information (this is in fact exactly what happens if the large random sets are compared with the meshes !).
  Also the overall meshsize is important: if the dimension is too small this may cause aliasing, while if it is too large we loose information as the autocorrelation drops very fast in that case.

- The threshold for the distinction between PP and PM/TC is currently set to 100 particles.

6 Restrictions

Although this analyzer heuristic seems to give reasonable clear indications at this time, it isn't perfect and it may still contain many flaws, as it is still very young and only little tested.

---

[3]Note that my definition for Θ differs slightly from the one given by Barnes-Hut (see [Lemmens])

Also many of the used characteristics need further investigation. Nevertheless : it is much better than trying to run a set through all 3 methods and see which one is best, if it is kept in mind that the results only give rough indications.

As already said earlier in this article : especially the sets that are somewhere in the "border-land" need further examination before the final choice for a certain method is made.

References

[BarnesHut] J. Barnes and P. Hut, *A hierarchical O(N log N) force calculation algorithm*, (Nature, vol. 324, p. 446- , Dec 1986).

[Greengard] L. Greengard and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, (Journal of Computational Physics, v 73, 1987).

[Graps] A. Graps, *Amara's Recap of Particle Simulation Methods*, (http://www.amara.com/papers/nbody.html, 1996 (taken from the WWW)).

[Hockney] R.W. Hockney and J.W. Eastwood, *Computer Simulations using Particles*, (Bristol: Institute of Physics Publishing, 1994).

[Lemmens] C.W.J. Lemmens, *An investigation, implementation and comparison of 3 important particle simulation techniques*, (Delft: Reports of the faculty of TWI no. 97-46, 1997)

[Loan] C. van Loan, *Computational Frameworks for the Fast Fourier Transform*, Vol. 10 in series "Frontiers in Applied Mathematics", (Philadelphia: Society for Industrial and Applied Mathematics, 1992).

[MacNeice] P. MacNeice, *Particle-Mesh Techniques*, (Greenbelt: Goddard Space Flight Center (NASA), 1996 (taken from the WWW)).

[Salmon] John K. Salmon and Michael S. Warren, *Skeletons from the Treecode Closet*, (Journal of Computational Physics, July 31, 1992; revisited March 10,1993).

FLUID-DYNAMICS SIMULATION OF COMPLEX GEOLOGICAL PHENOMENA BY CELLULAR AUTOMATA METHODS

S. Di Gregorio, R. Rongo, W. Spataro
Dept. of Mathematics, Univ. of Calabria, Arcavacata, I 87036, Rende (CS), Italy
(dig,rongo,spataro)@unical.it

Abstract

Parallel Computing models sometime represent a valid alternative to standard methods with differential equations in modelling complex phenomena. Cellular Automata, a paradigm of parallel computing, permit such an alternative approach for modelling and simulating some kind of complex systems, which can be described in terms of local interactions of their constituent parts; furthermore Cellular Automata are easily and naturally implementable on parallel computers and exploit the power of parallelism without significant limits. Lava flows and debris/mud flows can be viewed as dynamical systems based on local interactions. Two cellular models, SCIARA and SCIDDICA, based on similar principles, are shown, they were developed in order to simulate respectively lava flows and debris flows; the models were validated on real events: we present the cases of the first phase of the 1991-1992 Etnean eruption and the 1984 Ontake volcano debris avalanche in Japan. The comparison between the real and simulated events is satisfying within limits to forecast the flow path and the surface covered by the flows.

1. Introduction

The behaviour of many complex phenomena was investigated not so long ago only from a qualitative viewpoint, when the formal models, describing them, were so complex that the main (at that time) computational modality, represented by the integration of differential equations, was impracticable.

Consequently to the development of Computer Science in these years, the applicability limits have been elevated considerably because of the continuous rise of computing power; at the same time researches in Parallel Computing evidenced the relevant potentialities of Parallel Computing paradigms to represent a valid alternative to Differential Calculus in the description of very complex phenomena [Toffoli, 1984].

Cellular Automata (CA) capture the peculiar characteristics of the acentric systems, i.e. systems, whose evolution can be described considering exclusively the local interactions of their constituent parts; CA guarantee computational universality, furthermore, many aspects of modelling with CA have been widely investigated [Vichniac, 1984].

A CA can be considered intuitively as a d-dimensional Euclidean space, the cellular space, partitioned into cells of uniform size, each one embedding an identical finite automaton, the elementary (ea). Input for each ea is given by the states of the ea in the neighbouring cells, where neighbourhood conditions are determined by a pattern invariant in time and constant over the cells. At the time t=0, ea are in arbitrary states and the CA evolves changing the state of all ea simultaneously at discrete times, according to the transition function of the ea.

Applications of CA are very broad, they range from the microscopic simulation of physical and biological phenomena to macroscopic simulation of geological and social processes.

Lava flows and debris flows seem to be a good a CA application field because they can be considered as an acentric system; their governing flow equations (e.g. the Navier-Stokes equations) cannot be easily solved without making substantial simplifications. Particular complexity arises for the irregular ground topography and because such flows can range rheologically from nearly Newtonian fluids to brittle solids (lava flows because of cooling, debris flows by water loss)[Johnson, 1973].

The physical systems embedding lava flows and debris flows are given here in terms of a two-dimensional CA, where the states of the square cells describe the physical characteristics of the corresponding space portion: altitude (the third dimension is implicit), lava/debris/mud thickness, outflows, temperature, water content etc..

This paper presents jointly the main lines of SCIARA [Barca et al., 1993, 1994; Crisci et al. 1997] for lava flows and of SCIDDICA [Di Gregorio et al. 1994 a,b, 1997] for debris flows and the result of their validation, which are satisfying in terms of spatial and temporal evolution of flow path.

2. Main characteristics of the Cellular Automata model

The cellular automaton model is the quadruple $<R_2, X, S, \sigma>$ where

- $R_2 = \{(x, y)| x, y \in N, 0 \le x \le l_x, 0 \le y \le l_y\}$ is the set of points with integer coordinates in the finite region, where the phenomenon evolves. N is the set of natural numbers.
- The set X identifies the geometrical pattern of cells which influence the cell state change. They are, respectively, the cell itself and the "north", "south", "east" and "west" neighbouring cells: $X = \{(0,0), (0,1), (0,-1), (1,0), (-1,0)\}$;
- The finite set S of states of the ea:

$S = S_a \times S_m \times S_v \times S_f^4$

where the substates are:

S_a, correlated to the altitude of the cell;

S_m, correlated to the thickness of material (lava, debris, mud) in the cell;

S_v, correlated to the factor of viscosity variation (lava temperature, water content) of the cell;

S_f, correlated to the flows toward the only four neighbourhood directions.

- $\sigma : S^5 \to S$ is the deterministic state transition for the cells in R_2, it will be sketched in the next section.

The elements of S_a are integers, which express the value of the altitude (dm); the elements of S_m are integers, which represent the quantity of material inside the cell, expressed as thickness in dm; the elements of S_v are integers, which express the factor of viscosity variation (lava temperature as tenths of Kelvin degrees or water content); the elements of S_f are integers, which express the outflow rate as material thickness in dm.

At the beginning of time we specify the states of the cells in R_2, defining the initial configuration of the CA, of course initial values of substates S_f are zero. At each next step the function s is applied to all cells in R_2, so the configuration is changed in the time and the evolution of the CA is obtained.

Initial conditions are given by the morphology and for the landslide by the determination of the detached mass and its water content; eruptions need lava flows from the cells, corresponding to the vents, at each CA step.

3. Main characteristics of s

The main mechanisms of the transition function concern the computing of the "outflows" from the single cell toward the cells with common sides and the "viscosity" of the fluid in the cell; they are illustrated by the following Pascal-like procedures, which calculate the values of the cell substates at time t+1 according to the values of neighbouring cells states at time t. Indexes 0, 1, 2, 3, 4 are used for the cell itself and neighbours "north", "east", "west", "south" respectively.

Obviously the following procedures represent only the basic outline of the algorithms.

3.1 S_h

The new cell thickness 'new_th' at time t+1 is given by the thickness 'th' at time t plus the inflows 'inflow[i]' from the neighbour cell i minus the contribution determined by lava outflows 'outflow[i]' from the cell toward the neighbour cell i at time t. Since the dimensions of the cell surface are specified constant, the amount of lava accumulated can be described in terms of changes in thickness.

$$new\_th = th + \sum_{i=1}^{4} (inflow[i] - outflow[i])$$

Note that the thickness may change also by solidification according to the next paragraph.

The cell altitude 'alt' remains unchanged until lava or debris/mud begins to solidify; when the viscosity factor 'visc_fact' (temperature for the lava, water content for debris/mud) becomes equal or less than a threshold 'threshold', the altitude is increased by the thickness 'th' of the cell; at the same time the cell thickness is reset to zero.

```
procedure new_altitude..........;
begin
   if visc_fact = threshold then
      begin
         new_alt := alt + th;
         th := 0
      end
   else new_alt := alt
end;
```

Fig. 1: *Procedure new_altitude.*

3.3 S_v

Changes in temperature or water content are modelled as a two-step process: The first step simplifies the real situation by averaging the viscosity factor 'visc_fact' considering the residual matter 'res_matter' in the cell and the incoming flows 'inflow[i]' of its neighbours.

$$\text{aver\_visc} = \left(\text{visc\_fact}[0] \times \text{res\_matter} + \sum_{i=1}^{4} \left(\text{visc\_fact}[i] \times \text{inflow}[i] \right) \right) \Big/ \left(\text{res\_matter} + \sum_{i=1}^{4} \text{inflow}[i] \right)$$

The second step estimates the variation of the viscosity factor according to usual methods applied to the regular geometrical conditions of the cell.

3.4 S_f

The outflow from a cell (or alternatively part of the inflow into an adjacent cell) depends on the hydrostatic pressure gradients across the cell due to differences in matter thicknesses compared with neighbouring cells, and on the rheology of the lava in the cell. The effects of variation in the matter thickness are accounted for by minimising the total height difference (cell altitude plus matter thickness) between a cell and its four neighbours after each time interval.

The rheological resistance is strongly dependent on the viscosity factor. Because of complexities inherent in specifying the rheology and its variation, we have chosen to model rheological resistance in terms of an "adherence parameter" v, which represents the amount of matter (expressed as a thickness, as discussed above) that cannot flow out of a cell because of rheological resistance. v is assumed to vary with the viscosity factor (F) according a simple inverse exponential relation $v = ae^{-bF}$, where a and b are constants describing rheology. In the distribution algorithm the input data are the values at time t of the heights 'z[i]', which are given by altitude plus "adherence" for the cell and altitude plus lava thickness for its neighbours, la is the lava amount to be distributed.

The cells, where the lava cannot flow in, are individuated initially considering the average height given by:

$$\text{av\_height} = (\text{la} + \sum_{i=0}^{4} z[i]) / 5$$

where ; 'z[i]' greater than 'av_height' indicates that lava cannot flow towards 'i'; so 'i' must be eliminated from the distribution and from 'av_height' computations.

This computation is iterated with the remaining cells, calculating the new 'av_height' and eliminating cells with 'z[i]' greater than 'av_height' until there are no more cells to be eliminated; eventually the quantity 'av_height'-'z[i]' for the remaining neighbour cells represents the lava inflow to the neighbour cell i.

Let us consider the following example: z[0] = 2, z[1] = 30, z[2] = 7, z[3] = 13, z[4] = 3; la = 15
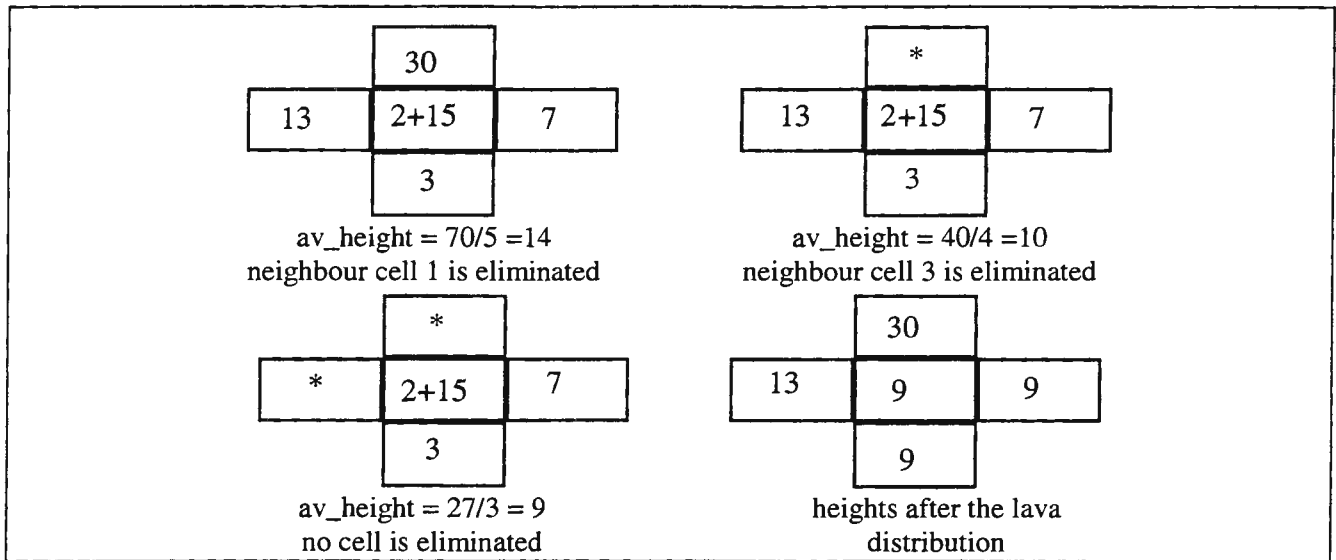
Fig. 2: *Distribution example.*

Furthermore we assume that flow can occur between the cell and the i-th neighbour, in some case, only if the following condition is verified: tan theta ≥ threshold where 'theta' is the local slope angle between the cell and the i-th neighbour, 'threshold' is a value determined by friction conditions.

The following Pascal-like procedure illustrates better the algorithm; 'eliminated[i]' is a logical variable true when cell i is eliminated; 'z_sum' specifies the sum of the heights plus the quantity of lava to be distributed; 'new_control' is a logical variable true, when a new 'av_height' must be calculated; 'outflow[i]' is the outflow towards the neighbour cell i, 'count' counts the remaining cells.

```
procedure distribution;.......................
begin
  for  i := 0 to 4  do
     eliminated[i] := tan(theta) < threshold;
  repeat
     new_control := false;
     z_sum := 1;
     count := 0;
     for i := 0 to 4  do
        if not eliminated[i] then
           begin
              z_sum := z_sum + z[i];
              count := count + 1
           end;
     av_height:= z_sum / count;
     for i := 0 to 4  do
        if (z[i] > av_height) and (not eliminated[i]) then
           begin
              new_control := true
              eliminated[i] := true
           end
  until not new_control;
  for i := 1 to 4 do
     if eliminated[i] then
        outflow[i] := 0
     else outflow[i] := av_height - z[i];
end;
```

Fig. 3: *Procedure distribution.*

4. Applications

Several versions of SCIARA and SCIDDICA, based on the previous model, had been developed on different computer types, from PC and McIntosh to a network of 32 transputers [Di Gregorio et alii, 1996]. We present the cases of the first phase of the 1991-1992 Etnean eruption and the 1984 Ontake volcano debris avalanche in Japan. The comparison between the real and simulated events is satisfying within limits to forecast the flow path and the surface covered by the flows.

4.1. Simulation of the first period of the 91-93 Etnean eruption.

The eruptive story of the 1991-93 event [Calvari et al., 1994] can be divided into two periods, the first undoubtedly the most paroxytic, characterised by high average values of flow rate (18-25 mc) that started the 15th of December 1991 and ended about the 20th of January 1992. After that date, besides a rapid decrease of the flow rate, a human intervention on the lava event started to prevent that the lava flows, that already covered 7 km, could reach the town of Zafferana situated 9 km from the vents. Such interventions, that started with the construction of a ground obstruction near Val Calanna, have in fact modified the natural evolution of the lava phenomena.

In the present work only the first part of the eruptive event was simulated, where the flow was not manipulated by human interventions, during which about 80% of the entire eruptive story was generated; the simulation regards only the first 39 days of the eruption.

The comparison between the real event (the first 39 days of the 1991/93 eruption in fig.4) and the SCIARA simulation (fig.5) can be considered satisfactory in relation to geometric characteristics (thickness, area, form) and evolutionary history, despite the approximations in the input data. The comparison between the real and simulated (fig.6) events is satisfying within limits to forecast the flow path and the surface covered by the flows.
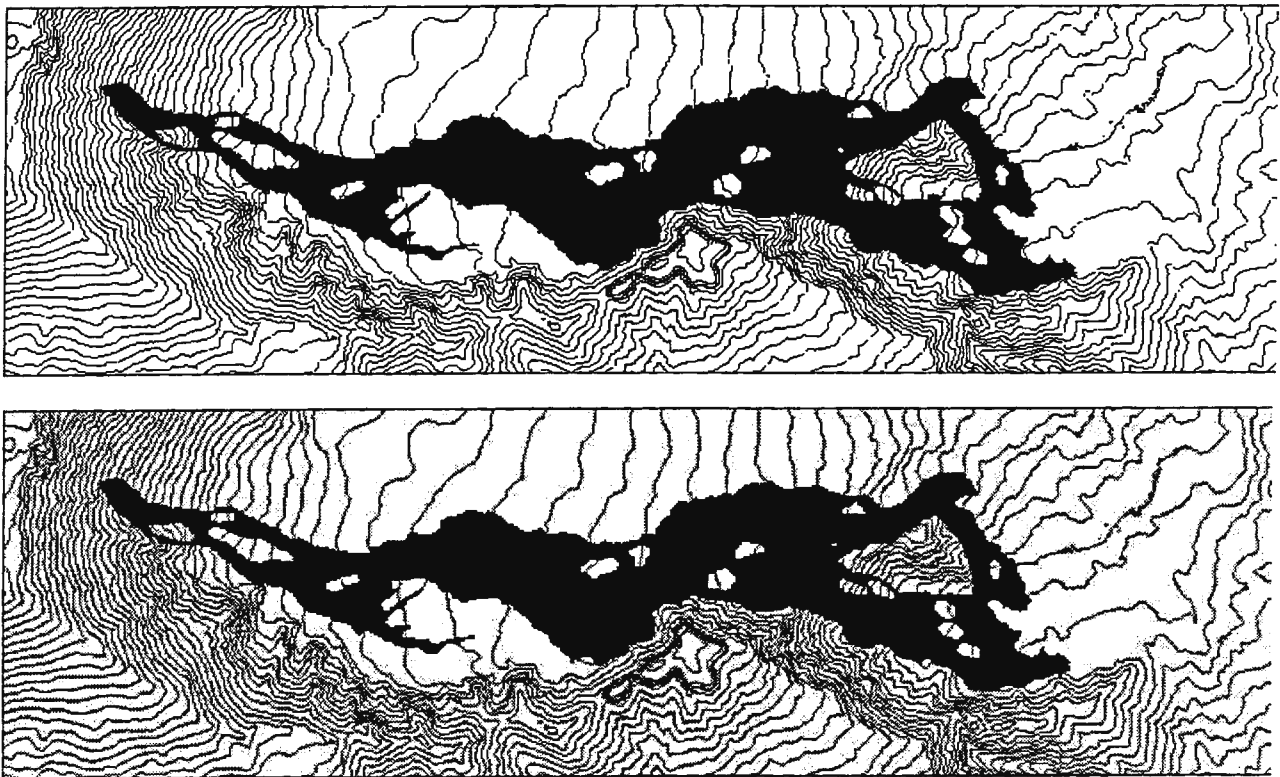
Fig. 5: *First phase of the 1991/93 Etnean eruption.*
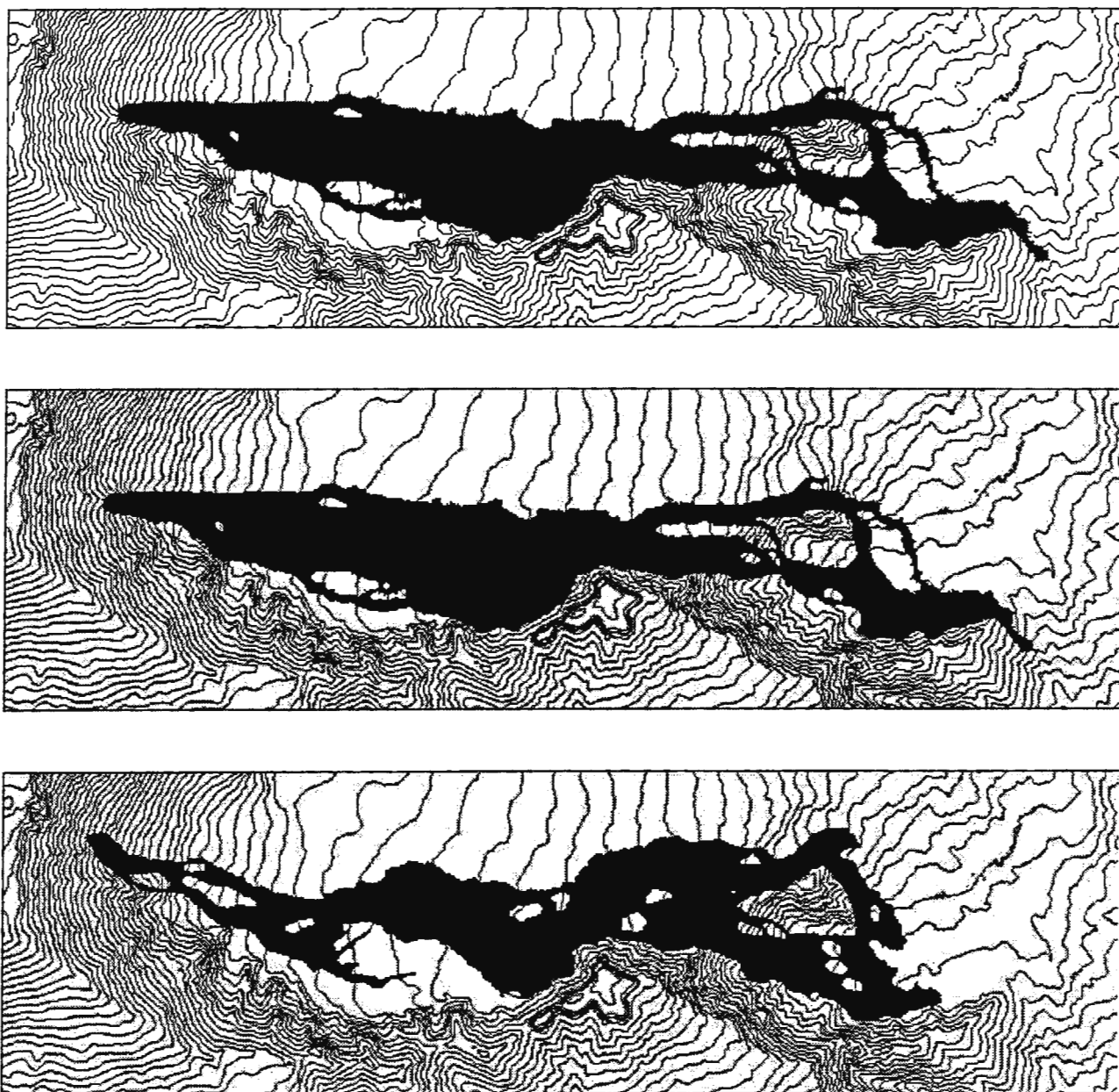
Fig. 6: *First phase of the 1991/93 Etnean eruption simulation with SCIARA.*

4.2 The Ontake Volcano Landslide Simulation

The Mount Ontake landslide was chosen as study case because of the large base of information and data before, during and after the event [Sassa, 1988].

A giant landslide (fig. 6) was triggered on the slope of Ontake volcano (Japan) by an earthquake in 1984 and moved along the Denjo river at about 20÷26 m/s over 9 Km, with a relief of 1625 m. The anomalous extension and speed is explained by the water saturated ground; it justifies that the debris water content is considered constant and the adherence is considered zero.
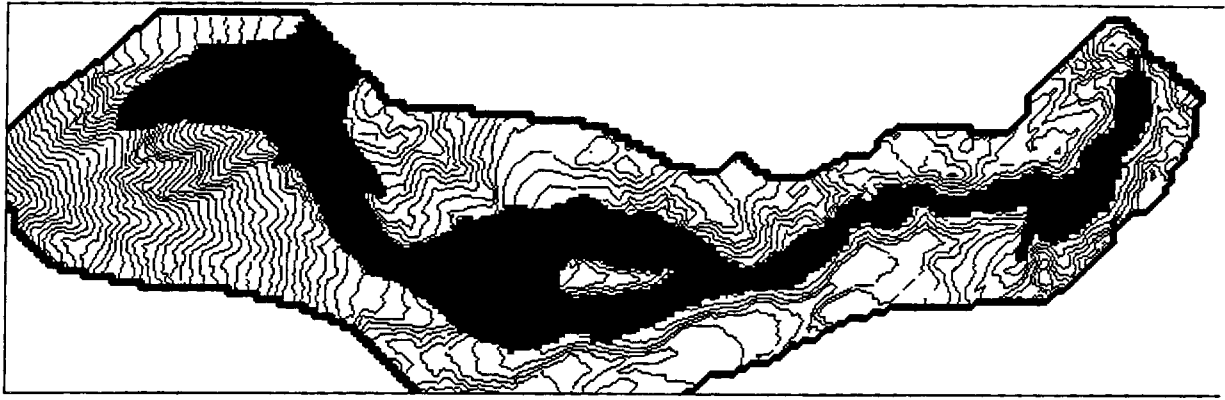
Fig. 6: *The Mount Ontake landslide*

Fig.7: *The Mount Ontake landslide simulation with SCIDDICA*

5.Conclusions

The models SCIARA and SCIDDICA can be considered working well in this phase, but more tests, involving all the model aspects, are necessary with more data detail and smaller cells; more precision would be gained with an hexagonal tessellation; the algorithm must be refined in order to describe particular effects.

Some applications could be given by this type of simulation; the most interesting ones concern the fields of intervention against the risk:

a) forecasting of the area interested by the landslide considering various initial water contents in order to locate potential risk areas and to permit the creation of microzonal maps of risk;

b) the possibility to precisely follow the progress of an event and predict its evolution;

c) the verification of the possible effects of human intervention on real or simulated flows in stream deviation. Introduction of data, which represent alterations of the original conditions or of the present ones (e.g., the construction of a canal or embankment), is possible during the simulation. The program can check the effects on the flow and allows corrections where necessary.

Aknowledgement

References

Barca D., Crisci G.M., Di Gregorio S., Nicoletta F.P., 1993: *"Cellular Automata methods for modeling lava flow: simulation of the 1986-1987 etnean eruption"* Chapt. twelve of Active Franas, Kilburn C. and Luongo G. Eds., 283-301, UCL Press London.

Barca D., Crisci G. M., Di Gregorio S., Nicoletta F.P., 1994: *"Cellular Automata for simulating lava flows: a method and examples of the Etnean eruptions"*, Transport Theory and Statistical Physics, 23 (1-3), pp. 195-232.

Calvari S., Coltelli M., Neri M., Pompilio M., Scribano V. 1994 *"The 1991-93 Etna eruption: chronology and lava flow-field evolution."*Acta Vulcanologica - Vol.4 p 1 -14.

Crisci G. M., Di Francia A., Di Gregorio S., Nicoletta F.P., Rongo R., Spataro W., 1997: *"SCIARA.2: A Cellular Automata Model For Lava Flow Simulation"* - Ed. V.P. Glahn Proceedings of the Third Annual Conference of the International Association for Mathematical Geology IAMG 22-27 September, 1997, Barcelona, Spain, Addendum pp 11-16, 1997

Di Gregorio S., Nicoletta F.P., Rongo R., Sorriso-Valvo M., Spataro W., 1994a: *"A two-dimensional Cellular Automata Model for Landslide Simulation"*, Proceedings of 6th Joint EPS-APS International Conference on Physics Computing "PC'94" Lugano, Switzerland 22-26 August 1994 Ed.s R.Gruber and M.Tommasini, pp. 523-526.

Di Gregorio S., Nicoletta F.P., Rongo R., Sorriso-Valvo M., Spataro W., Spezzano G., Talia D., 1994b: *"Landslide Simulation by Cellular Automata in a Parallel Environment"*, Proceedings of 2nd International Workshop on Massive Parallelism: Hardware, Software and Application October 3-7 1994 Capri Italy, pp 392-407.

Di Gregorio S., Nicoletta F.P., Rongo R., Siciliano C., Sorriso-Valvo M., Spataro W. – *"SCIDDICA-3: A Cellular Automata Model for Landslide Simulation"* in Advances in Intelligent System, F.C. Morabito ed. - pp. 324-330, IOS Press Amsterdam 1997.

Di Gregorio S., Rongo R., Spataro W., Spezzano G., Talia D., 1996: *"A Parallel Cellular Tool for Interactive Modeling and Simulation"*, IEEE Computational Science & Engineering, vol. 3, no. 3, pp. 33-43, Fall 1996

Johnson A.M. 1973: *"Physical Processes in Geology"*, Freeman, Cooper & Company, San Francisco CA.

Sassa K. 1988: *"Motion of Landslides and Debris Flows"*, Report for Grant-in-Aid for Scientific Research by Japanese Ministry on Education, Science and Culture (Project No 61480062)

Toffoli T. 1984: *"Cellular Automata as an alternative to (rather than an approximation of) differential equations in modeling physics"*, Physica 10D, 117-127.

Vichniac G. 1984: *"Simulating physics with cellular automata"*, Physica 10D, 96-115.

Prediction of spatially correlated coastal erosion and morphodynamic evolution processes

H. Jansen[1] and D.W. Dunsbergen[2]

[1] Dept. of Information Science and Systems, Delft University of Technology, P.O.Box 5031, 2600 GA, Delft, The Netherlands. H.Jansen@its.tudelft.nl
[2] RWS, National Institute for Coastal and Marine Management/RIKZ, P.O.Box 20907, 2500 EX, Den Haag, The Netherlands. D.W.Dunsbergen@rikz.rws.minvenw.nl

Abstract

Coastal erosion processes are complex, nonlinear processes exhibiting features characterized by a multiple of scales over space and time [3]. Although the insight in the physical mechanisms of these morphodynamic phenomena is growing, e.g. [8, 6, 6], this knowledge alone, however, is yet insufficient to make successful predictions of the migration of the coast line and bed forms in the surf zone. In this study we follow the opposite path and start from the observed evolution which has been measured and collected for the Dutch coast during the last 35 years. Our study begins with identifying a linear, low-order state-space model which assumes a separation of the spatial and temporal (dynamic) characteristics. A model reduction in L_2-norm ensures that only a relatively small number of characteristic states have to be identified. The temporal behavior is represented by the frequency (time-scale) and exponential decay (damping) of the spatial modes, the so-called POPs (*Principal Oscillation Patterns*). Thus, the most dominating and persistent POPs can be identified. The degree of damping is important for the quality of the prediction which can be achieved. Further, with additional insight, this concept can be generalized to nonlinear state-space models and include additional rules (laws, constraints) that should be satisfied. The POP method is already successfully applied for daily weather forecasting [4].

Introduction

The morphodynamic evolution of the coast can be conceived of as a spatially distributed random dynamic process, partially driven by the wave and current climate and partially governed by internal feedback mechanisms. Near to the shoreline, waves are breaking on the breaker bars, "plunge into the water" picking up sand which is carried onto the beach. At the same time a constant undertow current carries sand from the beach into the water. This mechanism (among others) explains the oscillatory behavior of sand particles from the sea-bed, onto the beach and back again. The time-scale of this and related processes is in the order of seconds.

Sand on the sea-bed is also transported by tidal (hence steady) currents. These currents can be cross-shore, long-shore or can have components in both directions. The time-scales involved are in the order of hours.

The net sand transport caused by all these effects modifies the sea-bed which in turn modifies the flow pattern etc. This explains the feedback mechanism of the sea-bed. At a macroscopic scale, the sand particles group themselves in sand bar formations. Under normal circumstances the migration of these sand bars is in the order of months to years. Sometimes, however, this is shortened to weeks, especially when weather conditions are severe for a longer period of time. Finally, beach nourishments interrupt the evolution of the natural process.

The prediction of bar migration on a yearly scale or more is of particular interest for the Dutch coastal maintenance program-me which is based on the principle of "dynamic preservation" [5], i.e. the utilization of natural processes along the coast wherever possible. Such predictions make only sense if the bar evolution is dominated by rhythmic, quasi-periodic features, rather than being a purely random process. In that case one may try to identify a dynamic model which captures these dominant features, leaving the unknown and more intricate aspects of the evolution outside the model and consider them as noise. The prediction then is made on basis of the hypothesis that the dominant dynamic components in the evolution can be "extrapolated" (in a dynamically consistent way) into the near future. This research is part of testing and verifying the validity of such hypothesis.

Identification of the prediction model

Assume the measurements of the morphodynamic evolution process be represented by the N-dimensional time-series $\{\mathbf{X}_{t_k}, t_k = t_0, \ldots, t_{K-1}\}$. When the process is sampled at equidistant time points Δt, one can assume that $\{t_k = t_0 + k\Delta t, k = 0, \ldots, K-1\}$ where we let $t_0 = 0$ and $t_{K-1} = (K-1)\Delta t = T$. The dimension N of this process is generally large. However, the number of dominant features M (which govern most of this dynamics) is often of low dimension: $M \ll N$. Hence, it can often be assumed that the evolution process $\{X_k\}$ can be approximated by a linear combination of suitably chosen time-invariant patterns $\mathbf{p}_j, j = 1, \ldots, M$, the interaction of which should satisfy certain dynamic constraints:

$$\mathbf{x}_k = \sum_{j=1}^{M} \tilde{x}_{k,j}\mathbf{p}_j \equiv P\tilde{\mathbf{x}}_k \approx \mathbf{X}_k \quad \text{(model reduction)} \tag{1}$$

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{g}_k(\tilde{\mathbf{x}}_k; \boldsymbol{\alpha}) + \tilde{\boldsymbol{\xi}}_k \quad \text{(dynamic constraint)} \tag{2}$$

where the N-dimensional patterns $\mathbf{p}_j$ and coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{N_\alpha}$ need be identified. Further, the following remarks can be made:

- In the first equation, a separation between the temporal and spatial dependencies is enforced. By taking $M \ll N$, a significant *model reduction* c.q. *data compression* can be achieved;

- Matrix $P = |\mathbf{p}_1, \ldots, \mathbf{p}_M|$ stores the time-invariant, so-called, *Principal Interaction Patterns* (PIP). The corresponding *PIP coefficients* $\tilde{x}_{k,j}$ are time-dependent amplitudes and stored in $\tilde{\mathbf{x}}_k$. The PIP model was introduced in [2];

- In the second equation, the pattern coefficient time-series $\{\mathrm{x}_k, k = 1, \ldots, K\}$ are made to satisfy a difference equation. This difference equation defines the dynamical relationship (interaction) between the PIP-coefficients (patterns). It will have to satisfy the properties of a realistic physical system.

  That part of the measurements that can not be identified and incorporated in the approximate model is left to the residual noise vector $\tilde{\xi}$. In the optimal case $\tilde{\xi}_k$ is a white noise vector signifying that no unidentified dynamics has been left unmodeled.

  As a result, the *dynamic constraint* ensures that the evolution of the PIP coefficient time-series will have to be smoother and much more regular than the random process $\tilde{\xi}_k$;

- The model class $\mathbf{g}_k$ must be specified beforehand: it can be linear or non-linear in $\tilde{\mathrm{x}}_k$. The parameters $\boldsymbol{\alpha}$ have to be estimated in the identification procedure;

- Although, in general, $\mathbf{X}_k$ could (and sometimes should) be a complex time series (i.e. the measurements and their rate of chance or 'velocity' c.q. momentum), this situation is not accounted for in the present situation [1];

- Once the dynamic model $\mathbf{g}$ has been identified, predictions can be made by computing the state $\mathrm{x}_{k+1}^{\tilde{}}$ from the previous state $\mathrm{x}_{k+1}^{\tilde{}}$.

Identification procedure

The algorithm for the identification procedure is obtained in two minimalization steps. First, assuming a set of PIPs, P, be given, the corresponding PIP coefficients, $\tilde{\mathrm{x}}_k$, for each time step are obtained by minimization of the square modulus of the residual process, ρ_k:

$$\min_{\tilde{x}_k} \rho_k^H M \rho_k \quad , \quad \rho_k = \mathbf{X}_k - \mathrm{x}_k \tag{3}$$

where M is a weight matrix taken equal to, for instance, the identity matrix (no weighing at all) or to the lag-0 covariance matrix. Partial differentiation of (3) with respect to $\mathbf{X}_k$ and setting the result equal to zero, yields the identity: $P^H M X_k = (P^H M P)^H x_k = P^H M P x_k$ so that the PIP coefficients can be identified from:

$$\tilde{\mathrm{x}}_k = Q^H M \mathbf{X}_k \quad , \tag{4}$$
$$Q = P N^{-1} \quad , \quad N = P^H M P$$

In general, the PIPs are not mutually orthogonal. Orthogonality is obtained with regard to the, so-called, adjoint PIPs (APIPs) Q: Pre-multiplying the (zero-mean) residual ρ_k with $Q^H M$ and taking the expectation, we obtain

$$\mathbf{0} = Q^H M \langle \rho_k \rangle = Q^H M \langle \mathbf{X}_k - P \tilde{\mathrm{x}}_k \rangle$$
$$= (I - Q^H M P) \langle \tilde{\mathrm{x}}_k \rangle \quad \Rightarrow \quad Q^H M P = I \tag{5}$$

which establishes the orthogonality between the PIPs and APIPs.

The unknown model parameters, $\boldsymbol{\alpha}$, and the PIPs, $\mathbf{p}_j$, can now be determined simultaneously by minimalization in L_2-norm the expected difference between velocities of the measured and approximate process, $\min_{\boldsymbol{\alpha},\,\mathbf{p}_j}\langle\dot{\boldsymbol{\rho}}_k^H\acute{M}\dot{\boldsymbol{\rho}}_k\rangle$ where $\acute{M}$ is a weight matrix (equal to the identity matrix or to the inverse of the covariance of the velocity process, $(C_0^{\dot{\mathbf{X}}})^{-1} = \langle\dot{\mathbf{X}}_k\dot{\mathbf{X}}_k^H\rangle^{-1}$). Assumedly, with measurements at equidistant sample points, the time step can be left out of the equation and the minimalization becomes

$$\min_{\boldsymbol{\alpha},\,\mathbf{p}_j}\left\langle\boldsymbol{\eta}_k^H\acute{M}\boldsymbol{\eta}_k\right\rangle \quad, \tag{6}$$

$$\boldsymbol{\eta}_k = \mathbf{X}_k - \mathbf{X}_{k-1} - (\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_{k-1})$$
$$= \mathbf{X}_k - \mathbf{X}_{k-1} - P\left[\mathbf{g}_{k-1}\left(\tilde{\mathbf{x}}_{k-1}(P);\boldsymbol{\alpha}\right) - \tilde{\mathbf{x}}_{k-1}(P)\right]$$

where $\tilde{\mathbf{x}}_k(P)$ is defined by (4). Partial differentiation of (6) with respect to the optimizing parameters yields a considerable set of $N_\alpha + KN$ nonlinear algebraic equations which need be solved for the set $\boldsymbol{\alpha}$ and the vectors $\mathbf{p}_j, j = 1,\ldots,K$ respectively. The solution of this system of equations is non-trivial as multiple solutions usually exist, even if a linear system is adopted, i.e. $\mathbf{g}_k = A\tilde{\mathbf{x}}_k$, similarity transformations between dynamically identical systems exist. However, with the adoption of a linear system we can make an assumption which greatly simplyfies the ensuing identification procedure. First, the model reduction will be addressed.

EOF-based model reduction

With neglect of dynamic constraints, it is common practice to approximate the measurement time-series as a sum of the most dominant *Empirical Orthogonal Functions* (EOF):

$$\mathbf{x}_k = \sum_{j=1}^{L}\breve{x}_{k,j}\mathbf{f}_j \equiv F\breve{\mathbf{x}}_k \approx \mathbf{X}_k \quad,\quad L \leq N \tag{7}$$

where the EOF patterns $\mathbf{f}_j$ are the eigenvectors of the lag-0 covariance matrix $C_0^{\mathbf{X}} = \langle X_k X_k^H\rangle$. (Because C_0 is symmetric the EOFs f_j are mutually orthogonal.) For each time step they are multiplied with their corresponding EOF coefficient $\breve{x}_{j,k}$. The order of importance (dominance) of the EOFs is determined by the amount of variance each EOF represents of the original (measured) data. In this way, a major data compression can be achieved when $L \ll N$ so that the measured data is projected on the EOF space with dimension $L \leq N$. The number of EOFs necessary to obtain a good EOF-representation of $\mathbf{X}_k$ forms an upper bound for the dimension of a good dynamic (PIP) model $\mathbf{g}_k$, that is: $M \leq L$, as the basis of the model can never be larger than that of the (compressed) data. Substitution of (7) for $\mathbf{X}_k$ in (6) slightly reduces the amount of computation.

Simplified identification: the POP method

When a linear dynamic model is assumed and a model reduction on the basis of EOFs (7) is used, the optimalization problem becomes:

$$\mathbf{x}_k = P\tilde{\mathbf{x}}_k \approx F\check{\mathbf{x}}_k \approx \mathbf{X}_k \quad \text{(EOF-based model red.)} \tag{8}$$

$$\tilde{\mathbf{x}}_{k+1} = \tilde{A}\tilde{\mathbf{x}}_k + \tilde{\xi}_k \qquad \text{(linear dynamic constraint)} \tag{9}$$

where the columns of P are the, so-called, *Principal Oscillation Patterns*, and system matrix $\tilde{A}$ replaces the parameter vector $\boldsymbol{\alpha}$. Pre-multiplying (9) with P and substituting (4) for $\tilde{\mathbf{x}}_k$, gives

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \xi_k \quad , \tag{10}$$
$$A = P\tilde{A}Q^H M \quad , \quad \xi_k = P\tilde{\xi}_k$$

Obviously, the columns of P must be the dominant eigenvectors of A. To that end, equation (10) is mapped onto the (much) smaller EOF-basis:

$$\check{\mathbf{x}}_{k+1} = \check{A}\check{\mathbf{x}}_k + \check{\xi}_k \quad , \tag{11}$$
$$\check{A} = F^H A F \quad , \quad \check{\xi}_k = F^H \xi_k$$

The assumption now is that $\check{A}$ can be identified directly from the EOF-based correlation matrices: Post-multiplying both sides of (11) with $\check{\mathbf{x}}_k^H$ and taking the expectation yields $\langle \check{\mathbf{x}}_{k+1}\check{\mathbf{x}}_k^H \rangle = \check{A}\langle \check{\mathbf{x}}_k\check{\mathbf{x}}_k^H \rangle + \langle \check{\xi}_k\check{\mathbf{x}}_k^H \rangle \Rightarrow C_1^{\check{\mathbf{x}}} = \check{A}C_0^{\check{\mathbf{x}}} + \mathbf{0} \Rightarrow$

$$\check{A} = C_1^{\check{\mathbf{x}}} C_0^{\check{\mathbf{x}}-1} \tag{12}$$

where it is assumed that the residual $\check{\xi}_k$ is uncorrelated with $\check{\mathbf{x}}_k$. In the eigenvector decomposition $\check{A} = \check{P}^H \Lambda P$, $\check{P}$ are the, so-called, EOF-based POPs. With regard to (11) this means that $A = P^H \Omega P = (\check{P}F^H)^H \Lambda (\check{P}F^H)$ and $\Omega = \Lambda$ so that the ("real") Euclidean-based POPs are obtained by $P = \check{P}F^H$.

Since, A ($\tilde{A}$ and $\check{A}$) are, in general, non-symmetric, the eigenvalues in the diagonal matrix Ω will be complex valued. These eigenvalues characterize the dynamics of the system identified, their real and imaginary parts representing the damping and frequency of oscillation of the corresponding POPs, respectively. The POPs, in general, will be complex as well, some of them may be real. However, when X_k is a real valued process, the complex POPs will occur in complex-conjugate pairs so that two real POPs can be constructed from each such pair.

Strongly damped POPs, in general, will have little potential for prediction as they will quickly disappear after a few, or even one, prediction step. Therefore, as regards the predictive power, the quality of the model identified is determined by the degree of damping (persistency, or lack thereof) of the POPs. [7] gives more theoretical background on the POP method and shows its use in meteorology.

Application: the Dutch coast

# of EOF	EOF data (%)	POP d/d (%)*
8	86	83/63
10	89	86/67
12	92	89/73
20	99	96/88
* data/change-of-data		

Table 1: Data coverage

POP	E-fold. (yr)	Per. (yr)
1R	3.63	4.96
1I	3.63	4.96
2R	1.61	6.96
2I	1.61	6.96
3R	7.52	43.76
3I	7.52	43.76
4R	1.86	16.18
4I	1.86	16.18
5	1.00	
6	0.75	

Table 2: POP (10) Characteristics

A POP analysis is applied to a subset of the JARKUS data set. At present, JARKUS consists of 33 years of data (1964-1996) of the measured surface of the beach and sea-bed in the surf-zone (a strip of approximately 800 to 1000 meters along the coast). A subsection of this strip is selected for a POP analysis (Figure 1). The POP model identification is based on the 31 years (1964-1994) of the available 33 year long time series.

The analysis was done for a model reduction based on 8, 10, 12 and 20 EOFs respectively. Table 1 shows how much in each case the original JARKUS data (measurements) was covered by the POP

Figure 1: Dutch coast: the IJmuiden-Katwijk section.

generated evolution (in terms of the coverage of its variance). The prediction power of a POP model depends on the characteristics of its POPs: their damping or *E-folding time* (the time by which a particular POP has damped out to $1/e$ of its original value) and their cycle period. The results for 10 POPs is shown in Table 2, results for 8, 12 and 20 POPs were worse. This will be explained later.

Apparently, for 10 POPs the dynamics is characterized by 4 complex POPs and 2 real ones. We observe cycle periods of dynamic phenomena in the order of 44, 16, 7 and 5 years. The identification of the 44 year cycle period must be looked at with suspicion as the underlying time-series was only 31 years long. Compared to the cycle periods the E-folding times are quite small which means that the POPs are substantially damped. This can be indicative of a poor prediction performance. Figure 3 shows the measurement for 1994 compared

to the reconstruction for that year based on 10 POPs. The resemblance is good. This was found to be characteristic for the entire reconstruction over 31 years. However, although the global structure of the patterns is shown in the prediction for 1995 (Figure 4), a much larger difference with the observation is found. This poor prediction quality is to be expected from a strongly damped POP model. Such strong damping may be identified for the following reasons:

1. poor quality of the data (measurements);

2. lack of dominant rhythmic, i.e. (quasi-) periodic features in the morpho-dynamic system — the driving factor of the wave climate may be more important than expected;

3. inadequacy of the linear POP model (the dynamics may be dominated by nonlinear interactions, including the possibility of chaos).

The solution to this problem is part of continuing research.

It was found that the prediction quality is not necessarily improved by the inclusion of more POPs (Figure 2). In the range of 8, 10, 12 and 20 POPs, the best prediction was obtained for 10 POPs. With a smaller number of POPs not all of the available dynamics in the data series is modeled. Higher order POPS ($M > 10$) will capture noise rather than rhythmic phenomena, which will improve the reconstruction but have a bad effect on the prediction.

Figure 2: Number of POPs vs accuracy.

Conclusion

A POP analysis was applied for the first time to a morphodynamic data set. It naturally derives from the general, nonlinear PIP method and the solution procedure is largely simplified with two additional assumptions: (1) a linear model, (2) model reduction based on EOFs. In comparison with the observation, the one-year-ahead prediction with the POP is not convincingly accurate. The several reasons what this may cause by were listed and subject to further investigation. Finally, it was found that, as regards the quality of the prediction, a certain optimal size for the reduced model exists: a smaller and larger model than the optimum would both lead to predictions worse.

Acknowledgments

References

[1] G. Bürger. Complex principal oscillation pattern analysis. *J. of Climate*, 6, 1992.

[2] Klaus Hasselmann. Pips and pops: The reduction of complex dynamical systems using principal interaction and oscillation patterns. *J. of Geophys. Res.*, 93(3):11,015–11,021, 1988.

[3] Wijnberg Kathelijne M. *Morphologic Behaviour of a Barred Coast over a Period of Decades*. PhD thesis, University of Utrecht, The Netherlands, 1995.

[4] Cecile Penland. Random forcing and forecasting using principle oscillation pattern analysis. *Monthly Weather Review*, pages 2165–2185, October 1989.

[5] A new coastal defence policy for the Netherlands, 1990. Rijkswaterstaat, Den Haag, The Netherlands.

[6] Howard N. Southgate and Luisa M. Beltram. Self-organized processes in beach morphology. *to appear in: J. of Coastal Res.*, 1996.

[7] Hans von Storch, Gerd Bürger, Reiner Schnur, and Jin-Song von Storch. Principle oscillation patterns: A review. *J. of Climate*, 8(3):377–400, 1995.

[8] Bakker W.T., S.C. v.d. Biezen, A.W. Heemink, and M. Knaapen. Breaker bars dynamics, predicted by kalman filtering. In *Proc. Coastal Dynamics '97, Plymouth, UK*, 1997.
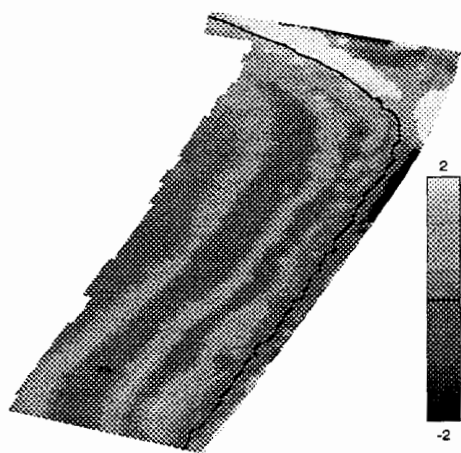
(a) JARKUS data (m)
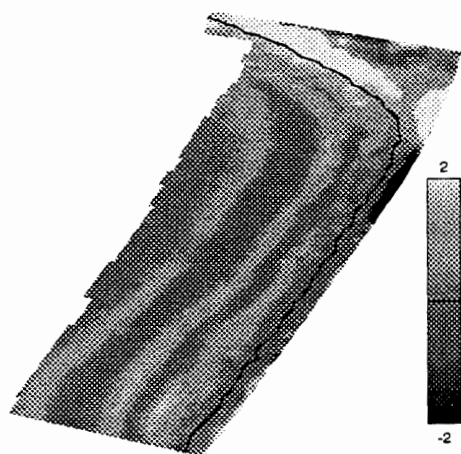
(b) POP (10) reconstruction (m)
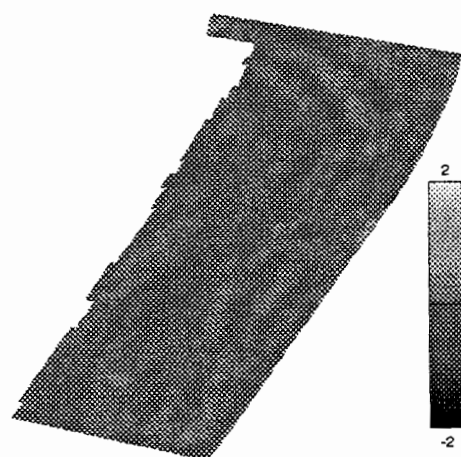
(c) difference (m)

Figure 3: Measurement vs reconstruction, 1994
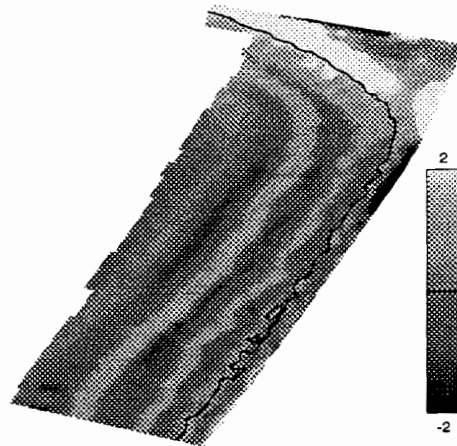
(a) JARKUS data (m)
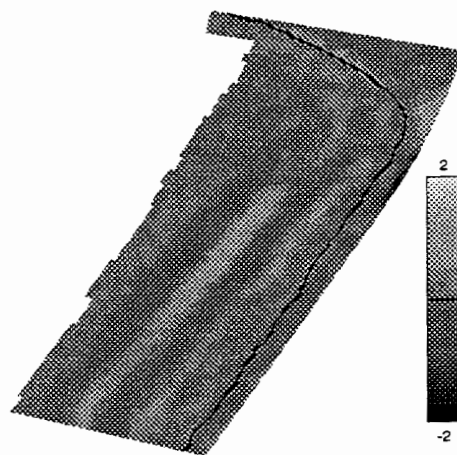
(b) POP (10) prediction (m)
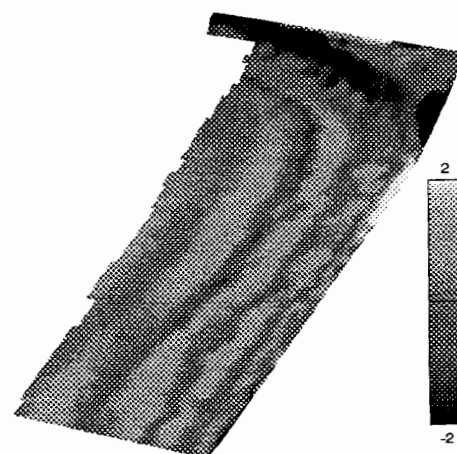
(c) difference (m)

Figure 4: Measurement vs prediction, 1995

93

The Use of CFD Simulation in the Design of Continuous Casting Tundish and Mould

S B Chin[1], J T Duncombe[2], and F Wang[1]

[1] Department of Mechanical Engineering, University of Sheffield,
Sheffield S1 3JD UK s.b.chin@sheffield.ac.uk
[2] Davy Distington Ltd., Distington House, 26 Atlas Way, Sheffield S4 7QQ UK

Abstract

In the design of continuous casting machines by Davy Distington Ltd. (DDL) the mechanical structure of the machine is the main focus of the design function. This, of course, must incorporate the various electrical, control, and support services. In addition it must take account of the critical fluid flow and heat-transfer of the liquid steel, particularly in the tundish and in the water-cooled copper mould. The fluid flow in the tundish must distribute the liquid steel as uniformly as possible to all the various outlets. The number of outlets could be one or two in a slab machine but could be more in a billet machine. It is also necessary to have only a small drop in temperature between inlet and outlets, and uniform temperature at each of the different outlets. In the mould the free surface is critical since this is where solidification starts. Hitherto, water modelling of liquid steel has been the best method of studying the flow of liquid steel but cannot easily include heat-transfer. For this reason and for its speed and adaptability, Computational Fluid Dynamics (CFD) is now being more widely used. The company is now using CFD modelling as an important analytical tool in the design process. This paper describes some ways in which these techniques have been used in the modelling of steel tundishes and moulds. The modelling of free surfaces and of argon argon injection are included with the use of sophisticated post processing techniques.

Introduction

In the design of a continuous casting machine the conversion of liquid steel at a temperature of nearly 1600 °C to a solid of the right shape and quality requires detailed knowledge of the process. At an early stage in the design process a flow and heat-transfer model is used to predict the surface and internal temperature profiles and the final point of solidification. As part of this process modelling examples are given of both tundish modelling and of mould modelling incorporating the introduction of gas bubbles. This is essential since argon is always injected to the mould to prevent the build-up of inclusions on the inside of the refractory submerged pour tube. In the engineering design of a tundish, there are specific design limits, such as size and production relative to other equipment. Within these limits, knowledge of flow patterns is crucial in advising the designers on the most appropriate design. The design can then be made to obtain the best distribution of liquid steel and in a way that achieves the cleanest product.

CFD is used as an analytical tool to help optimise the design process. Effective and efficient use of CFD techniques depends on collaboration between the CFD specialist and other engineers. This paper presents two modelling exercises, argon injection through a SEN (Submerged Entry Nozzle) in a mould and flow and heat transfer in a tundish. The effects of the argon injection were examined by carrying out a parametric study. The flow patterns in a tundish were viewed by tracing a number of streamlines. A temperature drop of the liquid steel from shroud to the exit was defined as a simple indicator of the heat problem in the design.

The Mathematical Model

The studies were conducted using real mould, tundish and SEN designs with fluid properties and boundary conditions obtained from existing continuous casting plant. A proprietary CFD package, FLOW-3D, from Flow Science, Inc. Los Alamos, has been used for all the computations.

The flow was assumed to be steady state and three dimensional. The governing equations are those for the conservation of mass, momentum (Navier Stokes) and energy. The standard k-ε model was employed to account for the effects of turbulence. Interface and free surface are calculated using the VOF method. A drift model was used in the argon injection modelling study, where liquid steel and gas bubbles are considered as a mixture of two components with different physical properties. FLOW-3D User's Manual describes fully the governing equations, numerical and solution techniques [1].

In an earlier collaboration between DDL and the University, the CFD software was developed and validated for modelling metal casting [2]. Subsequently the software has been applied to study the isothermal flow behaviour in continuous casting mould.[3,4]. This paper presents some of the results on the effect of argon injection in a mould and isothermal flow in tundish.

Argon injection in a mould

The geometry of a real SEN design was used. All crucial parameters were assigned the design values, including casting speed, immersion depth, port angle and length and mould size. As the SEN was positioned at the centre of the mould and the flow was assumed to be symmetric, only one quarter of the flow regime was modelled. This simplification significantly reduced the computational time with a small error when considering a vertical mould.

The effects of argon injection on mould flow were examined in terms of bubble size (R_o), injection rate (IR), and bubble distribution in the submerged pore tube. Each study allowed one parameter to be examined at a time. Such a parametric study ensured that the process of argon injection was understood properly. Table 1 lists the variables for the five cases of argon injection presented in this paper. Additional assumptions were made: steady state and isothermal flow with uniform bubble size. On the top of the liquid steel, a free surface boundary was defined.

Case	Descriptions	IR (L/min)	R_0 (mm)
1	Base case: no argon injection	0	-
2	Gas injected circumferentially in the SEN.	9	1.25
3	Gas bubble is smaller, uniform injection over the full cross-section.	9	0.5
4	Case 2 but with smaller gas bubble	9	0.5
5	Case 2 but with lower rate of argon injection	2.25	1.25

Table 1 Argon injection Modelling Study Variables

sothermal Flow in a Tundish

Due to geometric symmetry, only half of the tundish needed to be modelled. A maximum throughput of 6 tonnes per minute was used, which allowed flow at a speed of 2 m s^{-2} through an inlet submerged shroud of across-section area of 600 mm^2. The outlet of the shroud was submerged in the liquid. The height of the liquid steel in the tundish was assumed to be constant, and the flow to be in a steady state. A fixed heat flux of 3.5 kW m^{-2} was assigned to the tundish walls and 1.4 kW m^{-2} to its base. The flux on the top of the liquid in the tundish was assumed to be zero as it is normally insulated. This boundary was modelled as a free surface. The liquid steel was filled in at 1550 °C at the shroud.

The convergence of the solution to steady state values was verified by recording the values of variables at several critical locations, in addition to using the default convergence criteria of the software. The modelling output which is of interests to the process engineers were the flow pattern and the temperature drop of the liquid from the shroud to the exit.

Results

Argon injection in a Mould

Figure 1a shows one half of a vertical central plane of the mould, looking through the broad face. The arrows indicate the schematic flow pattern of liquid steel when no gas was added. Figure 1b shows details of the velocity field at the outlet of the SEN and the adjacent region. Due to the inertia of the fluid flowing vertically down the SEN, the stream emerging from the SEN nozzle occupies only the lower half of the nozzle. The jet flow impacts the mould narrow face wall and divides into two streams. One flows downwards and part of this flow returns to the SEN along the centre of the mould, forming a large recirculating flow occupying most of the mould. The other rises to the surface and returns to the outlet of the SEN forming a second recirculation zone. Adjacent to the outer SEN wall, a stream moves slowly down before joining the main stream exiting from the SEN via the upper half part of the outlet.
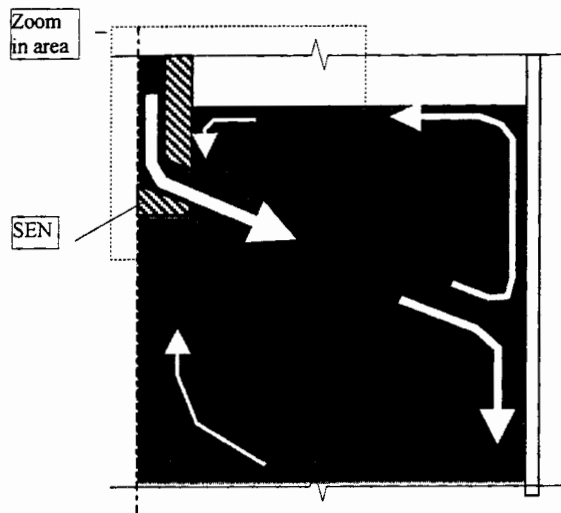
a) Schematic Diagrams of the Mould, SEN and Liquid Steel Flow.
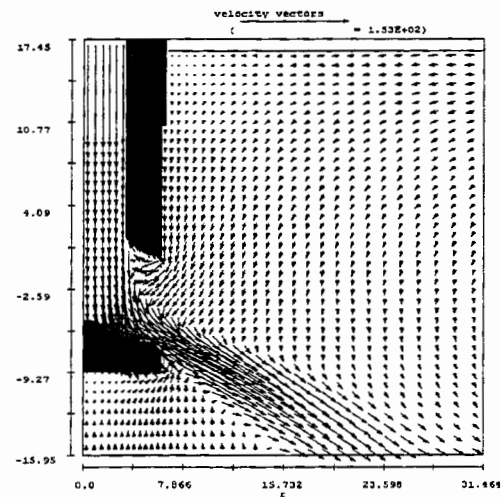
b) Details of the Velocity Field Around the SEN Outlet

Figure 1. Mould Filling Without Argon injection.

The addition of a gas changes the flow pattern, particularly in the region around the SEN nozzle outlet. When the gas is injected circumferentially around the SEN, it tends to mix and flow only with the liquid adjacent to the inner wall of the SEN and then accumulate at the top of the SEN nozzle. This can be seen in Fig. 2a where the liquid fraction at the top of the SEN nozzle is as low as 0.2 (Fig. 2a). When the liquid in this region is saturated, bubbles float to the surface resulting in a small upwards stream next to the SEN outer wall(Fig. 2b).

The gas stream is driven by buoyancy force the magnitude of which depends on the bubble size. The smaller the bubbles, the weaker the force, and therefore, the longer the bubbles travel with the main stream. Figures 3a and 3b show the results of CASE 5, where all the conditions are the same as in CASE 2 but the bubble is now one quarter of the size of that in CASE 2. These smaller bubbles travel further downstream with the jet flow issuing from the nozzle (Fig 3a) and they lift the jet stream slightly (Fig. 3b).

a) Liquid Fraction Contours   b) Velocity Field
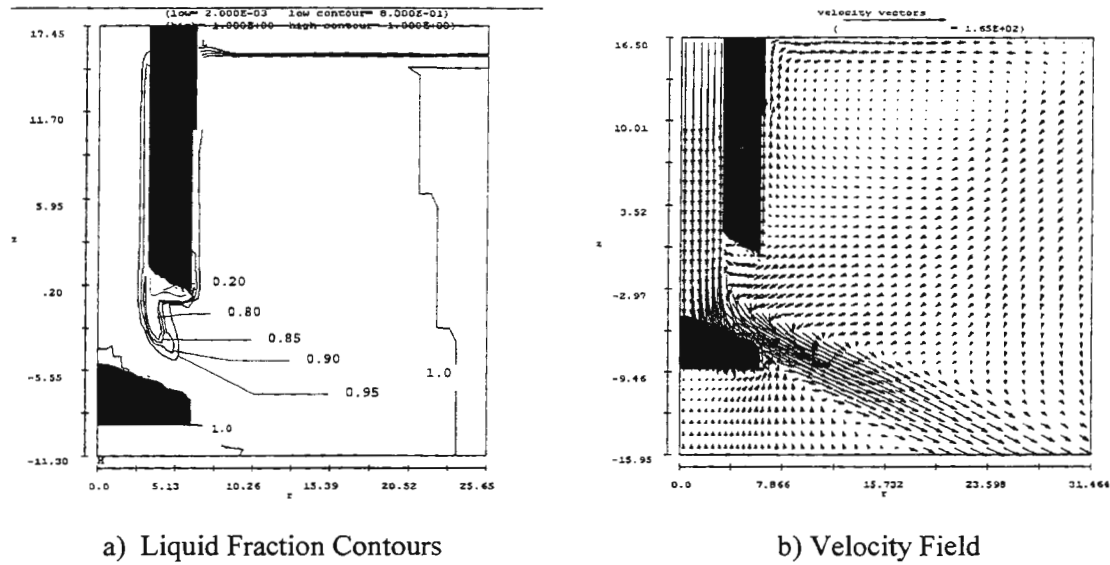
Figure 2.   CASE2: Maximum Argon injection (9L/Min), Maximum Bubble Size (R_0=1.25 mm) and Gas injected circumferentially in SEN

a) Liquid Fraction Contours   b) Velocity Filed
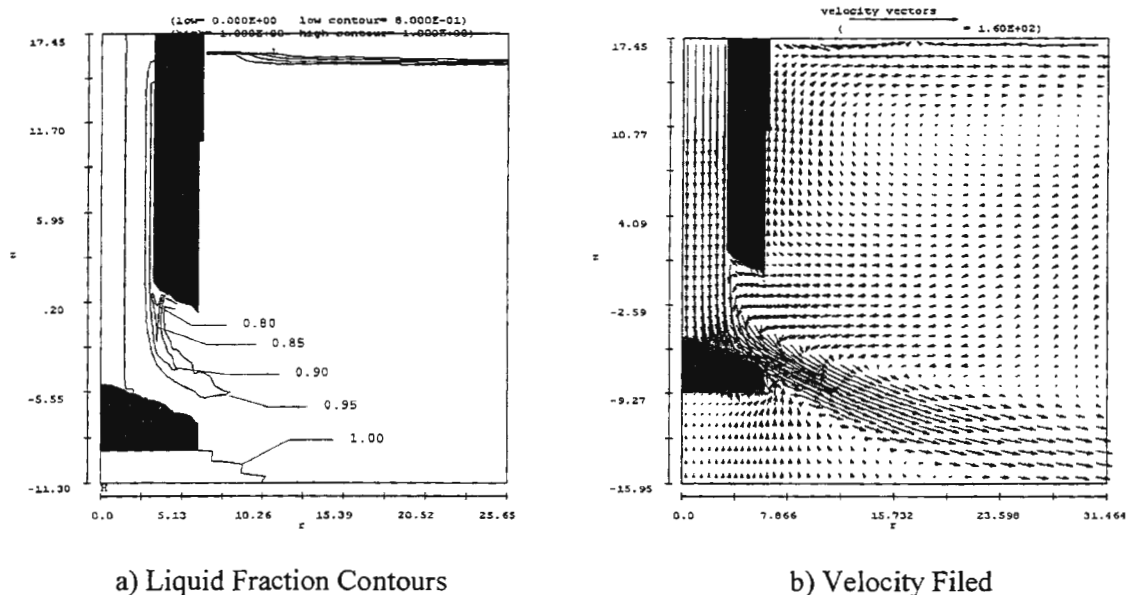
Figure 3. CASE 4: Maximum Argon injection (9L/Min), Small Bubble Size (R_0=0.05cm) and Gas injected circumferentially in SEN

The effect of the bubble injection method can be seen clearly when the gas is distributed uniformly in the liquid at the top of the SEN (Fig 4a). The small bubbles mix thoroughly with the bulk of the liquid steel whilst still in the SEN. The jet stream issuing from the nozzle is consequently less dense than the liquid in the mould. The jet is lifted a short distance from the nozzle and it does not impact the narrow face wall. As a result, the eye of the upper circulation also moves closer to the SEN outlet (Fig. 4b).

a) Liquid Fraction Contours   b) Velocity Filed
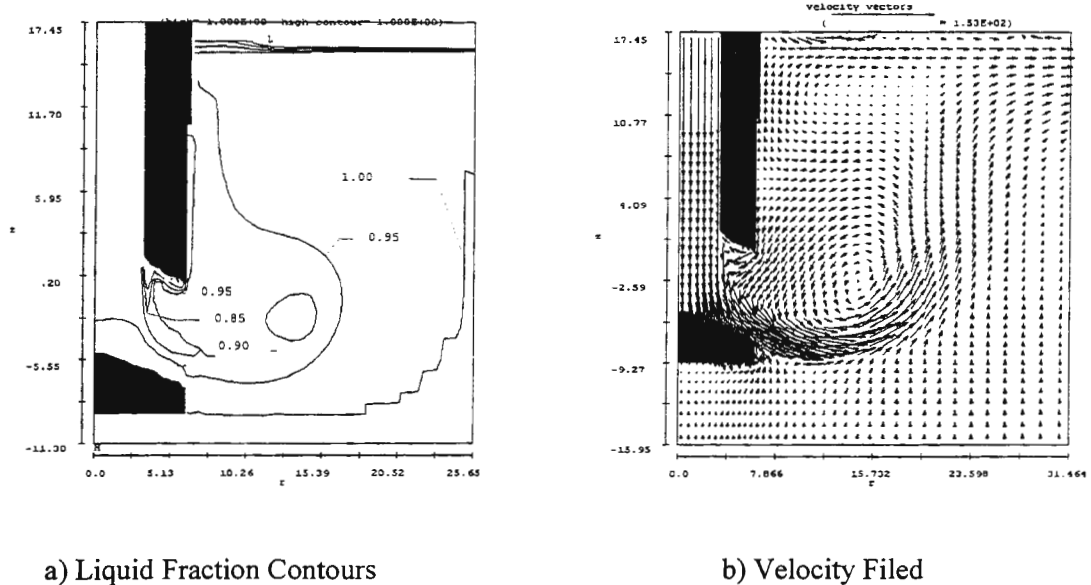
Figure 4   CASE 3: Maximum Argon injection (9L/Min), Small Bubble Size (R_0=0.5 mm)
and Uniform Argon injection in SEN

The effect of the argon injection rate appears to be secondary to that of the bubble size. In Case 5, all the conditions were kept the same as those for Case 2 except that the injection rate is decreased from 9 (Case 2) to 2.25 L/minute. Figure 5, with the smaller injection rate, shows essentially the same accumulation of the gas at the top of the SEN nozzle as that in Fig. 2.. It is unsurprising that Fig. 5b shows fewer bubbles floating out along the outer wall of the SEN.

a) Liquid Fraction Contours   b) Velocity Filed
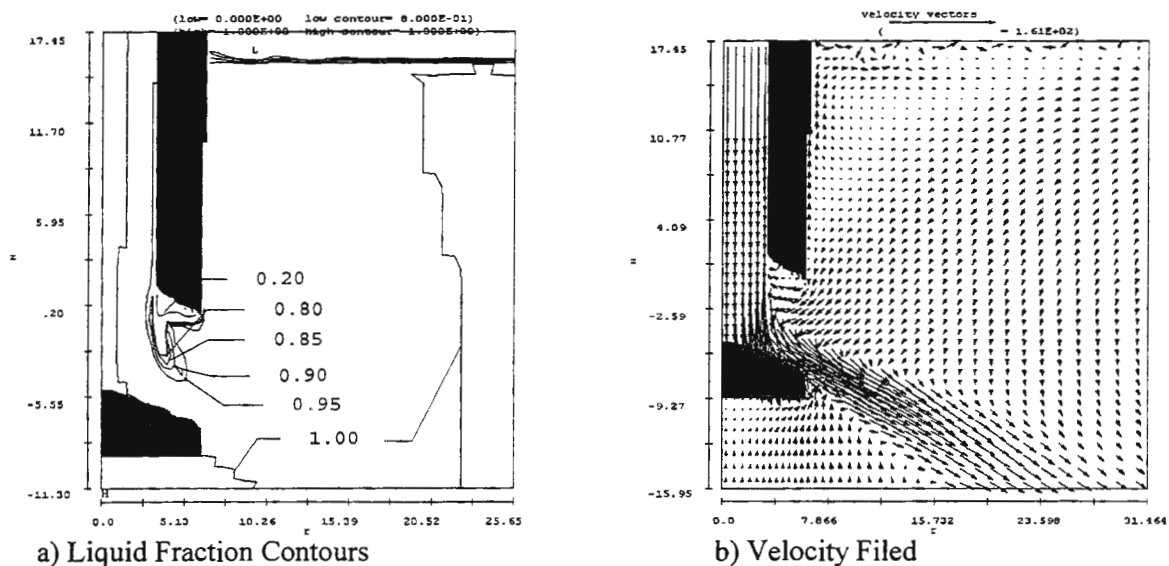
Figure 5 CASE 5: Smaller Argon injection (2.25L/Min), Maximum Bubble Size (R_0=1.25 mm)
and Gas Injected circumferentially in SEN

The effect of various gas injection rate and bubble size is crucial to the understanding of mould flow. One reason for using the gas is to prevent the build-up of alumina inclusions on the inner wall of the SEN and hence causing blockage of the SEN. The most effective method would be to introduce the gas circumferentially when it is already flowing near the surface. Also the methods of gas injection influences both the gas and liquid recirculations. For instance the position at which the gas escapes from the free surface will be affected. This is important to the caster operators and influences the way the various mould level detector systems operate. Modelling allows us to study this effect and with feedback from the plant can give considerable insight into the flow behaviour in the mould.

Isothermal Flow in a Tundish

The flow in the tundish is complex. Two methods were used to visualise the flow pattern, markers and streamlines. The markers were injected in the computer model after steady state flow has been achieved. The marker distribution evolution was recorded and compared with a water model where ink was injected to visualise flow patterns. The comparison shows not discernible difference in terms of the tracer distribution. The streamlines were created based on the simulated flow field by the computer model. Figure 6 shows one of typical streamlines predicted by the computer model, indicating that there are a number of recirculation zones. Other local circulation zones can be seen in the velocity plots in Figs. 7a and 7c.
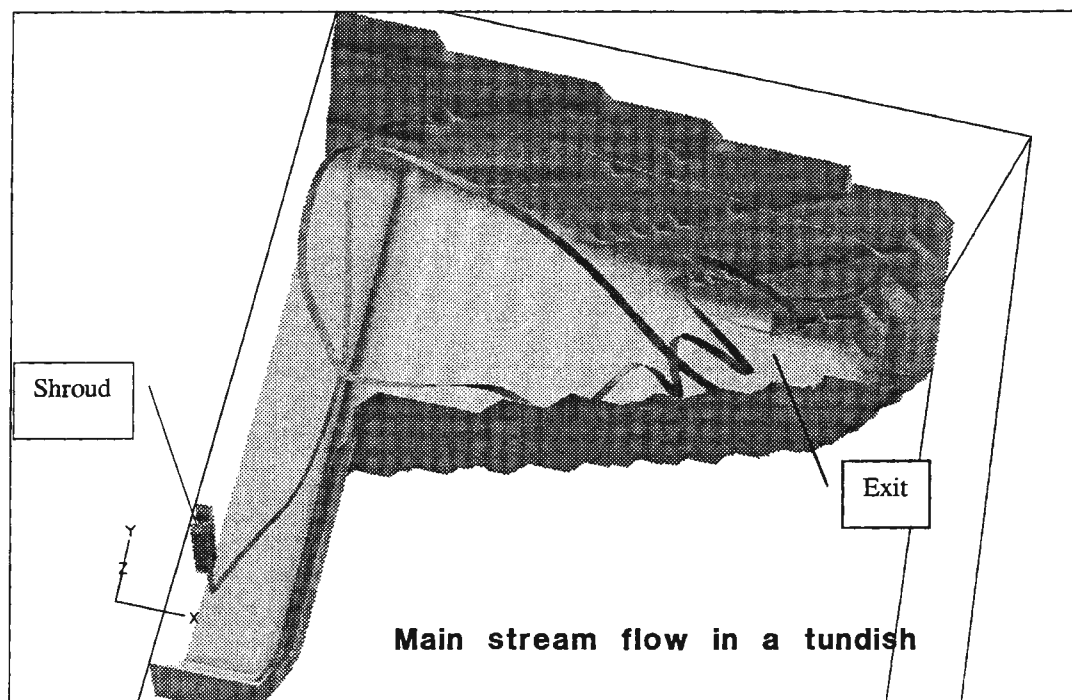
Figure 6. Main Stream in the Tundish Flow.

Figures 7b and 7d illustrate the temperature distribution on the central plane and the transverse plane, respectively. The temperature drop between the liquid steel entering through the shroud and leaving the tundish at its outlet is about 2°C. Figure 7d also shows the location of the liquid steel with the lowest temperature. This information, supported by a streamline analysis, suggests that a portion of fluid passing through this region has a longer residence time than others.

6
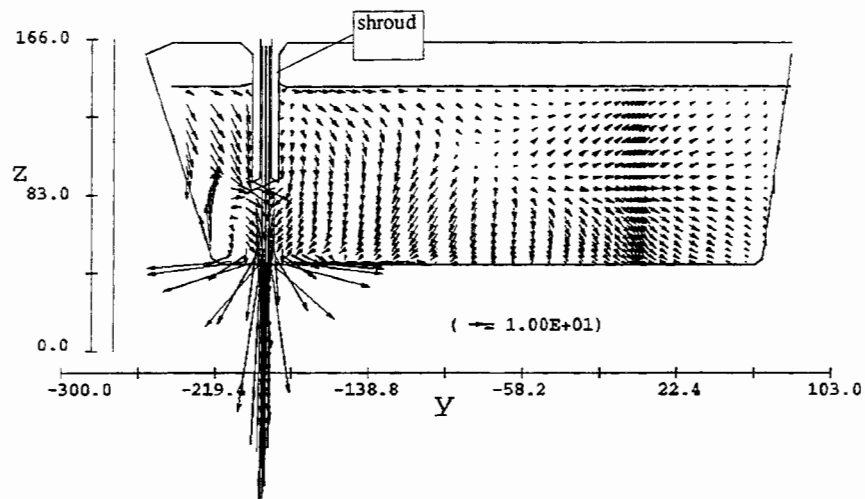
Fig. 7a Velocity Vectors (cm/s) on the Central Vertical Plane

fluid temperature contours

| 1821.21 | 1821.66 | 1822.10 | 1822.55 | 1823.00 |
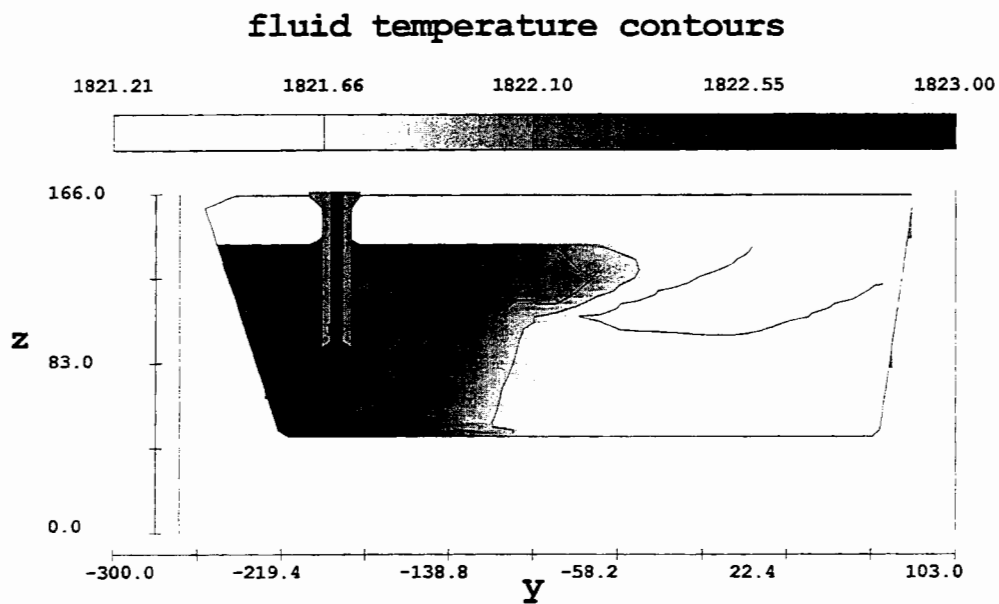
Fig 7b Temperature Contours (K) on the Central Vertical Plane
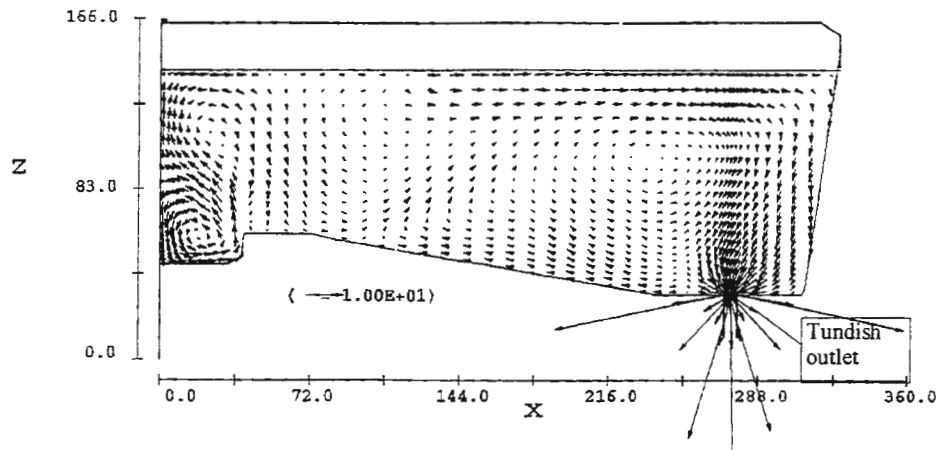
c) Velocity Vectors(cm/s) on the Transverse Vertical Plane (including tundish outlet)
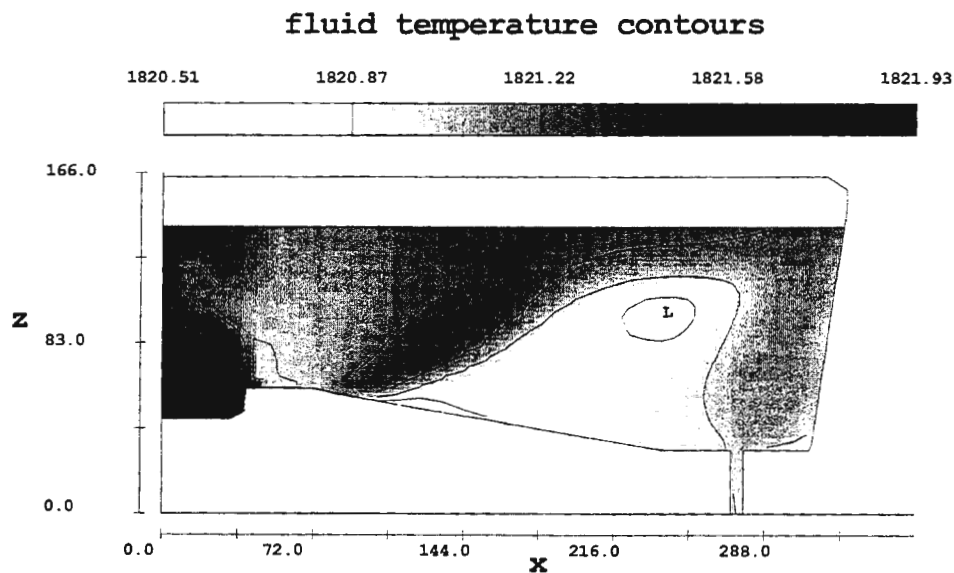
d) Temperature Contours (K) on the Transverse Vertical Plane (including tundish outlet)

Figure 7 Flow and Temperature Distribution in a Tundish

The purpose of the tundish is to distribute the liquid steel evenly to each of the outlets throughout the duration of casting which could be over an hour for billet casting. CFD modelling is now beginning to replace much of the water modelling of liquid steel flows in tundishes. This has been helped by the faster turnround time achievable with modern powerful PCs. The main benefit, however, is the ability to model the heat transfer in tundishes which cannot be carried out easily with water models. The capability of CFD

simulation of heat transfer provides confidence to control the temperature difference between the inlet and the outlet of the tundish. This is crucial when examining the effects of tundish heating. With CFD models we can study the effect of the heater position, the change of heat flux as well as changes in heat input from the ladle.

Discussion and Conclusion

1. Argon injection modelling suggests that argon injection affects significantly the flow pattern. Consequently, it can change the surface turbulence in the liquid steel and heat transfer in the mould. The latter has considerable engineering significance in designing SEN and mould. Currently a systematic investigation of the effects of argon injection and heat transfer on mould flow is being undertaken

2. The predicted flow pattern appears more complex than expected. There are certain regions of recirculating flow which should have been eliminated by the geometrical design of the tundish. The temperature drop from the shroud to the tundish nozzle is acceptable by the design standard. If the tundish geometry is improved to remove these recirculation zones, a more uniform temperature distribution might be achieved.

3. On the whole, the two modelling studies show that CFD modelling can be a powerful analytical tool when it is used interactively between fluid dynamicists and process engineers at the design stage. These studies also indicate that there are a number of areas to be explored, such as how to improve the understanding of the modelling results and make better use of predicted data. As every run of CFD model produces enormous amount of data, efforts should be addressed to extract maximum information from them.

References

1. FLOW-3D User's manual, Flow Science Inc., Los Alamos, 1998.

2. Barkhudarov M, You H, Chin SB, Beech J and Kirkwood DH 'Validation And Development Of FLOW_3D For Casting', Proceedings of the Extraction and Processing Division Congress, TMS Annual Meeting, pp595-604, 1992.

3. J Beech, K Chang, SB Chin, JD Duncombe and AA Moore, 'The Application Of Flow Modelling To The Continuous Casting Of Steel', Advances in Manufacturing Technology XI, pp124-128, 1997.

4. SB Chin, J Beech, J Duncombe and Wang F, 'The Role Of Heat And Fluid Flow Simulation In The Design Of Continuous Casting Plants', First Int. SMESME Conference, Sheffield, 1998

NEW SIMULATION TOOLS:
TOWARDS MASTERING THE EXPLOSION OF INFORMATION

Heikki Hyötyniemi, Ari Nissinen

Helsinki University of Technology, Control Engineering Laboratory
Otakaari 5 A, FIN-02150 Espoo, Finland

Abstract

This paper discusses the role of *simulation* as a way tackle the information explosion in the different branches of engineering work, in research and education. It is proposed that a general purpose simulation environment is needed, and some guidelines for developing this kind of an environment are drawn.

Introduction

In the field of control theory, the number of scientific papers and new methods is getting so huge that not even the domain area experts can master the whole area. This may result in a situation where good approaches can be neglected and forgotten, while the ones that are 'advertised' actively become recognised.

The network publishing is becoming commonplace. In Internet, the information availability is enhanced remarkably. Are the new data mining tools alone enough to help organising the delivered information? The answer seems to be *no* – it is the same 'scientific advertising' that has spread also in the net, and without special purpose methodologies, the information flow cannot be mastered.

In the Internet environment, the old publishing formats, based on textual representations, become more or less obsolete: there are many new possibilities that should be studied instead. In the fields of science where the core is mathematical, the scientific contribution is in the form of mathematical formulae, not as text – at least in principle, this makes it possible *to automatically verify the scientific claims*. This paper presents how the interactive nature and functionality of the new web browsers could be utilised to reach real benefits in our field.

Contrary to pure mathematics, in the technical sciences the theoretical derivation is not the right way to prove things – the right test for validity is *whether the idea works in practice*. It is *simulation* that is the way to prove the validity of an algorithm.

From the point of view of the simulation reliability, the problem that emerges is that just one simulation run with only one set of test data is inadequate: using the modern computing machinery, one can always find a zero-length example so that almost any algorithm works all right. To avoid this problem, the interactive nature of the network tools becomes invaluable. *It must be the reader of the article who can control the simulation process using his own data.*

The goal is 'interactive documents' with a possibility of free experimenting. The main functionality of these documents must be the simulation engine. In addition to scientific publishing, there are many other uses for this kind of general environments based on simulation – for example, new horizons would be opened for control engineering education in Internet.

On the role of higher education in engineering

In the technical universities, *research, education,* and *communication* with the outside world are the main activities. It seems to be difficult to find a common language between the research community and those who apply the results. Tools are needed to visualise the theoretical results, to make them understandable.

It is the research work that is the basis, and from this starting point, it is *simulation tools* that can be used to connect the theoretical work and the practical application of this research work. The users of theoretical results include students, practising engineers, but also other researchers. Simulation, if it is used in a clever way, is a bridge between different branches of engineering work.

About network publishing

The World Wide Web (WWW) and the Internet techniques will affect the engineering work in various ways. One of the most interesting application areas is network publishing, so that research results are easily available to everybody. Electronic form facilitates *hypermedia* documents, where different parts of text can be linked together. The reference mechanisms to other documents becomes simple – assuming that the other documents are also in electronic form. Network agent based search machines can be efficiently used for searching related documents.

It must be recognised that network publishing is far from mature. There are promises, but difficult questions have to be answered first. For example, the distributed storing of the information in various nodes of the net makes it difficult to master it: how can it be assured that a document is not modified after a scientific result is published? Some kind of a commonly recognised authority is needed to record the original contributions, or perhaps some public code ciphering method suffices to ensure the document authenticity.

Usually, it is thought that Internet and the World Wide Web are one of the main reasons for explosion of scattered information. Here, we take a more optimistic view: it is proposed that *Internet techniques and tools can be applied to cope with the wealth of complex, unstructured information.*

Challenges to research

There is a continuum from 'good scientific practice' to dishonesty and fraud. It is easy to neglect the experiments that are 'ambiguous', or do not support the working hypothesis – the results are only slightly streamlined? Yet, the data always has to be selected somehow.

It is typical that reports concentrate on positive aspects. It often seems that no projects would fail ... there always exists a set of simulation experiments that would support the assumptions that are being studied. This means that a kind of 'inflation' is a fact: the claims in research reports cannot be unconditionally believed. The variety of reports is so wide that it is not necessarily the reported results that are weighted – to be convincing, the author list of the paper should include respected names. This reluctance results in unnecessary inertia in scientific progress – novel results may become ignored.

The information explosion and the impossibility of mastering all of the available scientific material is largely due to the incomparability of the research results. The new, rapidly expanding research areas like *soft computing* are good examples of this development: the fuzzy systems, for instance, include so many free parameters that usually reports cannot be critically evaluated – there exist no experts that would have a consistent view over the whole field. This is really a 'fiddler's paradise'.

To become really plausible, the proposed new algorithms have to be tested by an 'absolutely reliable' person – this only impartial judge is, of course, the *reader* himself!

Even if the description of the algorithm were presented in a report, so that in principle the reader could validate the results by experimenting with the algorithm on his own, practically never this is realistic.

Implementing the algorithm would take too much time, and even if it were done, one could not be certain that the implementation would be correct. Often there are some minor factors that are not reported – it is difficult to know whether some details are really meaningful or not.

Rather than showing one fixed experiment in the paper, the author has to *prepare an environment where the reader can easily experiment the algorithm with his own data*. With a mouse click, a new simulation run can be produced.

It is not only the new theoretical work that would benefit from the proposed environment. It is also different kinds of demos and the benchmark problems that could be made much better accessible. How this universal simulation environment should perhaps look like, is discussed later. Before that, the other area of information explosion is studied – it is shown that some of the fundamental problems in education have parallels with the above explained problems in research.

New learning environments

Open and Distance Learning (ODL) [1] is becoming more and more important in the future, and WWW offers a nice medium for efficiently distributing the educational material. In control engineering education, hands-on examples are invaluable, and Internet offers new possibilities for this. Dozens of research projects have already been concentrating on developing material for computer-assisted learning also in our field. However, it seems that the results are not necessarily as good as it was assumed.

In modelling and control engineering understanding the process dynamics is of utmost importance. To achieve understanding of the dynamic behaviour, the basic functionality of an educational tool is measured in terms of its simulation capability. It can be claimed that "one simulation tells more that a thousand words".

It takes a lot of time to develop a functionally complete and visually appealing educational application. Very much time is needed for the user interface, for implementing the mathematical infrastructure, and for other less central tasks. These difficulties mean that often the applications have rather narrow scopes. Different development efforts should then be combined – but, as the projects have been uncoordinated, there are various operating principles and user interfaces, and the combinations of different applications become usually somewhat diverse and nonhomogeneous.

In the field of education, the emergent complexity is equally unintended as it is in the field of research work – the various uncoordinated efforts just do not fit together. Yet, it is not realistic to try and give some regulations on how to prepare the material – individual enthusiasts are, anyway, the real core of any development, and this driving force should not be discouraged. The evolution of the Internet itself, however, shows how local and uncoordinated efforts sum up nicely into an astonishing mosaic. The key point is the *tool* – the HTML protocol, etc., have made it possible that the different home pages together constitute a whole.

There is no reason why the above mentioned universal simulation environment for scientific publication purposes could not be the generally accepted medium for developing educational applications, too. A common platform can hide the irrelevant and annoying implementation-dependent features, and the student can concentrate on the main thing.

Quest for a simulation environment

The key to reliability and comparability between different algorithms, and the key to 'insight-oriented' education, is that the person getting acquainted with any proposed approach must be able to experiment with them using his own data. There exist applications that obey this principle, but, at the moment, the practical, seemingly irrelevant problems often make the installation a nontrivial task.

As an example, take the *Tennessee–Eastman* benchmark problem [2]. This benchmark problem consists of a dynamical model of a complex, unstable chemical process. The model is distributed as FORTRAN source code, and exhaustive documentation can be downloaded as well. In principle, no problems should emerge when installing the model – however, it turned out that the compiled model sometimes behaved strangely. After time consuming testing and consulting, it became clear that these anomalies were probably due to numerical errors – the FORTRAN compiler that was used did not support the same numerical accuracy as was needed.

The new Java-powered browsers facilitate machine independence, but, of course, a specialised theoretician cannot be asked to implement a testing environment from scratch using some general-purpose programming language. There must be tailored tools for realising the testing environment. It can be assumed that something like the MATLAB formalism [3] is *de facto* standard, not too complex for the application developer. The following properties are needed in the environment where simulations are developed:

- MATLAB like formalism with high-level matrix calculation capability,
- object-oriented structure for fluent representation and reuse of complex processes, and
- to achieve universality, automatic compilation to platform independent Java language [4].

This is the application developers perspective – from the point of view of the end-user, one of the main aspects is *transparency*: there should be no need to tackle with the special features of the simulation tool, and the outlook of the user interface has to be standardised to facilitate flexible use of the environment. Using the Java AWT packages, for example, a good-looking, window-based environment with mouse-activated control panels can be defined (examples can be found in [5]).

The goal is that developing the simulation environment would not mean an excessive overhead for the researcher. To reach this, one should be able to apply the *same environment* for experimenting, for reporting the results, and also for preparing the educational material. Only if the generation of the simulation environment is requires minimum effort, if it is almost automatic, generated as a side-effect during normal research work, it can be assumed that such environments will really be prepared.

The kernel of the environment is the simulation engine, and that is why, good mathematical tools have to be integrated in the system. The underlying Java environment has to be boosted with (public-domain) mathematical libraries. The whole tool has to be available as public-domain software to make it accessible to everybody[1].

Example application

Kohonen algorithm [6] is a well-known method for reaching self-organisation of data. The high-dimensional input vectors are projected onto a lower-dimensional manifold, so that the mutually related data samples are located near each other in the map. This gives a nice visual view of the data properties. However, the algorithm is based on Euclidean metric (or some other vector norm). It is not possible to map structurally differing entities using this algorithm.

---

[1] When proposing a new theoretical method, the running example, but also the source code, has to be made available to make it possible to verify that no errors are done, neither in the implementation of the algorithm, nor in the analysis of the results.
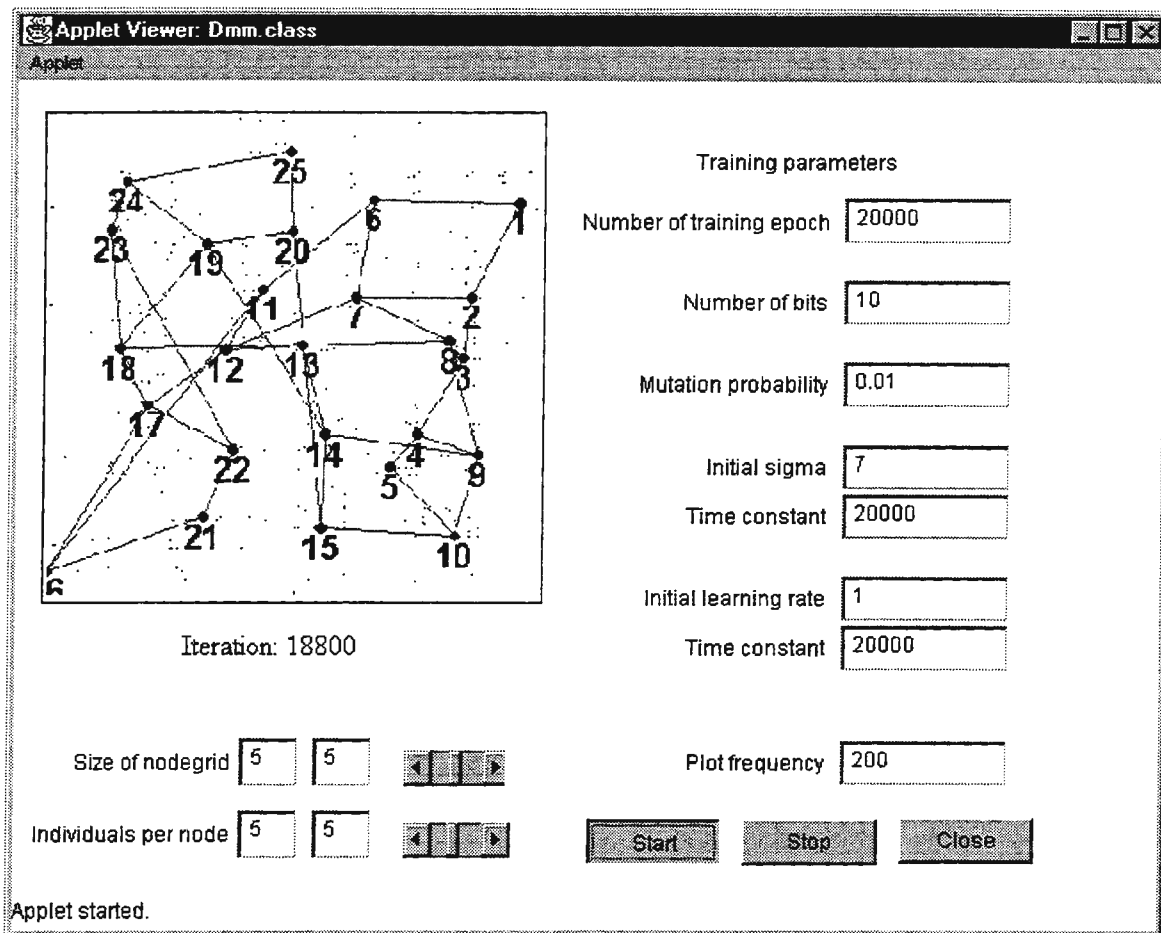
Figure 1. The user-interface of the GASOM simulation Applet in Internet. The user can experiment with different parameter values, gaining some intuition in the operation of the algorithm (the meaning of the parameters and how to modify them, is explained in the demo environment)

Genetic algorithms, on the other hand, are capable of optimising structurally complex representations. The idea is to optimise the structures using 'smart' random search, so that no gradient information is needed. There is a drawback, though – using genetic algorithms, only one optimised structure is returned. Generally, various different models are needed to represent complex measurement data, and, in the Kohonen sense, one would like to have a 'map of structures' where the models having similar behaviour are located near each other.

The example that will be elaborated on here is an algorithm called GASOM. The basic idea of the algorithm is to combine the advantages of genetic algorithms (GA) and self-organising maps (SOM) together. The details of the algorithm are not concentrated on – a more concrete discussion and a practical application can be found in [7]. The main point here is to 'prove' that the basic idea is sound: using the above approaches, a simulation environment is constructed so that a person who is interested in the approach can experiment with it[2]. Note that in this case the coding is done directly in raw Java, and in the tailored environment, more

---

[2] This demonstration is available in Internet and can be run using any Java enabled browser. It is located at http://SAATO014.hut.fi/Hyotyniemi/publications/98_eurosim/demo.htm.
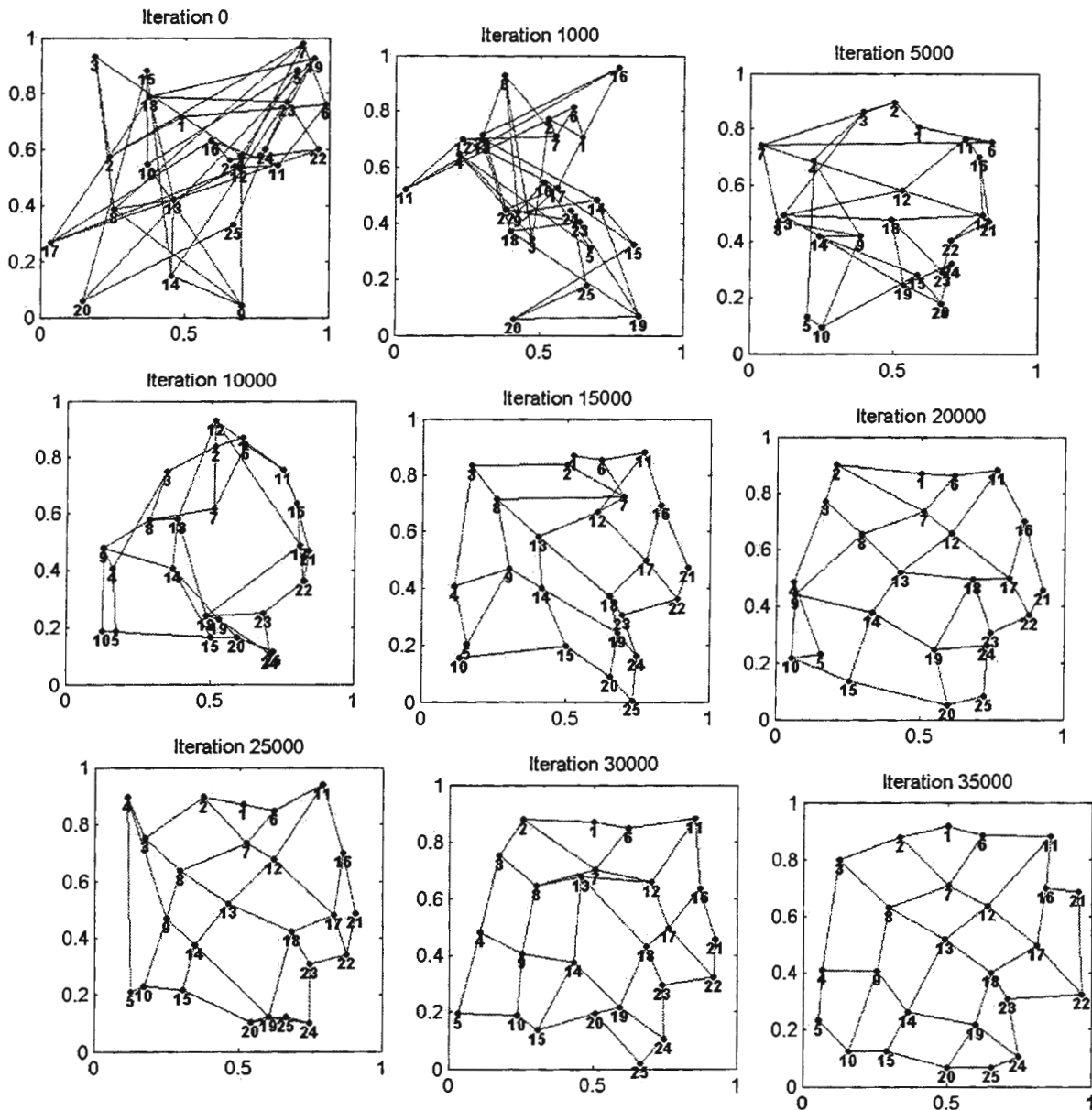
Figure 2. The evolution of the GASOM network when 'simple' input data is used. The results are similar to those when using the Kohonen self-organising map. This behaviour reveals that *at least* in special cases self-organisation of structures takes place

tailor-made systems definition tools are needed. Now, unfortunately, this kind of an environment does not yet exist.

The convergence of the self-organising map algorithms cannot be rigorously proven. What is more, the asymptotic properties of genetic algorithms are extremely complicated. Combining this kind of mathematically tedious problems together makes it very difficult to derive any analytical results. Yet, choosing appropriate parameter values, the algorithm 'almost always' finds a reasonable organisation. The simulation environment helps one to gain insight: the operation of the adaptation algorithm is fundamentally non-continuous, and the adaptation process looks very different compared to the Kohonen algorithm (the modifications are seemingly spurious) – however, what is important is that the final result looks nicely organised.

In Fig. 1, the user interface of the simulation environment is shown. There are various adjustable parameters that the user can experiment with. The 'structure map' is shown in the upper left corner of the window. In Fig. 2, a typical simulation run is presented – the originally random configuration gradually unfolds. It needs to be noted that the data that is now modelled is rather trivial (two-dimensional input samples are uniformly distributed on the unit square), and the basic self-organisation algorithm could be applied. Much more complex input data could be modelled, but the goal now is to use so simple data so that the results can be easily verified.

The approach and the algorithm are delivered on the 'as is' basis – it is up to the potential user whether he thinks it could be useful in his own application.

Discussion: practical simulations vs. theoretical proofs

In mathematics, one can study whatever one likes, the real world is not a restriction. Things are studied deeper and deeper with no interest on the question whether there are any applications for the theories in the first place. Sometimes, the theoretical constructions can be utilised later – more often this is not the case. Without any connection to real life, the new concepts have no meaning and no relevance. The relevant, fruitful areas are seldom the ones with analytical solutions. In mathematical research, the final deadlock is encountered when some hypothesis cannot be firmly established.

Today, many of the promising fields of research defy elegant formulations – these methodologies seem to solve some problems that have no explicit solutions. This kind of fields are, for example, neural networks and genetic algorithms. Of course, one *can* find mathematical formulations for the operations also in these fields – what is important is that this formalisation is usually done only afterwards, so that mathematics does no more have its prominent role as a tool for predicting the behaviour and guiding the research work. Today, the new methods are always first tried in practice, and only if the experiments are successful, theoretical explanations are searched for. The experimenting is done using simulation.

Technical sciences have usually been regarded as somehow inferior: technical research work means combination of applied mathematics, physics, economics, etc. Of course, during the phase of the actual product development, financial considerations become more and more central, but before this final (and the most visible) level, the natural dynamics of the scientific research takes place. This dynamics is dictated almost solely on the pure 'engineering aesthetics': what works, what is useful, and what is interesting.

> As an example, take the *identification theory*: there is a wealth of mathematically oriented recursive algorithms that can be shown to converge to the right parameter values. However, as everybody who has tried the algorithms in practice knows, this convergence is often extremely slow and the theoretically correct algorithms may turn out to be unacceptable. A 'simulation proof' is more convincing than the mathematical one! The strange thing is that this situation is not widely recognised: the appliers of the supplied algorithms are on their own, having to find tricks to cope with the practical problems, theory seems to live its own life ...

Technical sciences can be regarded as a paradigm of its own, separate from mathematics, having its own research methodologies and objectives. On the most fundamental level, even the *epistemological* principles differ: what is *information* and what are the methods for its acquisition. It is not the axioms and the deduction rules that are the key to knowledge. In this case it is *simulation* that really offers new visions. Interesting, unanticipated results can be obtained when complex calculations result in new structures under massive iteration. When starting from raw data, the algorithms may converge autonomously to some nontrivial configuration, so that some kind of *self-organisation* emerges.

There is a clear difference also between natural sciences and technical research work: it is not only the observed behaviour of some existing thing, it is also artificial 'software creatures' that are studied – perhaps these virtual creatures that are alive only in the simulations fit better in some special environment and can outperform the natural ones in some sense?

Summarising the above discussion, it can be argued that in technical research work a mathematical proof alone is, if not invalid, it is insufficient. The example above from the field of identification theory demonstrates this: only a set of simulations using real data can be a convincing proof for an identification algorithm. And as the applications become more and more complex, simulation is becoming the primal method for proving things in technical sciences. This central role of simulation should be recognised, and the challenges should be faced.

The proposed simulation environment has not yet been implemented. Now it is time to refine the goals: to make the environment useful in practical tasks, what kind of properties are needed and how the realisation should be carried out?

Acknowledgement

Reference links

Because of the topic of the paper, and because the papers will be available also in the electronic form, the references are in this case given as a list of HTML links.

1. The view of Open and Distance Learning (ODL) as the European Commission sees it is available at

   `http://europa.eu.int/en/comm/dg22/socrates/open.html`

2. The Tennessee–Eastman archive is available to anybody wishing to experiment with the benchmark problem:

   `http://www.che.ufl.edu/prototype/tenneast/tearch.html`

3. Information on MATLAB can be found in

   `http://www.mathworks.com/`

4. The most recent status of the Java is made available in the net by the developer of the language, Sun Microsystems:

   `http://www.javasoft.com/`

5. Examples of Java Applets are available in

   `http://www.developer.com/directories/pages/dir.java.html`

6. Up-to-date information about research on self-organising maps and the Kohonen algorithm can be found, for example, in

   `http://www.cis.hut.fi/wsom97/`

7. A practical application of the above mentioned GASOM algorithm has been presented by Heikki Hyötyniemi, Ari S. Nissinen, and Heikki N. Koivo: "Evolution Based Self-Organization of Structures in Linear Time-Series Modeling" can be found in the *Proceedings of the Third Nordic Workshop on Genetic Algorithms and their Applications* (ed. Jarmo Alander), Helsinki, Finland, Chapter 11, pp. 135-152. It can also be found in Internet at

   `http://SAATO014.hut.fi/Hyotyniemi/publications/97_nwga3.htm`

Application of Web Browser and Software Component Technologies in Operator User Interfaces in Process Simulation: A Case Study on Dynamic Simulation of Rotary Lime Kiln

Tommi. A. Karhela[1], Seppo M. Kuikka[1], Matti. A. Paljakka[1]

[1]Technical Research Centre of Finland, P.O.Box, 1301,FIN-02044 VTT, karhela@sah.vtt.fi

ABSTRACT

The use of the web opens a large variety of possibilities in dynamic process simulation. Simulators can be rather expensive to buy for a casual user and also heavy to run in an ordinary PC. Often the user wants to test the simulator in order to be convinced about its capabilities. In this paper a solution to these problems is discussed and architectural approaches for implementation are introduced. A lime kiln user interface and the communication architecture behind it is shown as an example.

There are a number of ways to design a client-server on-line web simulation environment. Some of the architectural options are discussed in this paper. The user interface can be implemented e.g. with Java applets or with ActiveX controls. The applet and the simulator may communicate directly by using low level communication (TCP/IP) or by following some application level standard e.g. CORBA. In this paper two particular architectures are discussed.

As a case study, a web user interface for the lime kiln simulation model is introduced. The lime kiln model is implemented as a part of APMS (Advanced Paper and Pulp Mill Simulator) and it uses mechanistic modelling to capture the dynamic behaviour of the calcining process in a kiln. This example also gives an overview to the reader on how to implement these kinds of controls in the web environment.

INTRODUCTION

Nowadays a web browser can be found on nearly every workstation. When connected to the server, several browsers may use a common simulation model as clients. Depending on the implementation, all users can have own copies of the system running at the server or they may use the same model. The users can perform changes to the process through a web UI and observe the simulation results on-line through the trend windows and monitors offered by the web applet.

The server may be located anywhere in the www and the maintenance can still be centralized at one place. This is an easy way to advertise a simulator and also to sell simulation time for the customers all over the world.

In the first Chapter of this paper the benefits are discussed in more detail. The following Chapter concentrates on the utilization of the browser environment as a user interface and also points out some security issues related to this usage. In the remaining Chapters a user interface of the lime kiln is shown and the underlying structure of the data transmission between the applet and the simulator is introduced.

BENEFITS OF WEB TECHNOLOGY IN PROCESS SIMULATION DOMAIN

In web technology, there are several significant benefits compared to other user interface technologies:

- Web user interfaces are independent of the operating system on the client host.

- Web technology provides an easy way for a simulator vendor or a simulation model developer to give access to demonstration models.

- The web visitor may be charged for the simulation time, which makes it easier for a simulation model developer to serve also occasional or temporary users.

Basically, process simulators may have two kinds of web user interfaces:

1. An interface for simulating a model may have fixed modelling graphics with a few buttons, hot spots or dialogs for viewing and editing a set of model parameters. The simulation results may be shown by monitor boxes in text format or graphically by trend windows. Also the use of changing colours and even sound effects is possible.

2. A user interface needed for demonstrating both the modelling and simulation features of a simulation tool is required to have facilities to change the structure of the model. By the aid of the user interface, the user should be able to browse model libraries and instantiate model units. He should also get a good view on the model structure and an easy access to its state, i.e. the properties of the model units

There are several advantages in using distributed software component technologies in the web environment:

- **Reuse of existing components**: Software components can be reused easily in a web application. Libraries and other components can be wrapped using object interface tools designed for the web environment.

- **Fast deployment:** Applications can be built and deployed quickly because existing technology is being wrapped using Web technology. The deployment can be done simultaneously into different platforms, because the web application is installed only on the server machine. After installation the application can be used via a web browser by an authorized user.

- **Maintainability**: Intranet applications are all installed, upgraded, and corrected in one place - a Web application server. It is not necessary to install new versions of applications or correct existing versions of applications on each PC because all application software is on the server.

- **Security**: Since all databases and applications are installed on Web servers, it is possible to build layers of security protection around sensitive information.

UTILIZATION OF WEB BROWSER ENVIRONMENT AS AN USER INTERFACE

User Interface from the User Point of View

The design of user interfaces is always a challenge. The users have different opinions about the lay out and ergonomics of the interface. This can be taken into account when developing a UI for some tool application. In this case the target is to build as flexible an interface as possible. This way the user can customize the interface according to his taste and demands.

In a control room environment the operator often needs services from several automation and information systems to carry out all of the tasks. The web browser environment could provide us a long waited plant-wide "single window" solution. Using a browser may, however raise new problems to the user. How to distinguish between less important, not so risky tasks and for example potentially hazardous process control actions. This may be a problem specifically when the user interface of the simulated and real system is similar.

User Interface from the Builder's Point of View

With high level tools, it is quite easy to build user interfaces to the web browser environment. Visual Java and C++ tools are getting better all the time and the component galleries contain an increasing amount of ready made Java beans and ActiveX controls. Also other visual development platforms have more options to deploy application and user interfaces straight to the web.

Interface definition language compilers, on the other hand, are used to generate low level inter process and inter machine communication code (to so called proxys and stubs). They thus relieve the user interface builder from the burden of coding low level communication primitives between the user interface applets or ActiveX controls and the server.

Specific Needs of Process Simulators

In building a training simulator interface for process industry the approach is influenced by the fact that a real interface already exists. The interface has been designed by the supplier of the automation system and the simulator interface needs to be like the real interface. Of course the best case is to use exactly the same user interface in other words the automation system UI. In a web application this is normally not yet possible. So the interface is tried to be built as close the real one as possible.

The interface may be framed and support the co-use of several editors.

- For an overview of the model structure, a graphical net editor is needed. Hence, the interface must support some web graphics format, and the client needs to have an appropriate plug-in software to use the model. In the graphical editor, questionnaire forms or dialogs for viewing and editing component properties is needed.

- For listing components by type, tree-like browsers and tables with component properties are good tools.

Security Issues

On a plant-wide Intranet, a firewall computer separates the Intranet from external Extranets and Internet. The use of web browsers requires thus little or no modification in simulation software as to the security issues involved. The level of security achieved is the same as the one with a normal desktop environment. It depends on the username and password protection schemes used, possibly complemented with additional, proprietary system level access restrictions.

The situation is quite different, however, when Extranets or Internet are used due to potential risks of intruders. By adding an Internet encryption package both to the server and the clients' web pages, simulation data can, however, be securely accessed from anywhere on the net. This procedure makes it necessary for the user to have (and keep secret!) an access code for the simulation system. Also other methods, for example RSA (developed by Rivest, Shamir and Adleman) with public and private keys may be used to encrypt messages and authenticate users of the system.

Also the way in which software components are implemented within the web pages has an effect on security. The current Java applet security policy is the same for all applets, and states that an applet may open a network connection only to a host where it came from, and all file system access is blocked. This policy works for some applets, but is unnecessarily restrictive for Intranet applications, as well as being too permissive in some kinds of confidential environments. The activeX security policy, on the other hand, trusts in effect on the reliability and skill of the signed vendors of the ActiveX controls used by the application. The ActiveX controls, which are loaded from the server to the client are normally permitted to utilize all the client host machine resources according to COM authentication and access control procedures.

CASE STUDY ON DYNAMIC SIMULATION OF LIME KILN

About the Calcining Process in the Kiln

Lime Kiln process is one of the unit operations in the recovery cycle in a pulp mill. The chemical recovery has three main purposes. First of all, fresh cooking liquor is formed from the spent one. Secondly, energy is produced from the organic waste, and thirdly, unnecessary air and water polution is avoided. The recovery process is actually a sequence of sub-processes. Calcining is the sub-process which takes place in the lime kiln. A schematic process diagram of the kiln is shown in Figure 1.
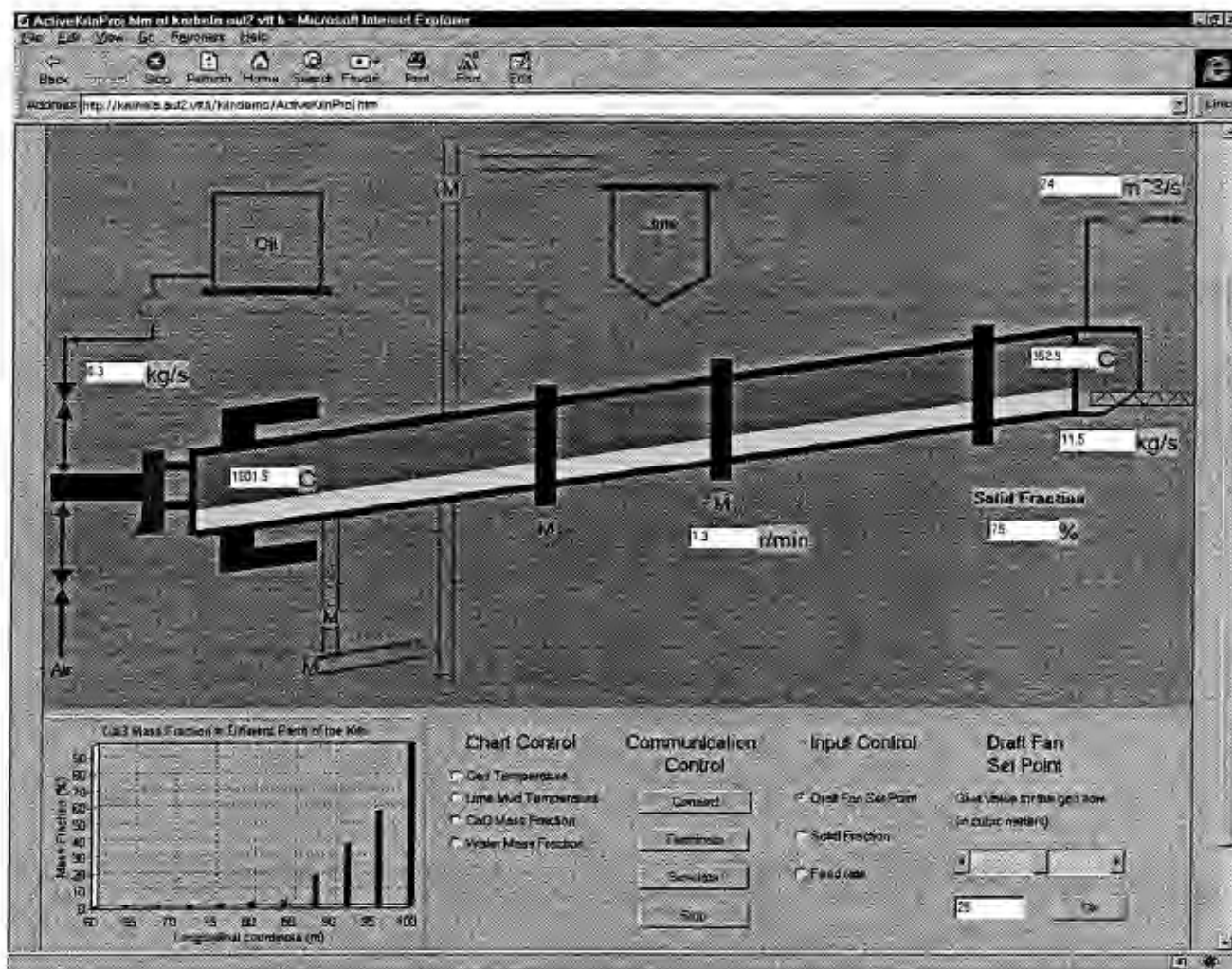
Figure 1.Lime kiln user interface

The main purpose of calcining is to convert calcium carbonate ($CaCO_3$) produced in the causticizing process back into lime (CaO) for reuse in causticizing. This is usually done in three different phases. First the lime mud ($CaCO_3$) is dryed, then the temperature is raised up to at least 800 °C where the calcination reaction starts, and after that the temperature has to be maintained in order to keep this endothermic reaction going. In this reaction lime (CaO) and carbondioxide (CO_2) are produced. This calcining process takes place in a lime kiln which is a cylindrical rotating oven with a length between 40 and 120 m and a diameter between 2 and 4 m.

Wet lime mud is fed into the high end of the kiln with a solid content of 60-70 % [3]. The solid phase moves downhill inside the kiln and hot gases flow in the opposite direction. The kiln also rotates constantly taking care of the transportation and mixing of the solids. A temperature between 1000 and 1100 °C is usually maintained at the hot end of the kiln. Without any cooling system at the end the lime would be released having a temperature of about 900 °C. The burner is located at the discharge end of the kiln. There either natural gas or oil can be used as fuel. Usually very large kilns with a length-to-diameter ratio over 30 are used in order to get the best time/temperature balance. The passing through time depends on the slope and the rotational speed of the kiln. Usually the slope is between 2 and 4 cm/m, which means 1° to 2.5°, and the rotation speed varies between 0.5 and 1.5 rpm. With these values the passing through time is around 2-3 hours [2].

Simulation Model

In a modeling sense a kiln can be divided into three different parts: the gas phase, the solid phase and the kiln wall, which can all be treated separately. However, the mutual interactions between all these parts have

to be taken into account. The gas phase sets some demands on the model. Mass flow, pressure, temperature and mass fraction of different components in flue gases have to be solvable in any parts of the kiln. In the solid phase mass flow, temperature and mass fractions are the interesting variables in different parts of the kiln. In addition the temperature in the kiln wall is needed for the energy balances.

The simulation model is built using mechanistic laws of physics and chemistry. The conservation equations for mass, momentum and energy are used. The model is validated using data from the Metsä-Sellu Äänekoski Mill. More information about the model can be found in [1].

First Approach Using Low Level TCP/IP Library

The model described above is implemented into APMS (Advanced Pulp and Paper Mill Simulator) environment. This environment offers an API connection for data and command clients to connect to the platform run time. The first approach uses this TCP/IP based library for the data transmission between client machine and simulator. This also means that the software component which uses this library has to be first downloaded to the client machine. Figure 2 illustrates the architectural structure of this solution.
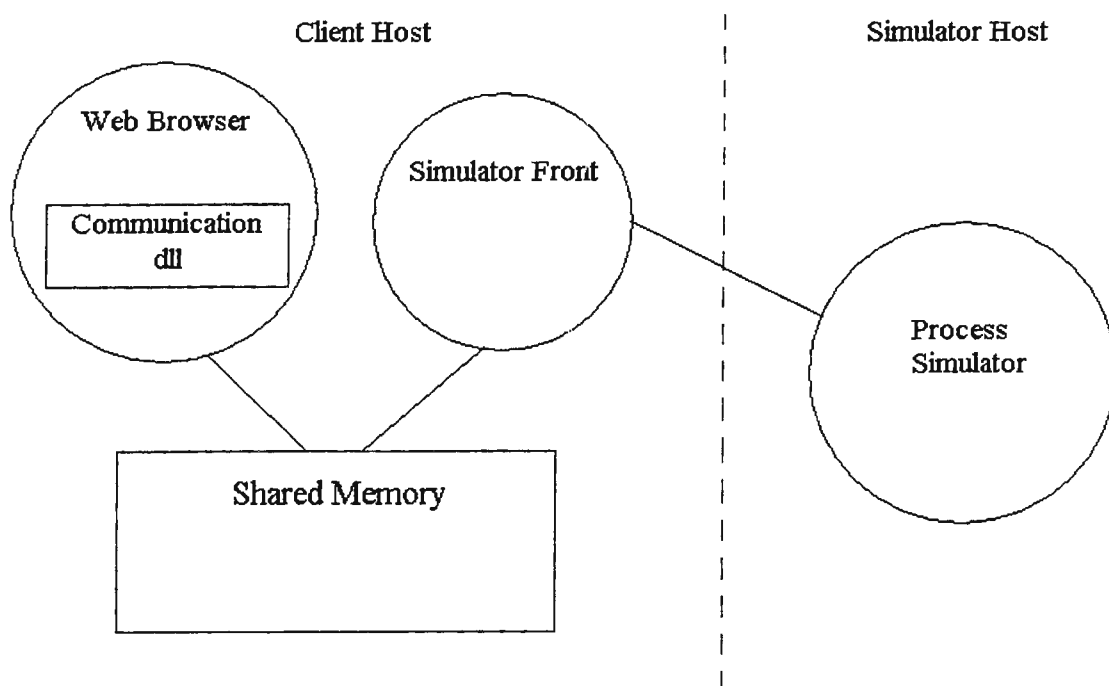
Figure 2: Architecture of approach 1

It can be seen that actually two components have to be downloaded to the client machine. Firstly the communication dll which handles the data transmission between browser and shared memory. Secondly the simulator front which handles data transmission between shared memory and simulator. Front executable uses the TCP/IP library mentioned above.

This architecture was chosen because it allows several data clients to connect to the same simulator using only one net connection between the front and the simulator. This is not a main requirement for web browser use, but there might be some other application or user interfaces which may exploit this feature. For the web use this choice is not so convenient because of the possible security problems. User has to accept to download two software components into his own hard disc. However, in the Intranet use or for some special needs where several clients needs simulated data at the same time this solution works well.

The applet in the first approach is built using Borland Delphi 3.0 Active Form component. The implementation language in Delphi is Object Pascal and the applet is in the form of ActiveX control. The communication dll is implemented using C++, but the dll interface to the Object Pascal was given in C form. The simulator front and shared memory were implemented using C++. When the user enters the web page

the system starts to download ActiveX control and other components. When the user connects to the simulator the front program is started and the shared memory is created. Data exchange starts at the start of simulation.

Second Approach Using Application Level Object Interface

The second approach, described below is better than the first from the point of view of the web usage. The structure of the approach is given in Figure 3.
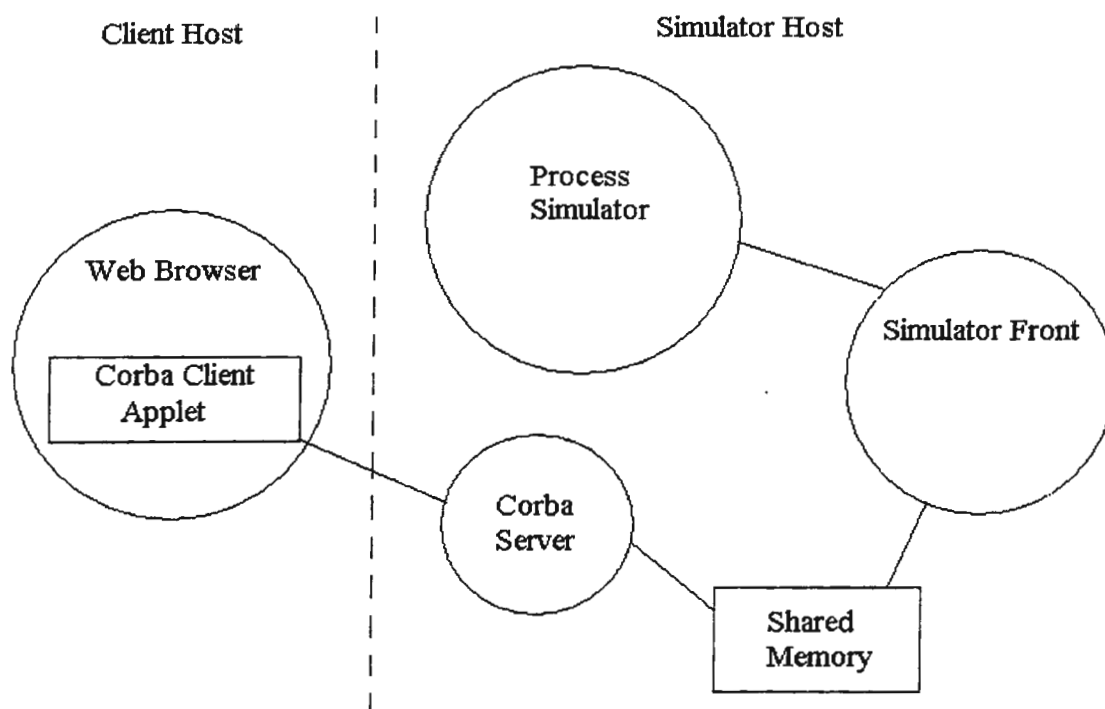
Figure 3.Architecture of the second approach

It can be seen from the Figure 3, that only the applet is downloaded to the user's machine. This doesn't cause the same kind of a security problem as the previous approach. The applet is implemented using Java. Orbix Web is used for the Corba connection and the Corba server is actually the same communication dll component from the first approach, now wrapped as an Orbix Corba server. The rest of the data transmission chain is the same as before. The second approach has the advantage that the user has only a light channel to the simulator. However, model developers may need also other channels to the model. For example they may want to analyse simulated data in some other application. If that is the case, the first approach may offer more possibilities.

Another way to build this kind of an application level object interface between the client and the server is to use DCOM (Distributed Common Object Model) provided by Microsoft as a part of the NT operating system. This approach would look the same as the Corba approach, only this time the object interfaces would be built according to the DCOM specification. However, this would in effect fix the operating system to NT (although there nowadays exist also Unix implementations of DCOM). The use of Corba enabled Java classes is not dependent on the operating system platform.

User Interface

The activeX control user interface is shown in Figure 1. This user interface was designed so that there is only one window where the control is located. On the top of it there is a schematic figure of the process. Some process related data is showed in the picture and it is updated online. This is a similar way as the the one in which the data is represented in the automation system. The control and trend tools are located at the lower part of the applet. In the middle the user has a communication control from where the user may

116

connect to the simulator, terminate the communication, start simulation or stop simulation. These functions are available only when the simulator is first connected to the applet.

In the second approach the connecting operation is not needed. It can be seen from the Figure (3) that the connection to the simulator has already been established at the server host. The trend tools are located in the left. Gas and mud temperatures versus time are presented from the hot and cold ends of the kiln. User may zoom in and out in these figures. Also lime and water mass fraction profiles are presented as pillar diagrams. These charts are also updated online. The update frequency was chosen to be 1s. On the applet side this was implemented using timer component from Delphi. At the simulator front the same update frequency to the shared memory was used.

The input control is located at the lower right corner. Using a scroll bar the user may change the draft fan set point, the lime mud solid fraction at the feed and the lime mud feed rate into the kiln. Changes in the draft fan speed affect on the gas flow inside the kiln and in this way also to the temperatures and chemical reactions inside the kiln. The moisture in the feed as well as the feed rate affects on the temperature profile in the mud and in this way also on the reactions.

The java applet looks very much the same as the ActiveX control. The main difference is that separate window dialogs can be used more easily with Java. On the other hand it was not so easy to find a good enough trend tool in the Java environment. Consequently, when writing this article, the trend windows are not yet functioning in the Java applet.

Concluding Remarks

The use of web browsers, found almost everywhere, gives new possibilities in dynamic process simulation. When connected to the simulation server, several browsers - acting as clients - may use a common simulation model. The server may be located anywhere in the www and the development and maintenance can still be centralized at one place. Using the web it is also easy to advertise a simulator and sell simulation time to customers all over the world. A browser is also functionally a viable alternative as a simulation user interface both for the end user and the model developer alike. However, precautions have to be considered when running both the simulator and the real process via the same interface as well as when using the simulator outside the plant wide Intranet in Extranets and global Internet.

Dynamic simulation model of rotary lime kiln have been used as an example. This sub process in pulp mills involves many different phenomena, which were modeled using a mechanistic approach. The results were validated using data from the Metsä-Sellu Äänekoski Mill.

Browser based user interfaces may be designed and implemented in various ways. In this paper two approaches have been described. The first one is based on a functionally rich client using shared memory and communicating with the simulator using a low level protocol. The other one is based on a thin client communicating with the server using an application level protocol. The solutions are both usable for end users and developers, the first one being more developer oriented, the second one more www-end user oriented.

References

1. T. Karhela, J. Lappalainen, H. Peltola and K. Juslin. Dynamic Simulation Model of Rotary Lime Kiln. To be published in 1998 TAPPI Recovery Conference Proceedings (accepted)

2. D.J. Kramm. Update on Lime Sludge Kilns in the Pulp Mill Environment. Paper Trade Journal, 163(9), May 15 1979.

3. G.A. Smook. Handbook for Pulp and Paper Technologists. Angus Wilde Publications, 1990.

Education in Ecology at the Internet with an Object-Oriented Simulation Language

Manuel Alfonseca, Rosa Carro, Juan de Lara, Estrella Pulido

Dept. Ingeniería Informática, Universidad Autónoma de Madrid
{Manuel Alfonseca, Rosa Carro, Juan de Lara, Estrella Pulido} @ii.uam.es

Abstract

This paper describes the procedure we have used to generate semiautomatically a course on Ecology for the high-school level, using an extension of the Volterra equations to describe the interaction between different species levels. The simulations used in the course have been written in our own special-purpose object-oriented continuous simulation language (OOCSMP). Our compiler for this language automatically generates Java code and html pages, which must be completed manually with the associated text and icons.

Introduction

System simulation [1] is one of the standard tools of multimedia systems and computer education. In continuous simulation, the appropriate mathematical tool is the set of algebraic-differential equations. Continuous simulation models are usually programmed in a special purpose language, or in general purpose code. Continuous simulation languages use different input interfaces, such as block diagrams [2], bond graphs [3], systems dynamics graphs [4], or mathematical equations [5].

Object-oriented programming originated in the sixties in a discrete simulation language, SIMULA67 [6], which incorporated many of the ideas later included by Alan Kay in the first general purpose object-oriented language, Smalltalk [7], and by Bjarne Stroustrup in C++ [8]. In recent years, object orientation has been added to continuous simulation languages [9]. The language we are using to generate our educational applications (OOCSMP) is our own extension to the old CSMP language (Continuous System Modelling Program), sponsored by IBM [10].

In a previous paper [11], we have described some of the new features of OOCSMP that make it very useful for the simulation of systems consisting of many similar interactuating parts, as in the ecological example we are showing here in detail.

The paper is structured as follows: the first section presents the equations used to model ecosystems; the second describes how the model is programmed in OOCSMP; the process of generation of the Internet course is explained next; the last section describes a specific course on Ecology; finally, the conclusions and our future work are presented.

Simulation of a complex ecosystem

We have made use of an extension of the well-known Volterra equations [12] to represent the dynamics of a complex ecological system. The original equations apply to a two-level two-species ecosystem, where a single predator population interacts with a single prey population in a simple way. The equations are:

$$X' = -m.X + n.X.Y$$
$$Y' = p.Y - q.X.Y$$

where X is the predator population; Y the prey population; and the four constants, m, n, p, q, are all positive. The prey is assumed to feed from an inexhaustible source, therefore, in the absence of the predator (X=0), its number will increase at the rate indicated by constant p. In the presence of the predator, its number will diminish in a proportion to the number of encounters (constant q). In the absence of the prey, the predator

will dwindle at the rate given by constant m. In its presence, its number will increase in proportion to the number of encounters (constant n). The system is in equilibrium if the following holds:

$$X0 = p/q$$
$$Y0 = m/n$$

where X0 and Y0 are the respective initial populations of predator and prey.

Our first extension is to assume that the source of food for the prey (a plant or any other primary producer) is not inexhaustible, but is subject to the same rules as the other two species. We have thus a three-level system, whose behaviour is ruled by the following equations:

$$X' = -m.X + n.X.Y$$
$$Y' = -p.Y - q.X.Y + c.Y.Z$$
$$Z' = a.Z - b.Y.Z$$

where all the constants are positive. The generalization is straightforward. Each equation defines the rate of change of a population as the algebraic sum of two kinds of terms: one (call it the "self" term) describes the evolution of the population in the absence of other species, and is proportional to the population of the same species; its coefficient is positive for the primary producer level, negative for all others. Additional terms are proportional to the product of the population of this species by those at immediately upper and lower ecological levels, and they are positive for encounters with a lower level species, negative for encounters with one of the higher ecological level. It is very easy to extend this model to more than three levels, by applying the same concepts.

Our second extension is to allow for several different species at the same ecological level. The only effect of this extension is to include additional equations, as well as new "encounter" terms in the old equations. For two species of prey, with populations Y1 and Y2, the equations would become:

$$X' = -m.X + n1.X.Y1 + n2.X.Y2$$
$$Y1' = -p.Y1 - q.X.Y1 + c.Y1.Z$$
$$Y2' = -p.Y2 - q.X.Y2 + c.Y2.Z$$
$$Z' = a.Z - b1.Y1.Z - b2.Y2.Z$$

Programming the model in OOCSMP

For a given number of species, the model described above may be programmed easily in any general-purpose or continuous simulation language. However, as the numbers of levels and species per level increases, the numbers of equations and terms per equation could become unmanageable. In OOCSMP, however, the object-oriented facilities make it possible to model an arbitrary ecological system, with almost no limitations.

We use a hierarchical set of classes to represent the different ecological levels. The top class in the hierarchy, called Species, is defined as indicated in listing 1.

```
**************************************
*      Definition of the Species class      *
**************************************
CLASS Species {
NAME name
DATA M, X0, start
X:=STEP(start)*LIMIT(0,1000,XT)
XT:=INTGRL(X0,XP)
XP:=M*X
PLOT X TIME
}
```

Listing 1: OOCSMP definition of the Species class (file Species.CSM)

There are three attributes in the Species class: the start time (which makes it possible to simulate invasion of an ecosystem by a new species through the use of the STEP block); the initial population at start time (X0), and the coefficient for the "self" term (M). The equations describe the evolution of a population in the absence of other species, and contain a limitation to a maximum of 1000 and a minimum of zero, to prevent the absurd of negative populations.

The different levels in the ecological system are defined as classes derived from Species, as indicated in listing 2.

```
***********************************          DATA APlH, APrH
*      Definition of the Plant class     *  XP+= APlH*X*Plant.X
***********************************          XP-= APrH*X*Predator.X
CLASS Plant : Species {                    }
DATA AHPl                                  ***********************************
XP-= AHPl*X*Herbivore.X                    *    Definition of the Predator class    *
}                                          ***********************************
***********************************          CLASS Predator : Species {
*    Definition of the Herbivore class   *  DATA AHPr
***********************************          XP+= AHPr*X*Herbivore.X
CLASS Herbivore : Species {                }
```

Listing 2: OOCSMP definition of three ecological levels (file Levels.CSM)

Each class defines only the additional encounter terms with which other species influence its population, and adds the corresponding multiplication attributes. All the species in a given level are represented by their class name. The compiler generates a loop for generic terms of this kind.

A given complete ecological model, combining several species for each level, can now be programmed very easily, as in the example at listing 3.

```
INCLUDE Species.CSM                        * Level 2: predators
INCLUDE Levels.CSM                         Predator PrA("Pred1",-7.2,  2,  0, 0.36)
***********************************          Predator PrB("Pred2",-4,    2, 10, 0.20)
*        Actual species         *          System := PA, HA, HB, PrA, PrB
***********************************          System.STEP()
* Level 0: plant                           ***********************************
Plant   PA ("Plant", 0.4, 100,  0, 0.02)   *         Timer and show data         *
* Level 1: herbivores                      ***********************************
Herbivore HA ("Herb1",-1.28, 20,  0, 0.02, 0.36)   TIMER delta:=0.005,FINTIM:=40,PRdelta:=0.1,PLdelta:=0.1
Herbivore HB ("Herb2",-1.28,  5, 10, 0.02, 0.12)   METHOD ADAMS
```

Listing 3: OOCSMP definition of an ecosystem

Semiautomatic generation of an educational course for the Internet

The OOCSMP models we have developed are translated into standard object-oriented languages by means of a compiler we have built. Depending on the compiling options used, we can tailor the compiler to:

- Generate C++ code and test it under DOS.
- Generate Java code and test it under a Java interpreter or an applet viewer.
- Generate Java code and embed the applet in an html page. Automatically add buttons to select between different execution alternatives of the model.
- Provide special windows and widgets to modify the values of the model parameters and make it possible to perform "what if" experiments.

Figure 1 shows the procedure used to generate courses accessible through Internet.

To debug the models, we usually start by generating C++ code, provided with a graphic interface that makes it very easy to adjust the values of the different parameters in the system and shows the results of the simulations as graphical plots.
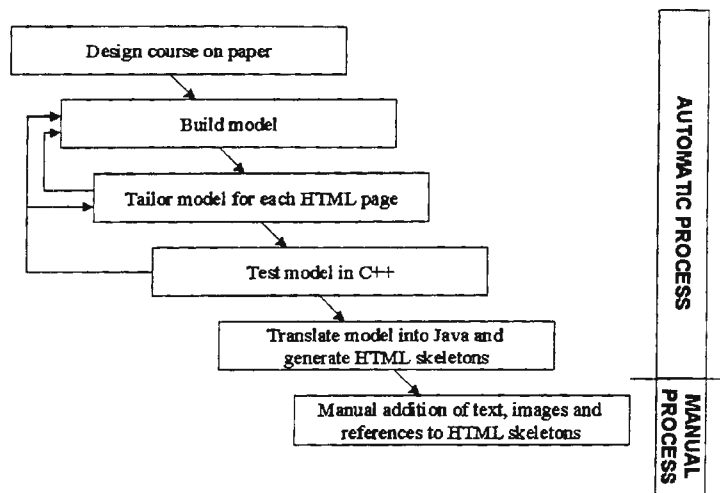
Figure 1. Steps followed to build Internet educational courses

Once debugged and adjusted, the same compiler, invoked with different options, can be used to generate Java applets and html skeletons. The same model, with a few adjustments, may be used to simulate different ecological systems, and thus generate the several html pages that make up the course.

The automatic preparation of the course ends here. Manual adjustment are needed to fill the html skeletons with explanations, images and cross references to other pages. The resulting set of pages is ready to be made available to the students through the Internet.

A practical course on Ecology

The procedure described in the previous section has been followed to generate a course on Ecology aimed at high-school students. The course consists of five pages presenting the following ecosystems:

- An isolated three-species system (one primary producer, one prey and one predator), with the appropriate parameters to bring the ecosystem out of equilibrium. The student can observe the periodicity of this kind of systems.
- A three-species system in equilibrium.
- A three-species system, originally in equilibrium, which is invaded after some time, first by a new prey, then by a new predator. The first invasion takes the system out of equilibrium. After the second invasion, the whole five-species system reaches a new periodical stability.
- The same system, invaded after some time, first by a new predator, then by a new prey. The periodical stability attained after some time is completely different to the preceding case.
- A five-species system with an user interface that lets the student modify the different parameters to perform experiments. Some of these experiments are suggested by the text of the page, but the student may perform many more.

As mentioned above, the main components of the HTML pages are the simulations of ecosystems presenting different features. The outputs of these simulations are presented graphically in two different ways:

- A plot of the evolution of the populations of the different species along time.
- An iconic representation of the same populations. The first four pages use preestablished icons to represent the different species involved. The last page allows the student to select the desired icons. The number of icons of a given species shown is proportional to its population at any moment. As populations wax and wane, icons appear or disappear.

121

The plot generator is automatically added by the OOCSMP compiler. The icon generator has been programmed by hand and added manually to the simulation applet. It is our future objective to extend the compiler so that it is able to generate this type of out put representations automatically.

Conclusion

The procedure described here makes it very easy to build Internet educational courses based on the simulation of physical, biological and other systems. The OOCSMP language, designed for the purpose, contains features that simplify programming the models. The compiler we have developed automatically generates C++ and Java code, as well as html skeletons. Some extra effort is required to complete the course, with the addition of textual explanations and an iconic representation of the evolution of the system. However, this effort is small, as compared to the actual development of the model.

The example proposed in the paper (a practical course on Ecology, executable on the Internet) can be found and tested at the following address: html://www.ii.uam.es/~epulido/ecology/simul.htm

In the future, we will extend the language and the compiler to integrate with the model the text for the html pages. The iconic output will also be provided as one of the standard output display options. In this way, the manual effort required to complete the course will become negligible. We also intend to develop a control system with access to a data base, so as to be able to control the performance of each student with the courses we generate. We are currently working to allow the dynamical generation of objects (species, in the ecological example), during the execution of the model.

The ecological model can also be improved by defining ecological niches. Species belonging to the same level, but occupying different niches would not compete among themselves. With this feature, more complicated and real ecological models could be included in the course.

References

[1] Y.Monsef, "Modelling and Simulation of Complex Systems", SCS Int., Erlangen, 1997.

[2] M.Alfonseca, "SIAL/71, a Continuous Simulation Compiler", in "Advances in Cybernetics and Systems", Ed. J. Rose, Gordon and Breach, London, Vol. 3, 1974, 1319-1340.

[3] D.Karnopp, "Bond Graph Models for Electrochemical Energy Storage: Electrical, Chemical and Thermal Effects", Journal of the Franklin Institute, Vol 324, 1990, pp. 983-992.

[4] A.A.Legasto Jr., J.W.Forrester, J.M.Lyneis, editors, "Systems Dynamics", North Holland, 1980.

[5] M.Alfonseca, "APL Continuous System Modelling Program: an Interactive Simulation Language", Advances in Engineering Software, Vol.1:2, 1979, 73-76.

[6] O.J.Dahl, K.Nygaard, "SIMULA - An ALGOL-Based Simulation Language", Comm. ACM, 9:9, 1966, pp.671-678.

[7] Digitalk Inc., "Smalltalk/V PM", Digitalk Inc., Los Angeles, 1990.

[8] B.Stroustrup, "The C++ Programming Language", Addison-Wesley, Reading (Mass.), 1991-1997.

[9] H.Elmqvist, S.E.Mattson, "An Introduction to the Physical Modeling Language Modelica", Proc. 9th European Simulation Sympossium ESS97, SCS Int., Erlangen, 1997, pp. 110-114. See also http://www.Dynasim.se/Modelica/index.html.

[10] IBM Corp.: "Continuous System Modelling Program III (CSMP III) and Graphic Feature (CSMP III Graphic Feature) General Information Manual", IBM Canada, Ontario, GH19-7000, 1972.

[11] Alfonseca, M.; Pulido, E.; Lara, J.; Orosco, R.: "OOCSMP: An Object-Oriented Simulation Language", Proc. 9th European Simulation Symposium ESS97, SCS Int., Erlangen, 1997, pp. 44-48.

[12] Volterra, V.: "Leçons sur la Théorie Mathématique de la Lutte pour la Vie", Gauthier-Villars, Paris, 1931.
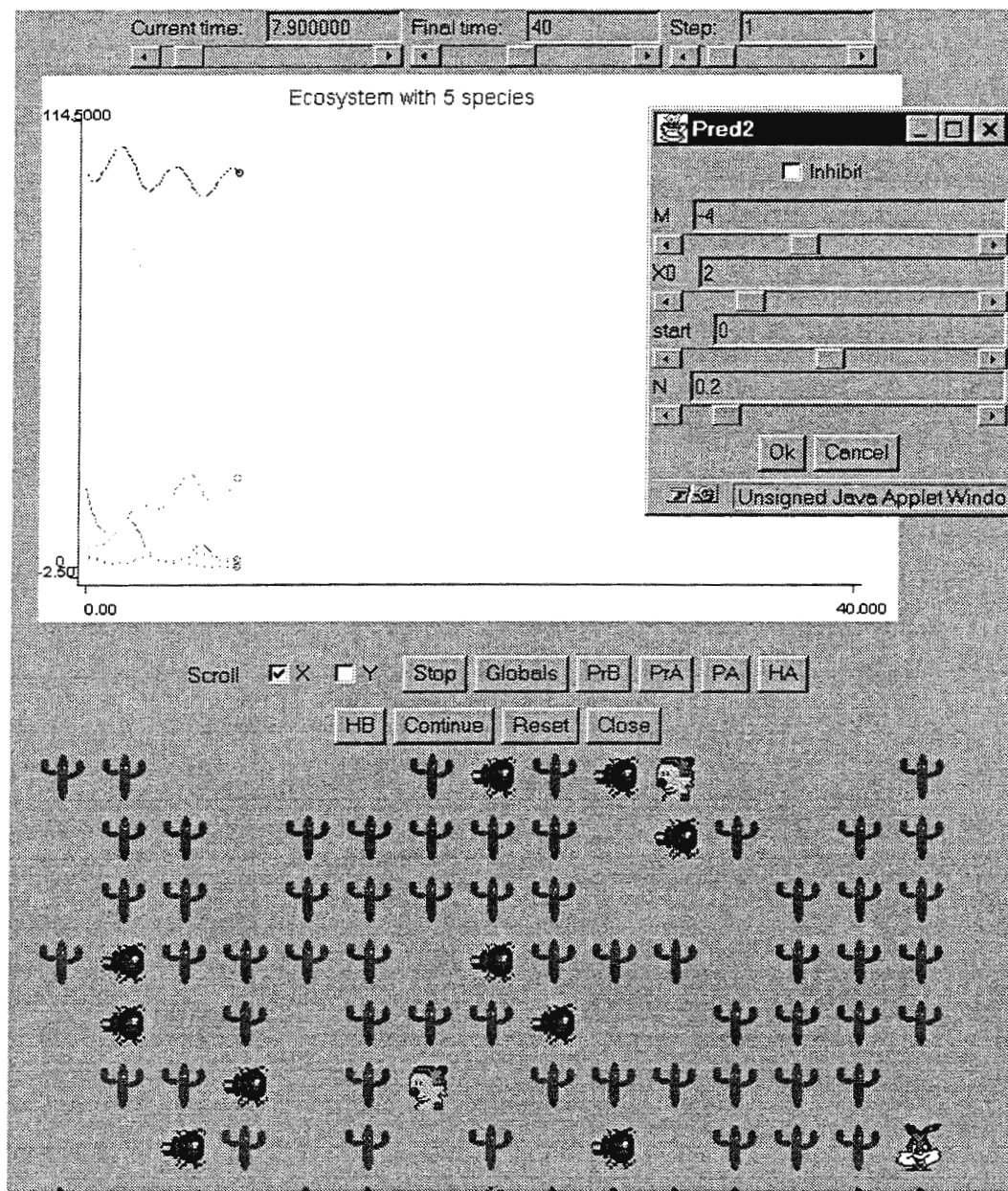
Figure 2. Simulation of an ecosystem with five species

SIMULATION OF SCANNING PAPER MACHINE MEASUREMENTS

Jouko Berndtson[1], John Shakespeare[2], Taisto Huhtelin[3], Heikki Koivo[4]

[1]Helsinki University of Technology, Otakaari 5 A, FIN-02150 Espoo, Jouko.Berndtson@hut.fi
[2]Valmet Automation, PL-237, Lentokentänkatu 11, FIN-33101 Tampere, John.Shakespeare@valmet.com
[3]Valmet Automation, PL-237, Lentokentänkatu 11, FIN-33101 Tampere, Taisto.Huhtelin@valmet.com
[4]Helsinki University of Technology, Otakaari 5 A, FIN-02150 Espoo, Heikki.Koivo@hut.fi

Abstract

In a paper machine the profile control of the paper web is based on scanning measurements. In this paper, scanned profile data has been analysed by statistical and spectral methods. The scanning measurements have been simulated in order to study effective scanning techniques. Profile estimation algorithms have also been introduced.

Online scanning profile measurements

The online scanning measurement gauges measure several paper profile quality characteristics, such as caliper, basis weight and moisture. In a typical machine there is one to three scanning frames, which are situated at the end of the machine just before the reel and/or at the end part of the drying section. The number of the scanning frames depends on the produced paper grade. For a multilayer board machine typically several scanning frames are needed. The measurement gauges are placed on a scanning frame where they traditionally traverse back and forth in cross machine direction above the paper web. New scanner control techniques enable variable scanner paths over the web (Fig. 1).

Currently the scanning speed of the measurement head across the paper web is around 20 to 40 centimetres per second. After each scan the measurement stops on the side of the web and then traverses backward. One measurement sequence can take almost one minute (in a machine having five meters width). The physics of the measurement gauges limit the scanning speed, for example the basis weight is measured with absorption of radioactive β-particles, in this case the strength of the radioactive source and size of its aperture ultimately limit the scanning speed. In modern paper machines the speed of the paper web can be more than 1500 meters per minute, for slower board machines the speed of the web ranges from 300 to 600 meters per minute. The high speed of paper web and the slow scanning speed, cause a zigzag shaped scanning path trajectory which is only few meters wide and several hundreds of meters long.

Profile Identification

The scanned paper profile variations are usually divided into three components [1,2]: Machine direction (MD), cross direction (CD) and random or residual component (R). The current partition is justified by generally used control algorithms, which control the MD and CD directions independently.

The profile identification algorithms can be classified into two main categories: input/output (I/O) model based identification and phenomenological identification. In most I/O –based identification algorithms an interaction matrix has been used to model the effect of CD actuators (such as the slice lip screw positions) on the profile shape. Typically the interaction matrix is identified by step response tests or with Recursive Least Squares (RLS) algorithms; and thereafter the CD profile can be identified with Kalman filtering. In connection with identification of the CD profile, orthogonal basis functions are often utilised. Low order orthogonal basis functions can describe controllable CD-variations. Fourier, Chebysev and Wavelet functions have been used as orthogonal CD profile basis.

Normal scanning path over the web   **Subscanning path over the web**
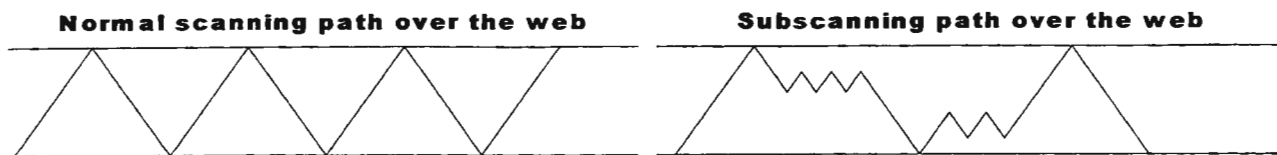
Figure 1. Different scanning paths. On the left, a traditional scanning path; on the right, a scanning path including subscanning.

Identification of MD/CD profiles without an I/O model is not mathematically well defined problem by itself. Whatever MD profile is chosen one can always find a CD profile so that the sum of the MD and CD profiles match exactly to the measurement signal.

To avoid this conspiracy some assumptions of the MD and CD profiles have to been made. For MD/CD profile estimation it is typically assumed that MD-variations occur on the whole width of the sheet. The CD variations on the contrary change much slower than the MD variations. CD variations are often assumed to be approximately time invariant. Most of the industrial algorithms are based on these assumptions. In this paper we have concentrated on MD/CD profile estimation from raw scanned data, without using any I/O models.

In industry MD and CD profiles are extracted from the scanned raw data by averaging. The standard industrial scheme for scan averages is to use an average over a scan in one direction for MD measurement, CD profile deviations are then averaged from the difference of the scanned raw profile and the MD average. Various avaraging windows can be used for CD-profiles, the most common being exponential and square windows [3,4]. In addition to averaging algorithms, bootstrap algorithms have also been developed. In bootstrap algorithms an ARMA model can be used for MD modelling and CD profile can be identified recursively, however the converge of the bootstrap algorithms can not be shown.

Basis function methods have also utilised for analysing the raw scanner data. For example Principal Component Analyses (PCA) based on a Karhunen-Loeve expansion can extract significant information from MD and CD variations.

Data collection and spectral analyses

The data used in analyses have been collected from two different plants. The age and condition of the paper machines varied greatly from plant to plant. Great variability is seen also in the spectrum content of the MD / CD profiles and in the stability of CD profiles.

The collected data sets consist of scanned and fixed point measurements, in fixed point measurements the scanner head have been stopped producing only MD directional measurement information. Fixed-point data were collected also simultaneously with scanning measurement, which enables relatively accurate MD / CD profile estimates for simulation purposes.

An example of estimated power spectrums of MD and CD basis weight profiles are shown in Figs. 2 and 3. In both figures the spectrums are shown with respect to the Nyquist frequency of data sampling, the frequency of the data sampling was 25 Hz. In Fig. 2 low frequency MD oscillations have been compared to CD spectra, note the periodic structure of the CD spectrum which is a side effect of the scanning process (periodicity starts for 1/39 Hz and we made 39 full scans). In Fig. 2 it is also noticed that low frequency oscillations below 0.02 Hz have somewhat larger amplitude in MD direction than in CD; this effect arises because wavelength of those oscillations are larger than the width of the paper web. In Fig. 3 MD and CD power spectrums during one scan have been compared, the spectrums have been calculated by averaged MD and CD power spectrums over 39 scans. As seen in Fig. 3 the CD spectrum has slightly larger amplitude than MD in region 0.02-0.1Hz, however the spectrums are practically almost indistinguishable, e.g. peak in spectrum at 0.1 Hz can indicate either MD variation having wavelength around 20 meters or CD variation having wavelength around 0.5 meter.
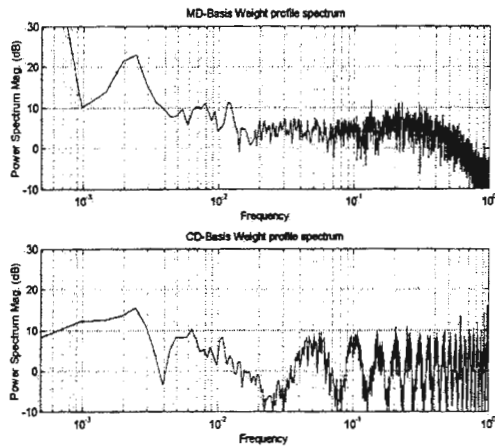
Figure 2. Power spectrums of MD and CD profiles. The spectrums have been calculated over 39 successive scans.

Figure 3. Averaged power spectrums of MD and CD profiles. Spectrums within one scan are presented.

As shown in Figs. 2 and 3 the power spectrum of MD and CD profile estimates cannot be separated by using two scanners; one for scanning and the other one for fixed point measurements. In the measurements presented here an assumption of classifying short wavelengths as CD variations and long wavelengths as MD variations is superficially plausible but very poor for separating the measurement data into MD and CD profiles.

Our laboratory measurements of succesive CD strips indicate that 70% to 80% of the total variation is not separable in this fashion. This is obvious from Fig. 4, which contains true CD measurements from 67 such strips. In this example, the random component accounts for 80% of measured variation.

Figure 4. CD profiles from laboratory analysis of paper strips.

Principal component analysis

As seen in spectral analyses the raw scanned data cannot be easily divided into MD and CD components within one scan. However if a set of few scans is used for analysing raw data, then in some cases it is possible to extract significant features in MD and CD directions. In principal component analyses (PCA) the data is divided into main components according to its variance contents (see e.g. [5]). Typically the PCA components are listed in order of importance. The first PCA-component show the dominant feature of the data (e.g.. average CD profile) and the successive components show the variability of the data around it's main component.

Second PCA component of Oven Dry Basis weight

Figure 5. Second largest PCA component calculated from successive scanned profile measurements. The profiles have been denoised with wavelet filtering.

PCA is successful in identifying both MD and CD changes with periods exceeding about 8 scans [6]. It is less useful in identifying shorter period variations which can be aliased from MD to CD.

In Fig. 5 an example of second PCA component of oven dry basis weight is shown. The PCA was calculated from scanned measurement data, which was denoised with wavelet filtering. As seen in figure the second PCA component of CD profile oscillates heavily from one side to another, the period of the oscillation is around 15 scan cycles. Fig. 5 serves as an example of a case, where the assumption of relatively constant CD profiles is not true.

Simulations and Subband scanning

The examples shown above were taken from different machines, and demonstrate that commonly used assumptions for making MD / CD profile estimations cannot be taken for granted. The complexity of MD/CD coupling and the overlapping of MD and CD spectrums suggests that MD / CD profile estimation should first be studied with simulations. In addition to using conventional scanning across the web we have simulated the possibility to utilise subscans (see Fig. 1).

The simulation model is particularly simple and it was in most cases based on industrial measurements. The basic model consists of routines, which construct the simulation web from industrial data collected by a particular measurement system, and then generate data provided by an arbitrary simulated measurement system. Effects of feedback control were not simulated because we were interested in relatively short time scale identification; as a result, I/O type models were not required. In addition to scanner code, several algorithms for calculating CD / MD profiles and analysing the results were also written.

Subscanning has three potential benefits, (i) it can provide high definition local CD profiles, (ii) it can give information of high frequency MD oscillations, and (iii) it can be used for breaking harmonic MD modes produced by constant scanning speed in connection with paper machine control system. In this paper we concentrate on the last case. A set of simulations with different subscanning lengths have been made in order to study the behaviour of the CD profile estimate as a function of subscan width. With each subscanning length a subscan position was randomly chosen for each scan from a fixed set of CD positions. To maximize the statistical significance of the results, all simulations were repeated with fourteen different starting positions in the industrial data.

In simulations an industrial MD profile with a constant CD profile was used. The CD profile was also collected from real plant data, but it was normalised to have a zero mean and unit variance. In order to compare different subscannings a CD profile estimate was calculated for each subscanning type, the CD estimate was calculated pointwise in MD direction over scanned profile points. The root-mean-square error of the estimated profiles for each subscanning type is presented in Fig. 6. In the simulations presented here, 7000 successive samples along subscanning path were used (one full scan was 428 samples). Other sample numbers were also simulated (from 2000 to 8000 samples), all the results have similar behaviour as a function of subscanning length. Each of the lower traces in Fig. 6 presents simulations with different starting

127
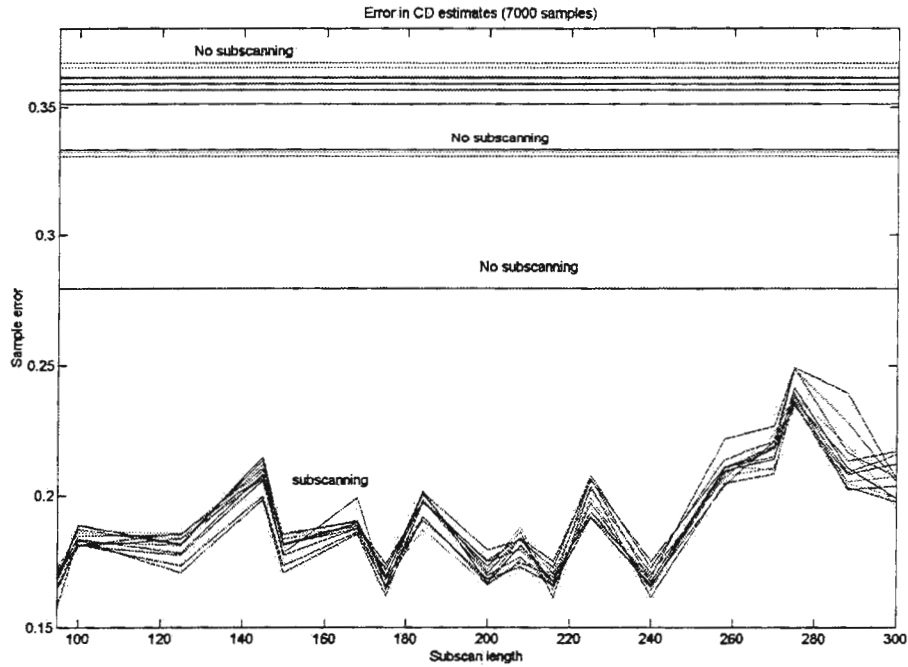
Figure 6. Square error between original and estimated CD profile as a function of subscan length. Each simulation was repeated with fourteen different starting positions.

points in the data, using the indicated subscan length. The upper constant traces show the result with conventional scanning, equivalent to a subscan length of zero.

As seen in Fig. 6 the error oscillates somewhat as a function of different subscanning types. This makes it difficult to give a precise recommendation for optimal subscanning lengths. Although the subscan results are significantly better than conventional scanning, most differences between individual subscan types are statistically less significant. However, longer subscans can be expected to give deteriorating results. More simulations should be made to define accurately the effective subscanning lengths, nevertheless even the simulations presented here show than significant improvement in profile estimates can be gained when appropriate subscanning is used.

The estimation accuracy was best for the greatest number of samples; when low number of samples were used, the subscanning estimates gave no improvement over conventional scanning. This is as expected, since for a given number of samples, subscanning provides fewer estimates of the profile than conventional scanning. As the number of samples increases, the improved quality of the subscanning estimates more than compensates for their reduced frequency.

Tentatively, short subscans can be expected to give best results, limited by the practical issues of decellerating and accellerating the traversing sensor platform. Each change of direction causes a slight data gap, so that the subscan width should be large enough that the missing data have a negligible effect. Moreover, it would be important to vary the subscan length where feedback controls are regulating the measured scan average data. If a constant subscan length were used, the potential would exist for a poorly tuned control to create a harmonic at the effective scan frequency. Breaking such harmonics requires that the scan period (and hence subscan length) should not be constant.

Conclusions

When MD / CD profile estimates are calculated from scanned paper machine data, it is necessary to employ some assumptions regarding separability of variability. In this paper we have studied the validity of

commonly used assumptions experimentally at two plants and through simulations based on experimental data. According to our sample data the assumption of stability of CD profiles and the assumption of separability using an independent MD reference measurement are not reliable in all cases. Moreover the power spectrums of MD and CD profiles were found to be severly overlapping.

We have also tested subscanning techniques in order to improve the quality of scanned profile estimates. According to our simulations subscanning seems to be a promising method for making better profile estimates without additional equipment. It also provides a simple means of avoiding or breaking harmonics induced by feedback controls. The current simulations were based largely on data from only one plant. More simulations with wider range of data should be made, before final recommendations of the subscanning can be made.

References

[1] K. Kristinsson, G.A. Dumont, Cross-directional Control on Paper Machines using Gram Polynomials, *Automatica*, 32(4) 533-548, April 1996.

[2] X.G. Wang, G.A. Dumont, M.S. Davies, Modelling and identification of Basis Weight Variations in Paper Machines, *IEEE Transactions on Control Systems Technology*, 1(4) 230-237, December 1993.

[3] G. A. Dumont, I.M. Jonsson, M.S. Davies, F.T. Ordubadi, Y. Fu, K. Natarajan, C. Lindeborg, E.M. Heaven, Estimation of moisture variations on paper machines, *IEEE Transactions on Control Systems Technology*, 1(2) 101-113, June 1993.

[4] R. Münch, Influence of Measurement Errors on CD Profile Quality, *Proc. Int. CD Symp. at XIV IMEKO Congress (Tampere Finland, June 1997)*, vol.XB, p.199-203, 1997.

[5] A. Basilevsky, Statistical Factor Analysis and Related Methods, Theory and Applications, Wiley series in probability and mathematical statistics, John Wiley & Sons, Inc. 1994.

[6] A. Rigopoulos, Y. Arkun, F. Kayihan, Identification of Full Profile Disturbance Models for Sheet Forming Processes, *AIChE J.* 43(3) 727-739, March 1997.

SIMULATION OF STOCK AND WHITE WATER SYSTEM DYNAMICS OF A MODERN PAPER MILL – A CASE STUDY

Ismo Laukkanen, Jari Lappalainen and Kaj Juslin

VTT Automation, P.O.Box 1301, FIN-02044 VTT, Finland
Ismo.Laukkanen@sah.vtt.fi, Jari.Lappalainen@sah.vtt.fi, Kaj.Juslin@vtt.fi

Abstract

Dynamic process simulation has been adapted as a powerful tool for studying the behaviour of industrial processes operating under transient conditions. The use of simulation in the paper industry, however, has so far been very limited. In this paper, process dynamics of a modern, closed-cycle paper mill was studied using a computational simulation model. The scope of the model was base paper production process, starting from the pulp storage towers and ending to the reel of the paper machine. The model was used in simulation studies including disturbance propagation analyses and grade change studies. The model was also used for operator training purposes. In operator training, a real process control system, Valmet Damatic XD, was used to operate the simulator. All simulation runs were accomplished using the Advanced Paper Mill Simulator (APMS) simulation program. Dynamic process simulation turned out to be a powerful tool for verification of process and automation design as well as for operator training.

Introduction

The major sections of a modern paper mill producing coated paper grades can be classified into base paper production process and finishing operations. The base paper production process includes the stock preparation, white water system, broke system, short circulation and the paper machine (Figures 1 and 2). Stock and whiter water system is major part of base paper production and in a modern paper mill it is highly integrated.

The optimal stock and white water system fulfills the following criteria [2]:

- steady in running and break situations
- easily controllable in grade change situations
- efficient material economy in steady-state and grade change situations
- minimal environmental impact
- minimal fresh water consumption
- maximal heat recovery
- easy to maintain and clean
- minimal capital investment

White water system in a paper mill is a compromise, which is found at the design phase by weighting these factors. The importance of the factors depends on the paper grade, the product mix and the type of the machine used. The wire retention of the machine is either low or high and it is also a significant factor in the design. In practice, the mill conditions like available space for equipment affect to the design of stock and white water system /11/.

On multigrade machines the controllability and operational efficiency in grade change situations are very important factors /2,11/. In optimal design, the mass tanks are quite small and the circulating loops are short so that same raw materials can be used for several grades. On the other hand decrease in tank volume decreases ability to eliminate high-frequency-disturbances.

The dynamics of the stock and white water systems depends strongly on the operational state of the mill. Some general rules for the operation are:

- *In the normal operation*, the level of the broke tank decreases, because more broke is used as a raw material than is produced in trimming and finishing operations. The level of the white water silo increases.
- *In the web break situation*, the whole production flows to the pulpers and to the broke system, so the level of broke tanks and silos increases. White water is used in break showers and in pulper dilutions, so the level of white water tanks decreases.
- *In a grade change*, the behaviour of broke and white water system depends on the change direction in basis weight.

FRESH WATER INTO PAPER MILL (5-10 m3 / ton)

FUNCTIONS	- Mass preparation - Transportation - Dilution - Cooling - Lubrification	- Transportatation - Dilution - Cooling - Lubrification	- Sheet formation - Cleaning - Strength - Shower water
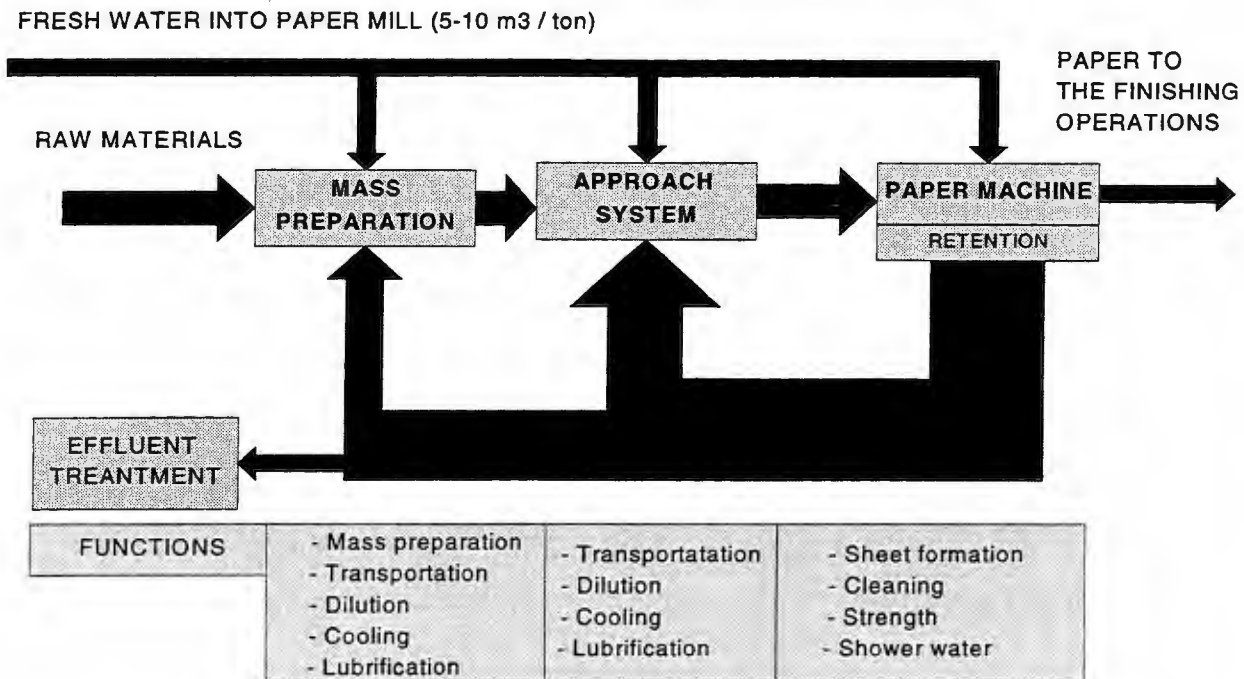
Fig. 1. Schematic diagram of the most important functions of water systems [10]

Decomposition of papermaking process

The major sections and objectives in the base paper production process are (Figure 2).

Stock preparation section starts from the pulp silos, which are storage buffers between pulp mill and paper machine. The *objectives* for the operation are to optimize required quality properties of the paper by controlling the pulp refiners, furnish composition in stock proportioning and flow of chemical additives. Also the pulp consistencies and tank levels are controlled to minimize the pressure-flow variation in the piping network. Possible sources for the problems in the control are long dead-times (still some lab. measurements), nonlinear dependence on chest level and flow, difficulties in consistency control and disturbances in stock proportioning

Short circulation starts from the machine chest and ends to the headbox of the paper machine. The *objectives* for the operation are to control the uniform basis weight and composition of paper, clean the furnish before feeding it to the paper machine, maintain steady flow to the headbox and keep the composition, consistency and the chemical state of the furnish in a proper state. Possible sources for the problems in the control are chemical disturbances changing the retention and flow disturbances caused by equipment or controller tuning.

Broke system starts from the paper machine pulper system. In pulpers, the paper broke is slushed and the broke is circulated through broke system to the proportioning, in which the composition of paper is controlled. The *objective* for the broke system operation is to improve material economy of the mill, so that also the quality properties of the paper are met. Long dead-times can cause problems in the broke system control (slow process dynamics).

White water system collects the filtrated water from the wet end of the paper machine called white water. The white water includes fillers, fibers and dissolved chemicals, and it is circulated back to the approach system to recover these raw materials in the saveall filter. The filtrates from the saveall filter, classified as clear and cloudy filtrates, are used e.g. for dilution and level control in the paper mill. The *objectives* for the operation are to improve material economy of the mill and to decrease the environmental load of the mill. Also the heat economy is improved. Problems in the control are caused by long deadtimes (slow process). In some operational situations the saveall filter can generate disturbances through the integrated white water system (see the case study).

Paper machine starts from the headbox and includes wire, press and dryer sections. In the wet end of the machine, the white water is removed from the web by using vacuum and mechanical pressure, which cause pressure difference over the web and forces water to flow through the web. Vacuum is used in the wire section vacuum while in the press press section mechanical load is used to enhance the dewatering of the wet paper web. After press section final moisture is evaporated in the dryer section.
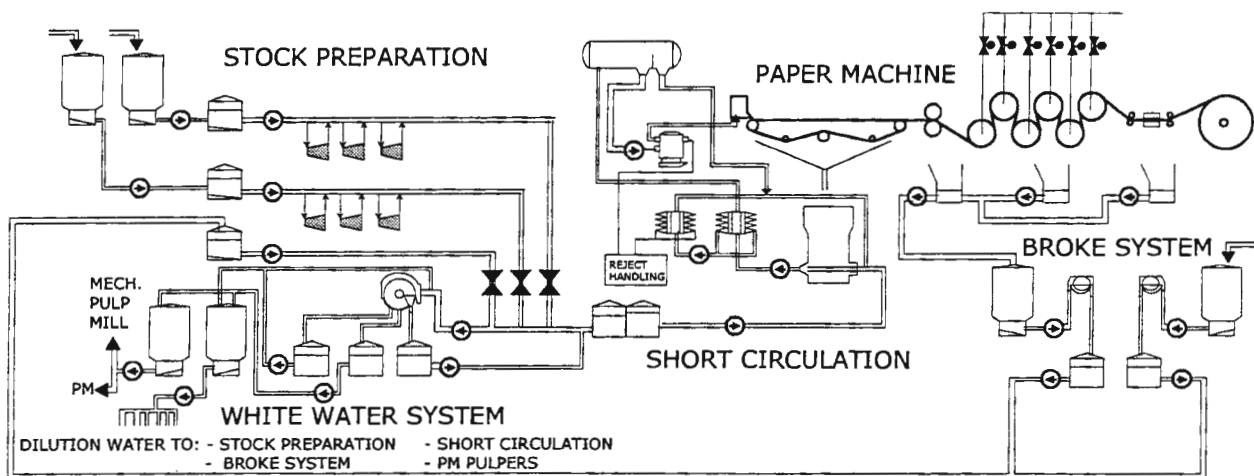
Fig. 2 The principal structure of the modern paper production process

DYNAMIC ELEMENTS IN THE PAPER MILL SIMULATION MODEL

In this study, mechanistic simulation models are used to simulate the process dynamics. As input data mechanistic models need equipment and control structure specific parameters. The dynamics of the process, in these kind of models, depend on equipment dimensions, operational parameters and dynamics of the equipment (table 1).

The solution of pressures, flows and temperatures for the whole piping network is based on numerical solution of the conservation equations for mass-, energy- and momentum [6]. In the APMS simulation program, these equations are solved simultaneously for the whole simulation model.

Definition in simulation model	Effect to the dynamics	Simplification
Dimensions of process equipment : tanks, pumps, valves, pipes etc.	The dead-time in the process depend on the dimension data of equipment. The relative height levels of the equipment are also needed and taken into account in high-fidelity dynamic models.	Only the tanks are dynamic
Major control circuits and parts of mill automation (see also table 3.2)	The effect of control circuit tuning is very important. Paper mill automation like logic sequences are needed in training models.	Ideal controllers
The dynamic of equipment – The shut-off time of valves, the pump dynamics	Important effect on the flow dynamics of a paper mill. Also the device controllers and power supply can be included in the simulation models.	Ideal process equipment
Control sequences e.g. automatic grade changes	Causes dynamic effects to the whole process	Not any control sequences included

Table 1. Definitions data in the mechanistic simulation model needed to model the dynamics of paper mill

Case –enhancement studies on stock and white water system dynamics in a closed cycle paper mill

Why dynamic simulation model was needed?

The starting point for the project was:

- New paper mill in the design phase
- Very strict environmental regulations for the new mill. The fresh water usage of the mill should be minimal as well as the environmental impact of the mill effluents.
- Part of the water usage philosophy was new. There were no practical experiences about operability and controllability of this kind of mill.

Dynamic simulation model was used to test and verify the process and control design structures. The model was also used for process analysis studies concerning the dynamic behaviour of stock and white water systems e.g. how certain disturbances propagate at the mill piping network? The environmental impact of the mill at specified water usage levels was estimated by simulation studies. Finally the simulation model was used for operator training, to get the operators understand the behavior and dynamics of the paper mill, water usage philosophy and its impact to environment. All these applications were performed before the start-up of the plant.

Scope of the simulation model

To simulate the dynamics of integrated stock and white water system, all major water usage points should be modelled. The following parts of the mill were included in the simulation model:

- Approach system from the stock towers to the blend chest
- White water system (completely)
- Complete broke system
- Fresh water preparation and warm water system
- Short circulation

- Paper machine
  - Former and press sections
  - Dryer section
  - Pulpers and also machine shower water system
- 57 control circuits of the mill
- Mechanical pulp mill and chemical pulp disintegrators were included in the model as water consumers

The flow network simulation model of the mill was formed consisting of thermohydraulic points at specified heights, pressures, temperatures, volumes and compositions. The process equipment pipes, pumps, valves, tanks etc. connect these points. Inside the whole network the conservation of mass, energy and momentum is calculated.

Model component	Number of components in the simulation model
Thermohydraulic point	245
Tank or silo	41
Pump	37
Pipe	133
Control valve	93
PI-controller	57
Analog signal	918
Binary signal	214

Table 2. Scope of the dynamic simulation model in terms of process equipment

The dynamics of the simulation model was 1.5 - 3 times faster than the dynamics of the real plant in HP 9000 series UNIX workstation. The time spent for developing the model was six man months and the modelling was done in co-operation with a Finnish paper mill. During modeling work both process and design structures were re-evaluated. As a result of the modelling project some fine-tuning in design was done. These modifications could also be implemented to the mill, because the modelling was done during the design of the mill.

Operator training

The operator training was the first and the most important applications of the model. There were several reasons why operator training was needed [10]:

1. The differences in operators' backgrounds.
   Some operators were old professionals while some were very new 'freshmen'.
2. The strict environmental limits of the mill
   Controllability of the mill environmental impact in all process states is the only way to keep the mill in operation.
3. Goals for training
   To verify and deepen process understanding and to create a picture how millwide process works and what are the interactions. For all operators it is important to understand the whole process.

The key operators, responsible for running the mill, were chosen for the simulator training program. The overall goal for training was to teach the process operations of the new mill and to broaden the operability scope of the mill personnel.

Requirements for the training environment

The starting point for the operator training program was to make the training environment very realistic, concerning the functionality and the operability of the system [10]. Some operators were old professionals and the requirements for the simulator were high. The requirements were met by using a high fidelity simulation program as a modeling environment and a real process control system (PCS) as a user interface to the simulator (Fig. 3). The simulated measurement data was transferred from the simulator to the process control system displays using Ethernet network. The operations and control actions, e.g. set point changes, were transferred from PCS to the simulator.

The pictures of the process control system were real ones and 15 pictures were in operational use during training. In addition to the learning the dynamics of the mill, the operators also get familiar with the functionality and the hierarchy of the PCS pictures, before the start-up of the plant. After the training session the same PCS system with the same pictures was installed to the real mill.

The training situations can be classified into three categories: normal operation of the plant, operation of the plant during operational transients e.g. grade change situations and operation of the plant during a malfunction or a disturbance.
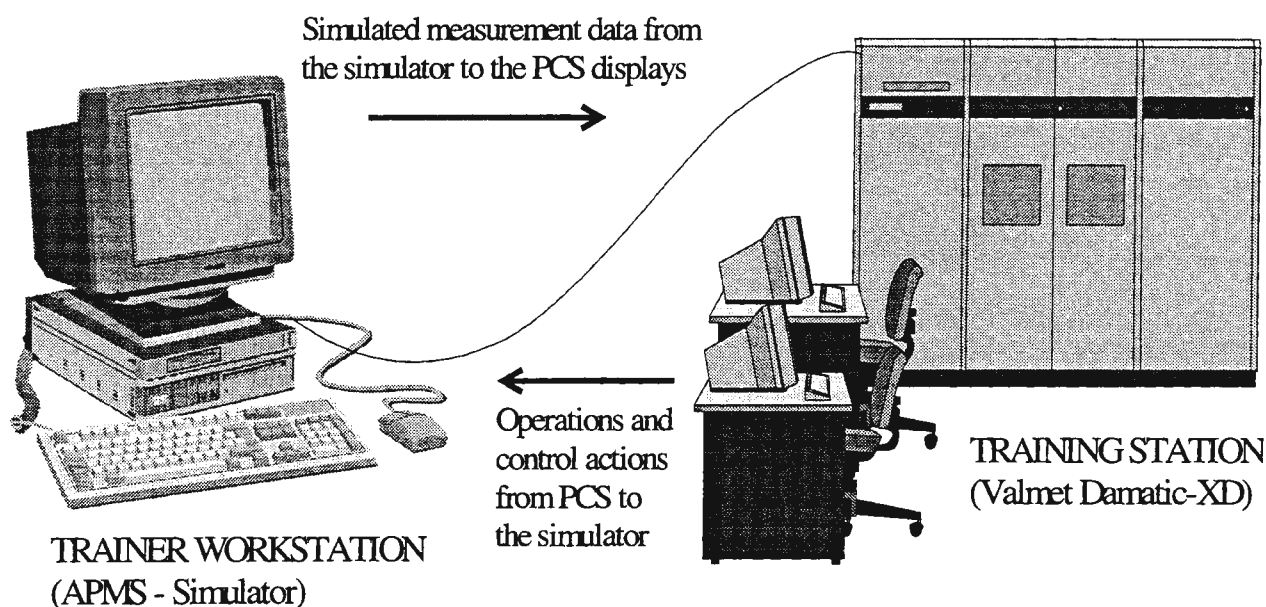
Fig. 3. The structure of the training environment. Operators used a real process control system (Valmet Damatic XD), as a user interface to the simulator. This made training sessions very realistic.

All these cases could be trained to the operators. Essential part of the training environment was the instructor station (Trainer workstation, Fig. 3). From there the instructor created the training situations. Instructor could easily cause malfunctions (e.g. web breaks) and disturbances (e.g. flow disturbances) to the whole process. After that he could follow the dynamics of the process from the trend curves and also follow the actions of the operators during these abnormal situations. Important part of the training was to go through the best operational practices in different situations.

Results from simulator training
The simulator training was a part of the mill operator-training program. The experiences from were positive - simulator training is an efficient way to teach dynamic behavior of the mill and teach how the process should be operated during normal and abnormal situations. As an extra benefit the start-up phase of the plant was easier.

Process analysis studies

The simulation model was used for process analysis studies before the start-up of the plant. These studies included chemical disturbance propagation analyses, production capacity change analyses from water circulation point of view and grade change studies. Results from these analyses were plotted as a function of time into trend curves like in Appendix 1.

Case – The disturbances in the saveall filter operation caused by the level change in the blend chest

As an example of the simulation studies the effect of a change in blending chest level to the saveall filter operation was studied. Note that all variables in the simulation results below are scaled.

The set point of the blending chest level is decreased by one meter (3.5 m -> 2.5),
the controllers were loosely tuned (Appendix 1. Figures 1,2)
1. The stock proportioning controller, which is in cascade with the level control, so this change affects to paper composition. The flows to the blending chest and to the saveall filter decreases (App1. Fig 2).
2. The flow rate to the saveall, however, is kept constant, so the flow of the chemical fiber is not enough. More broke and mechanical fiber is taken to the saveall filter and it can be seen from the furnish composition. In practice, this change causes variation into clear and cloudy filtrate consistencies. This effect has been notified at the paper mills.
3. The short time effects of the change can be seen in furnish composition change at the headbox (App 1, Fig 1)
4. Long time effects of the change come through long circulation (fluctuations through integrated flow network

APMS simulation environment

The dynamic simulation tool used in this project is developed for the simulation of pulp and paper mills [6-9]. A simulation model is constructed by combining unit models from the model libraries: thermohydraulic flow, basic process component, special paper mill component and automation system model libraries. The appropriate modules are selected from the model libraries, the process related query forms are filled and the components are connected to each other. The model structure can be changed any time and the model can be expanded without need to recompile or link the program. Simulation is an interactive process in contrary to batchwise calculation procedures.

The mathematical models describing the process components are based on the first principles of physics and chemistry, combined with phenomenological models and empirical correlation. Tuning parameters are used to depict differences due to various component structures and properties of the stock, and to refine the models. Models have different levels of accuracy, the user can choose the appropriate model fidelity and easily change it later [6-10].

The same simulation model can be used for many purposes both before and after the plant commissioning. This approach facilitates profitable use of dynamic simulation throughout the complete life cycle of a paper mill. The model can be used for verification of the process and automation design, for analysis purposes, planning the operational procedures and as a basis for a training simulator. For operator training it is either connected with a program emulating control room displays or to a real automation system [8,10].

Summary and recommendations

Minimization the environmental impact of paper mills by minimizing the fresh water usage is a very actual problem in many mills. Computational simulation provides an efficient tool to study and analyze the dynamic behavior of paper mills. The reliable and realistic simulations needed require accurate and

detailed models, which shall also be extensive and yet fast enough. Dynamic simulation, based on mechanistic models, turned out to be a powerful tool for these kind of studies. The same simulation model can also be used for verification of process and automation design as well as for operator training.

Acknowledgements

REFERENCES

1. Syberg, O., and Wild N.W. (Editors), Introduction to Process Simulation, 2nd ed., TAPPI Press, Atlanta, Georgia, 1992.

2. Paulapuro, H., Advanced course in paper technology - Stock and white water systems, Helsinki University of Technology, Laboratory of paper technology (In Finnish)

3. Laukkanen, I., Applications of dynamic simulation in chemical process industries, M. Sc. thesis, Helsinki University of Technology, 1993, 78 p.

4. Winter, P., Computer aided process engineering: The evolution continues, Chemical engineering progress, vol. 2 (1992), pp. 76-82

5. Hudson, M., Dynamic simulation as a design tool, The Chemical Engineer, no 2 (1992), pp. 24-28

6. Niemenmaa, A., Lappalainen, J., Laukkanen, I., Juslin, K., Proceedings of the 5th New Available Techniques Conference, "An Advanced Platform for Dynamic Simulation of Paper and Board Mills", 4-7 June 1996, Stockholm, 9 p.

7. Juslin, K., and Pollari, H., Proceedings of ESS'94, "Mass and Energy Dynamics of Displacement Pulping Process", European simulation symposium, Istanbul, Turkey, October 9-11, 1994, San Diego CA, The Society for Computer Simulation, 1994, 6 p.

8. Honkoila, K., End-user Experience from plant analyzer to full scope training simulator development with APROS, Proceeding of 10th European simulation multiconference, Budapest, Hungary, 1994, 6 p.

9. Niemenmaa, A., Välisuo, H., Lappalainen, J., Laukkanen, I., Juslin, K., Dynamic simulation of paper and board mills: a case study of an advanced grade change method, TAPPI Engineering Conference, Chicago IL, Sep. 16-19, 1996, TAPPI Press, Atlanta, 1996, pp. 491-498

10. Laukkanen, I., Silvennoinen, J., Lappalainen, J., ja Juslin, K. 1997, Enhancement studies on operation and control of water usage in paper mills, Proceedings of Tappi Engineering and Papermaker's Superconference 1997, 6-9.10.1997 Nashville, Tennessee, pp. 67-73

11. Ryti, N. and Paulapuro, H., Chapter IV, Stock and white water systems in Tappi pulp and paper manufacture series, Vol. 7, Paper machine operations, Tappi Press, 1991, pp 85-96

Fig. 1. Effects of blending chest level setpoint change on the stock composition at the headbox. The short time effects can be seen here and the long time effects come through long circulation.

APMS	Effects of level change

blending chest level (sp)	2.000e+00	5.000e+00
blending chest level (meas.)	2.000e+00	5.000e+00
mass fraction of filler	3.500e-03	5.000e-03
mass fraction of mechanical fibre	5.500e-03	7.000e-03
mass fractions of chemical fibre	1.600e-03	1.900e-03
filler content of paper	2.500e-01	2.800e-01
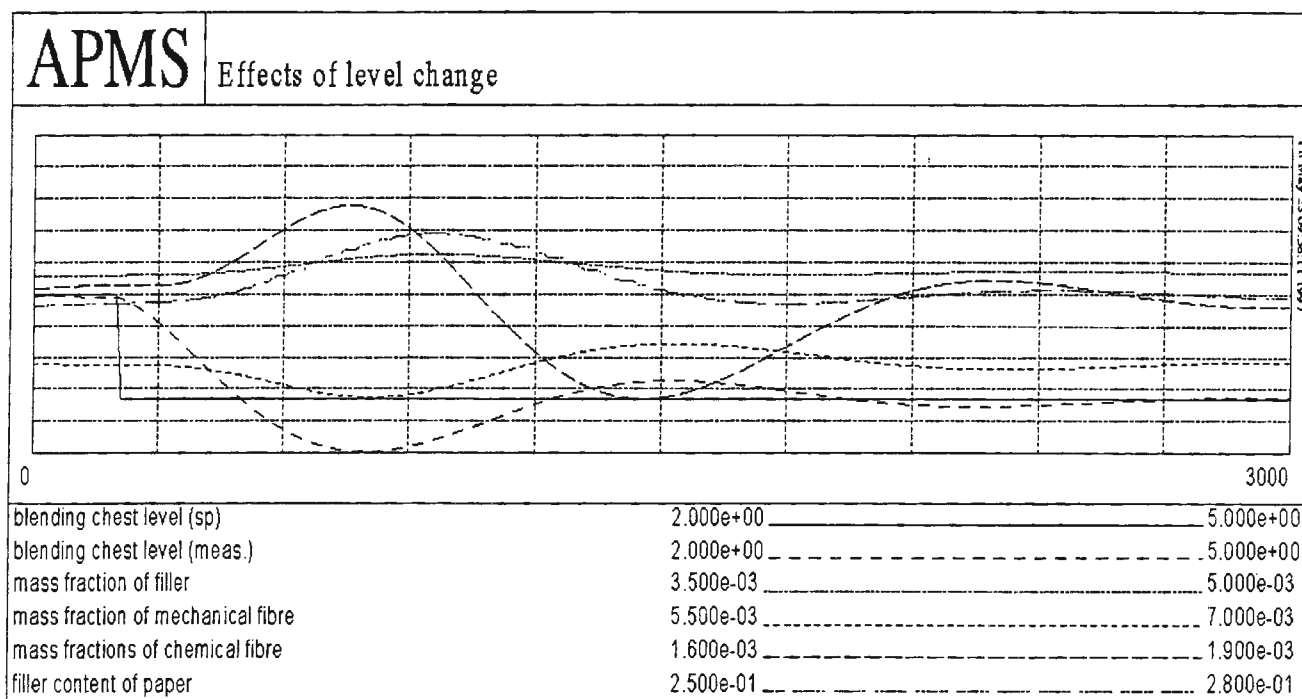
Fig. 2. Effects of the blending chest level setpoint change on the stock proportioning. The fluctuation in flows and composition can been seen in fluctuating paper properties (e.g. strength).

APMS	Mass proportioning

mechanical pulp flow	1.000e+02	3.000e+02
broke flow	1.500e+01	4.000e+01
chemical pulp flow	3.000e+01	1.000e+02
mech. pulp fraction (to saveall)	1.000e-02	3.000e-02
chem. pulp fraction (to saveall)	2.000e-03	1.500e-02
filler fraction (to saveall)	1.000e-03	4.000e-03
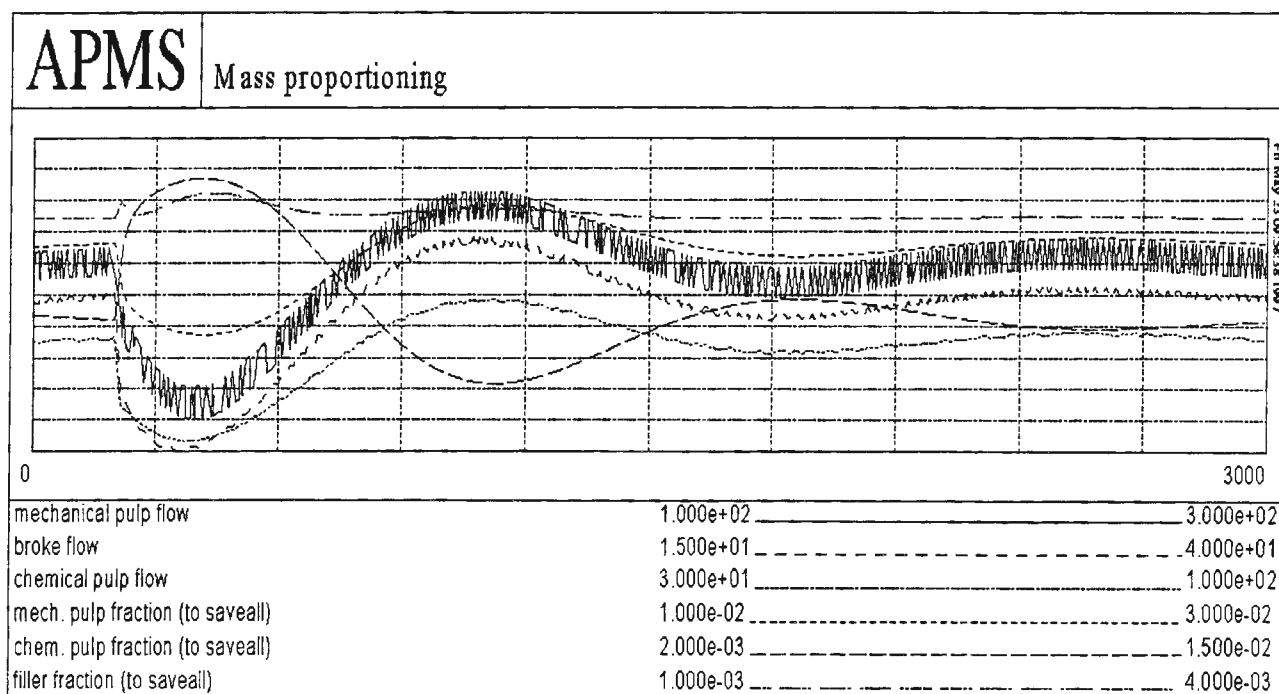
Dewatering of a Deformable Porous Media:

A Simulation Study of the Press Nip of a Paper Machine

Karin Danielsson[1], Jonas Funkquist[2]

[1]STFI, Box 5604, SE-114 86 Stockholm, Karin.Danielsson@stfi.se
[2]STFI, Box 5604, SE-114 86 Stockholm, Jonas.Funkquist@stfi.se

Abstract

Different models of a press nip in a paper machine are compared. By fitting the models to measured process data collected from a pilot paper machine, it is showed that all the models describe the data equally well. However, simulations show that the internal behaviour differs substantially. This fact demonstrates the importance of validating the model to get a reliable quantitative model.

Introduction

For processes where the physical mechanisms are well understood, simulation may increase the knowledge about the process. However, when only approximate theories are available, *model validation* becomes necessary in the development of a quantitative model, especially if the model is to be used for prediction or extrapolation. Validation is not only important for checking the theory and the model equations, but can also be useful for finding shortcomings in the numerics and the coding.

This study compares some different models describing dewatering in a press section of a paper machine. The models studied are chosen as representative models found in the literature (see *e.g.* [1]). The purpose is twofold: to demonstrate the need for thorough validation and to determine which model structure would be suitable for validation of a physically based model against online data collected from a paper machine. A qualitative comparison between the process models and data is presented. For that purpose, experiments have been conducted on the pilot paper machine EuroFEX at STFI in Sweden.
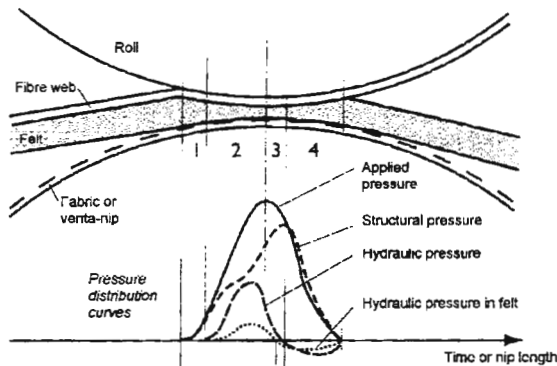
Figure A. *Roll press nip (from [2])*

figure a shows a wet pressing roll press nip and how the total applied pressure is believed to be divided between a hydraulic and a structural pressure, according to the commonly accepted qualitative theory [2]. The hydraulic pressure forces a water flow from the fibre web into the felt and from the felt into the open roll. The structural pressure is caused by the compaction of the fibre web. A model thus has to describe the fluid flow of water (and probably air) and the consolidation of a porous medium.

Several attempts have been done to describe this situation mathematically. Published models range from rather simple ones consisting of one static press equation, *e.g.* [3], to models describing the fluid flow in one dimension for water [4] and for water and air [5, 6], as well as in two dimensions [7]. The model by Kataja [4] is a simplified version of the two-phase model found in [8]. All of these models are more or less physically based, but include also a number of empirical relations describing the properties of the paper web and the press felts. These relations typically describe the permeability as a function of porosity as well as the compressibility of the paper web and the press felt.

The most detailed wet pressing model of today is developed by Bloch [7]. It describes the flow of water and air in both the thickness and the machine direction. It would of course be interesting to also include such a model in a comparative simulation study. However, it is beyond the scope of this study due to the very cumbersome

139

implementation procedure required. The model structure is also far too complex to be feasible in identification activities such as parameter estimation, particularly when the observability of the process is limited.

Model Descriptions

The model by McDonald and Kerekes [3] is the most simple model considered in this study. It consists of one static equation relating the dryness after the press nip to various press variables like the linear load, the machine speed and the initial moisture content. There are also parameters for describing the permeability and compressibility of the paper web and the amount of rewetting, *i.e.*, the flow of water back to the web occurring when the paper web expands. This model may predict how the dryness after the nip changes, however, it is unable to describe *e.g.* the dryness change within the nip and how the hydraulic and structural pressures are distributed. Internal variables like these are of interest, *e.g.* when using the model to understand the fluid flow and fibre network consolidation process within the nip.

The models by Kataja [4] (KATAJA) and Riepen et al. [5] (RIEPEN) are based on the assumption of one-dimensional flow in the thickness direction of the paper web. In both these models the transport of water in the web under a compression force is described by Darcy's law. Combining this law with Terzaghi's principle and the mass balance equations for the fluids results in a nonlinear diffusion equation determining the moisture transport through the web structure. One difference between the models is that KATAJA considers only the flow of water through the porous fibre mat, whereas RIEPEN includes also the flow of air for the expansion side of the press nip. Another difference between the two models are the description of the press felt. In KATAJA the felt is described purely as a flow resistance which varies with the degree of compression. In RIEPEN the felt is described in the same way as the fibre web. It is, however, not possible to specify the compressibility independently from the web compressibility, the same value being used for both materials. In addition, the initial porosity of the wet web and the felt in RIEPEN has to be equal.

The key parameters describing the physics of the dewatering in a fibre web are the ones coupled to the permeability and the compressibility of the web. In the model by McDonald and Kerekes [3] these are lumped together into permeability and compressibility factors, whereas KATAJA and RIEPEN use the slightly more detailed empirical relations:

$$K(\phi) = k_0 \frac{\phi^3}{(1-\phi)^2} \qquad \text{(KATAJA)} \qquad (1)$$

$$K(\phi) = A \left(\frac{\phi}{1-\phi} \right)^B \qquad \text{(RIEPEN)} \qquad (2)$$

for the permeability K, and

$$P_s(s) = E_0 \frac{r}{1-r} \begin{cases} r = \dfrac{s}{s_0}, & \text{during compression} \\[2mm] r = \dfrac{s - \mathcal{E}s_m}{s_0 - \mathcal{E}s_0}, & \text{during expansion} \end{cases} \qquad \text{(KATAJA)} \qquad (3)$$

$$P_s(s) = p \cdot \left(\frac{X}{D} \right)^{-q}, \ X = \frac{\phi}{1-\phi}, \ \phi = \frac{s-\phi_0}{1-s} \begin{cases} D = 1, & \text{during compression} \\ D < 1, & \text{during expansion} \end{cases} \qquad \text{(RIEPEN)} \qquad (4)$$

for the compressive stress P_s. Here ϕ is the porosity, ϕ_0 the initial porosity, X the void ratio (void volume/solid volume), s the degree of compression, and k_0, A, B, E_0, p and q positive parameters specific for the pulp. For certain choices of parameters, the difference between the two permeability relations is found to be small in the porosity interval of interest. It should also be noted that the compressibility function (4) does

not give a structural pressure equal to zero when there is no strain. *figure c* shows an example of the relations (1)-(4).

The length of the press pulse is properly determined by the compression of the fibre web, the felt and the roll cover and it is therefore a result of the model solution. None of the above models, however, include that calculation, but they consider the press nip length as constant.

Simulation results

In order to establish the required step sizes for the numerical integration, a numerical sensitivity analysis was performed. This analysis showed that the paper should be divided into 11 layers in KATAJA and 20 in RIEPEN to get a solution which is almost independent of the discretization in the thickness direction. In RIEPEN also the felt is discretized. The sensitivity test, however, shows that this is really not necessary, at least not for the operating conditions tested here. Considering the felt as one layer gave almost the same result as for 30 layers. The number of required discretization steps in time (or machine direction) differs substantially between RIEPEN and KATAJA. For the former it suffices with 40 steps whereas the latter needs at least 2000 time steps. The execution time is approximately the same though. It is interesting to notice that RIEPEN becomes unstable when the time integration step is chosen too small. This is an unexpected behaviour which should be further investigated.

Prior to parameters estimation, it is important to test the sensitivity of different model parameters on the measurable outputs. It gives a hint on which parameters are more or less easy to estimate. As expected, the sensitivity to the key parameters in Eqs. (1)-(4) is high. From the analysis, it is also clear that the similar parameters used to describe the felt properties are of less importance. Physically this implies that the properties of the felt do not affect the dewatering to any degree of significance. However, this may be a result of the simplified felt models used. Since practical experience shows that the felt design is important, a more rigorous felt description may probably result in a different conclusion concerning the felt sensitivity.

The ability of the three models to explain the process data external to the press nip is shown in *figure b* where the three lower curves are model inputs and the upper plot shows the output dryness, both the measured and the simulated from the three models discussed in this paper. The models have been fitted to this data using the Maximum-Likelihood method combined with a Newton-Raphson search method. The parameters estimated in the model by McDonald and Kerekes [3] are a permeability coefficient, a compressibility factor and a rewetting factor. In KATAJA the estimated parameters are E_0 and k_0 in Eqs.(1) and (3), and in RIEPEN A and p in Eqs. (2) and (4).

Since different permeability and compressibility relations are used by KATAJA and RIEPEN, the difference in the estimated model parameters is most easily illustrated by comparing the permeability and compressibility functions using the identified parameters. This is done in *figure c*, which shows the estimated permeability relations on the top and the stress-strain relations on the bottom. The large differences between the relations used in the two models indicate that the models can be expected to give significant different predictions of the fluid flow and web consolidation within the nip.

From the comparison with the process data, it is seen that all models manage to describe the data well. To investigate if there are some differences between the models, simulations were performed to investigate the internal variables. The model of McDonald and Kerekes only describe the input-output relation and is therefore not included in these simulations, where the operating point in **table a** has been used. Note that the model parameters are now chosen such that the models are as equal as possible.

figure d shows the effect permanent deformation may have on the web dryness predicted by the two models. Comparing the two elastic cases (solid), it is seen that the decrease in dryness due to network springback is smaller in RIEPEN. The two simulations with different degree of permanent deformation
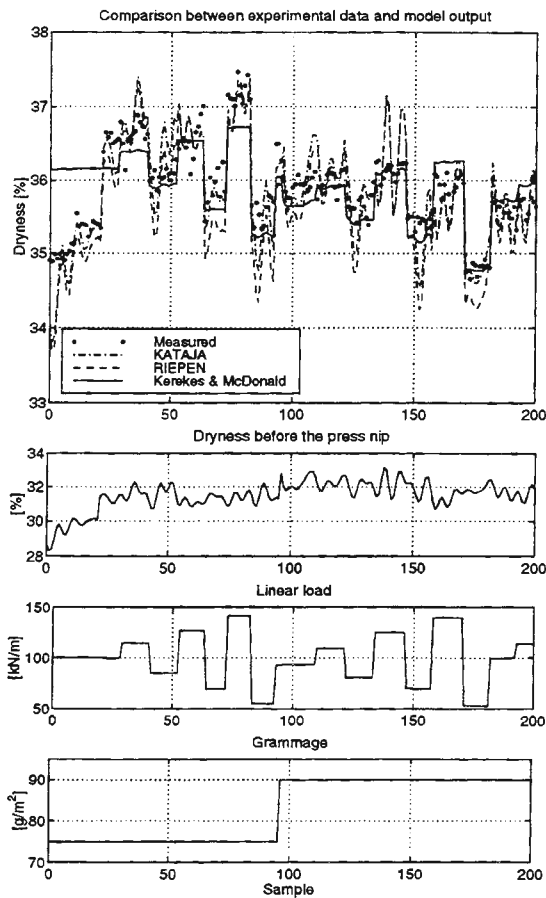
Figure B. The upper plot shows the measured dryness after the press nip and simulations of the three models. The three lower plots shows the input data.
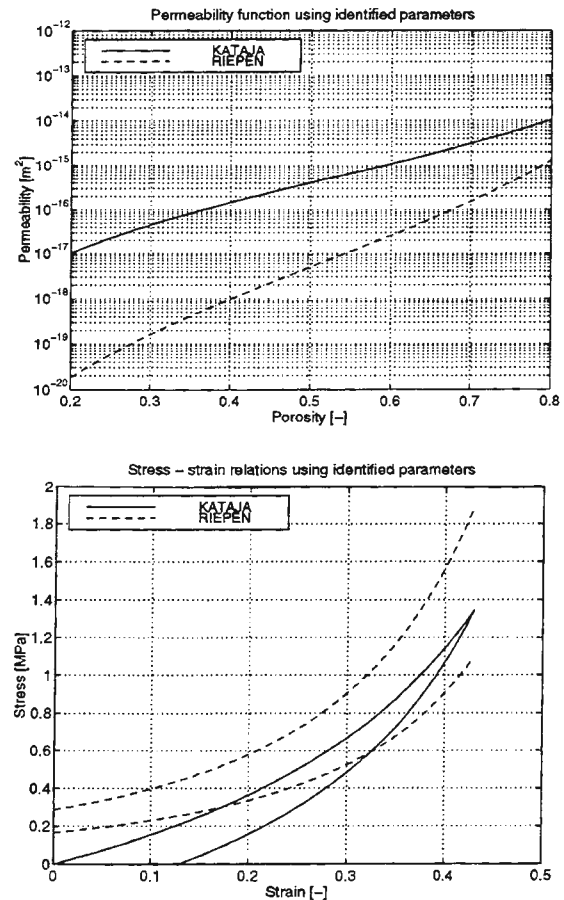
Figure C. Permeability and compressibility functions when using the parameters identified using the experimental data.

Table A. Operating Point

Process condition	Value
Dryness before the nip	30 %
Machine speed	450 m/min
Linear load	100 kN/m
Grammage	90 g/m^2
Nip length	40 mm
Viscosity	1.0

(dashed and dash-dotted) show that a similar effect can be reached by decreasing the forces of the network available for spring back.

How the two models describe the effect of the paper permeability is seen in *figure e*. The typical range for the web permeability is 10^{-16} to 10^{-14} m^2. Both models, level out when the permeability gets close to the felt permeability which is 10^{-13} m^2 for these simulations.

The dependence on the paper compressibility for KATAJA is shown in *figure f*. The RIEPEN model is excluded since the paper compressibility cannot be changed independently from the felt compressibility making a comparison difficult. The dryness curve may be explained as follows. High compressibility makes compression impossible resulting in no dewatering. Decreasing the compressibility increases the dewatering up to a certain point. At this point, the web becomes so soft that the layer closest to the felt rapidly compact and act as a blocking for the flow as the solid line in the right plot shows.

Discussion

From the results shown in *figure b*, it is clear that all three models manages to follow the data quite well. There is no big difference in the behaviour of the models by KATAJA and RIEPEN. The simple model by

Figure D. *The effect of permanent deformation on the in nip web dryness. Solid lines represent a completely elastic case and the other two curves represent different degrees of permanent deformation. The RIEPEN model includes semi two-phase flow in the exit side of the nip.*
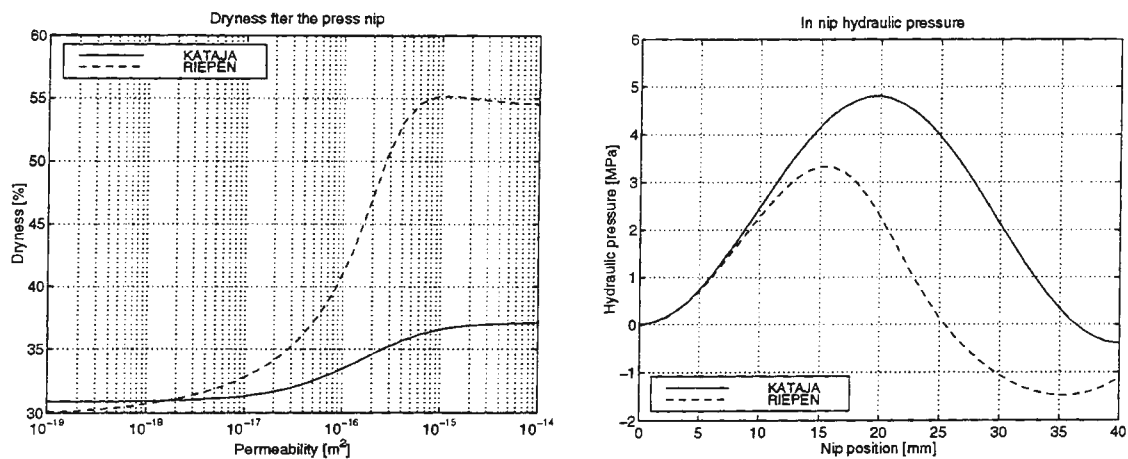
Figure E. *The effect of the paper permeability on the outgoing web dryness and thickness. The initial dryness is 30 % and the felt permeability is 10^{-13} m^2.*

McDonald and Kerekes also seems to be able to follow the changes quite well, at an average. It is however, insensitive to rapid changes in the dryness of the web going into the nip.

The estimated permeability and stress-strain relations shown in *figure c*, however, reveals that the internal behaviour differs substantially. The tests performed so far are not sufficient for determining which model can best explain the available data, but a more rigorous validation procedure has to be applied, using *e.g.* more process data and maybe also disturbance modelling. It is also clear that the model by McDonald and Kerekes [3] can only be used to model process data external to the press nip

figure d shows that assuming one-phase flow (water only) and the web completely elastic, gives a decrease in web dryness in the expanding side of the nip. As is also seen from the figure, this effect is decreased by allowing air to flow back or letting the web undergo permanent deformation. It is obvious that to quantify the relative influence of these two phenomena, measurements of the web thickness, as well as the web dryness, is probably needed. This exemplifies how important it is to measure the right quantities to make possible model identification and validation. Note also that it has to be measurements, preferably online, from the actual process since the relative influence of the mechanisms may be very process specific.

The different sensitivity with respect to the web permeability as seen in *figure e* is remarkable considering that the models are run with the same parameter values under the same operating conditions. It once again shows the importance of performing a thorough validation before a quantitative confidence for the model can be achieved.
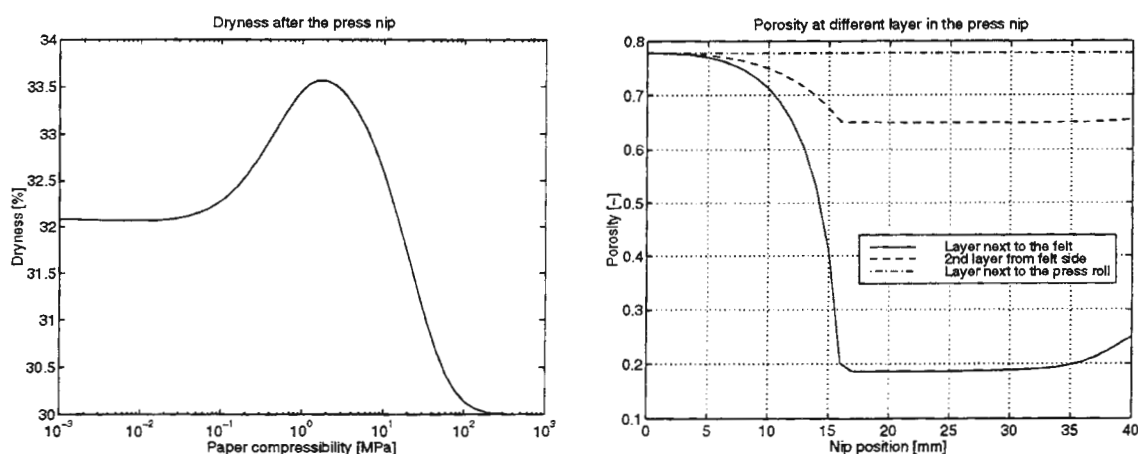
Figure F. The effect the compressibility has on the outgoing dryness of the web is seen in the left hand side plot. In the right hand figure it is seen that the layer of the paper closest to the felt collapses when the web is very compressible.

It seems that the natural choices for further exploitation are the models by Kataja [4] and Riepen et al. [5]. However, when *model validation* is a part of the modelling activity, the available experimental data usually limits the degree of detail which is possible to describe by the model. The complexity of the desired model structure is thus in a lower limit bound by the complexity of the physics and an upper bound given by the information content of the experimental data. If this set is non-empty, a feasible model structure should exist for the specific process-data case. Therefore, at this point, the one-equation model [3] should not be rejected since it is still unclear whether the available process data are informative enough to make validation of [4] or [5] possible. It might be that with the possibilities of today to collect data, it is not possible to validate more complex models than *e.g.* [3]. In such a case, the main effort should be put into improving the observability of the press section in the paper machine in order to make more complex models possible to validate.

Conclusion

The aim of this study has been to find a suitable model structure for identification and validation of a press nip in a paper machine. It has also been able to demonstrate how important it is to validate the proposed model structure for the process under consideration.

Although there are some uncertainty about the information content of the available process data, the most promising models for our purpose are the one-phase model developed by Kataja [4] and the semi two-phase model by Riepen et al. [5]. Enough physics is included to make possible predictions at different operating points. If properly identified, they should also contribute to increase the insight of what mechanisms that dominate the process under similar conditions as for industrial paper production

[1]   K. Danielsson and J. Funkquist, "A Literature Review of Wet Pressing Models 1996," STFI, Internal Report 1997.

[2]   P. Nilsson and K. O. Larsson, "Paper Web Performance in a Press Nip," *Pulp Paper Mag. Can.*, vol. 69, pp. T-438-445, 1968.

[3]   J. D. McDonald and R. J. Kerekes, "Decreasing Permeability Model of Wet-Pressing with Rewetting," presented at TAPPI Engineering Conference, 1994.

[4]   M. Kataja, "SimPres: an implemented pressing model," personal communication, 1995.

[5]   M. Riepen, A. M. J. Sinon, and B. M. P. Mulder, "The Integrated Paper Machine Dewatering Model (IPM)," presented at IDS '96, Krakow, Poland, 1996.

[6]   K. B. Jewett, W. H. Ceckler, L. H. Busker, and A. C o, "Computer Model of a Transversal-Flow Press Nip," presented at AIChE Symp. Ser., 1980.

[7]   J.-F. Bloch, "Transferts de Masse et de Chaleur dans les Milieux Poreux Deformables Non Satures: Application au Pressage   du Papier," . Grenoble: Institut National Polytechnique de Grenoble, 1995, pp. 267.

[8]   M. Kataja, K. Hiltunen, and J. Timonen, "Flow of Water and Air in a Compressible Porous Medium. A Model of Wet Pressing of Paper," *J. Phys. D: Appl. Phys.*, vol. 25, pp. 1053-1063, 1992.

STATIC SIMULATION OF STEAM PRESSURES IN PAPER MACHINE GRADE CHANGES

Paavo Viitamäki

Helsinki University of Technology,   P.O.Box 3000, FIN-02015 HUT, Finland,
Paavo.Viitamaki@hut.fi

Introduction

World wide competition in the pulp and paper industry has increased of the number of grades and also grade changes in paper mills. Thus an efficient grade change (GC) has become an important competitive factor. It can be estimated that 3 % of the production time of LWC (Light Weight Coated) paper is being used on GCs (Ranta & al. 1992). This means the decrease of sales income worth 4 million USD/year and loss of about 340 000 kWh/year electric and 2600 GJ/year heat energy per paper machine due to off-quality production. Usually the breaks are also increased due to the increased number of GCs (Kallela 1996). It is estimated that at least 50 % of the losses could be saved if automatic GCs could be used. However the products that automation vendors have developed for GCs have not been successful. Operators are not using them although nearly every paper machine has got one installed. Operators consider them too slow. They prefer manual actions instead (Viitamäki 1993a, Viitamäki 1993b, Viitamäki et al 1995).

A typical grade change consists of a calculation of target values and a dynamic coordination of paper machine speed, pulp stock flow to the white water circulation and steam pressures in the drying section It is crucial to a successful GC that the new target values are accurately known. The target values for steam pressures are the most important due to long time constants and dead times in the drying process of paper web.

The main objective of this research is to develop a tool for paper machine work teams to facilitate the execution of GCs. This is done by making a simulator that calculates the target values for paper machine speed, pulp stock flow and steam pressures. An additional objective is to develop models for grade change automation that would be easy to tune.

Paper machine grade change

A Grade change is a product quality change on a paper machine. Most of the GCs are usually basis weight transitions. The scale of basis weights in production is run in a cycle (Fig 1).
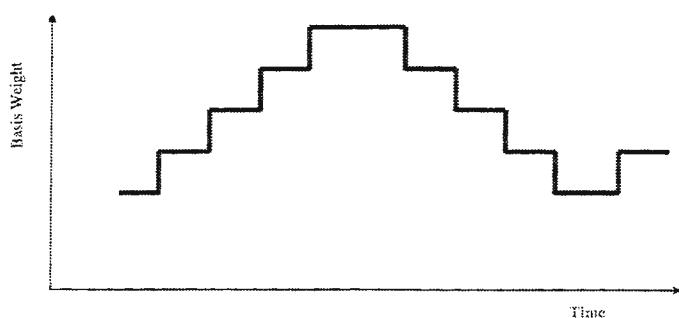
Figure 1.   A production cycle of different basis weights

The cycle is optimized so that the basis weight changes are as small as possible. The ideal condition is that the acceptance ranges of sequential grades overlap.

The most common method to execute a GC is ramping. An open-loop method suits well to CG because there are not exact target values for the basis weight and moisture. It is satisfactory if the values will hit inside acceptance range after the GC. More information about more advanced algorithms is in Huhtelin 1996, Ihalainen and Ritala 1996 and Välisuo 1997.

steam pressure

thick stock feed

machine speed

Timing planned in advance
fixed shape
formula to compute steam pressure target
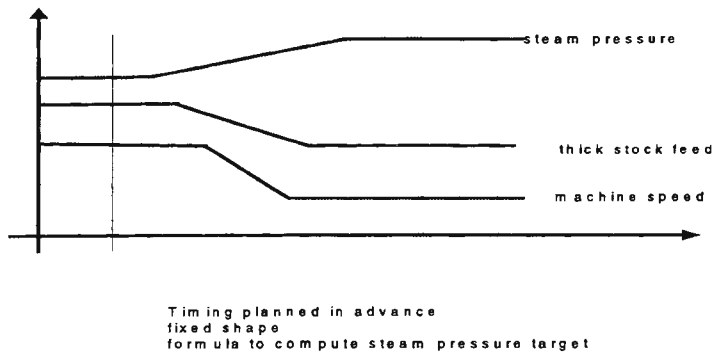
Figure 2. Grade change by ramping of manipulated variables

The procedure for GC with ramps starts with the calculation of target values for manipulated variables such as paper machine speed, stock flow and steam pressures of the drying section (fig. 2). The stock flow controls the production rate and also the basis weight but the machine speed determines the basis weight more distinctively. Drying is controlled with steam pressures but often only last steam groups are used for controlling purposes. The target values for steam pressures are much more difficult to calculate than the other target values. For example also properties of raw materials, condition of the paper machine, basis weight, moisture and speed have effect on the drying rate. Machine tender usually gets the initial target values from the records of the previous runs. In this research a simulator is developed to calculate these target values for the operators.

Modeling and simulation

The simulation model is developed by using a measurement data from a paper machine that is high dimensional, noisy and internally correlated data. It was gathered during normal runs because extensive testing is not possible due to high expenses of lost production. The problem was dealt with projecting the samples onto lower dimensional hyper space with orthogonal, uncorrelated new variables by using PCA (Primary Component Analysis) and PLS (Projection to Latent Structures). Process data matrix X can be for example decomposed into principal components by using PCA. It is a common procedure for reducing dimensionality of the variable space (Wold et al., 1987; Mardia et al., 1989; Jackson, 1991).

$$X \in R^{m \times n}$$

m = sampled measurements

n = number of process variables

$$X = t_1 p_1^T + t_2 p_2^T + \ldots + t_A p_A^T + E$$

$p_i \in R^n$,   loadings vectors

$t_i \in R^m$,   score vectors

$E \in R^{m \times n}$,   residual matrix

PLS was used to make a global and local model for a GC PLS is used to find out the variations in the process variables that are most influential in the calculation of target values (Höskuldsson 1988, Geladi and Kowalski 1986, MacGregor and Kourti 1995). In the local model the data was classified according to a priori known quality measurements and by using scores from PCA and PLS analysis. The simulations were verified by a test set of data that was not used for the development of the simulation model.

The classifying was done by fuzzy c-means clustering. In this method each data point belongs to a cluster specified by a membership grade (Bezdek 1981). The algorithm is based on minimizing an objective

function that represents the distance from a cluster center. Both of these classification methods are used in this research.

Results

The steam pressures were simulated using the test data set for steam groups 4,5 and 6 (fig. 3). These were the three last steam groups for the drying of the paper.
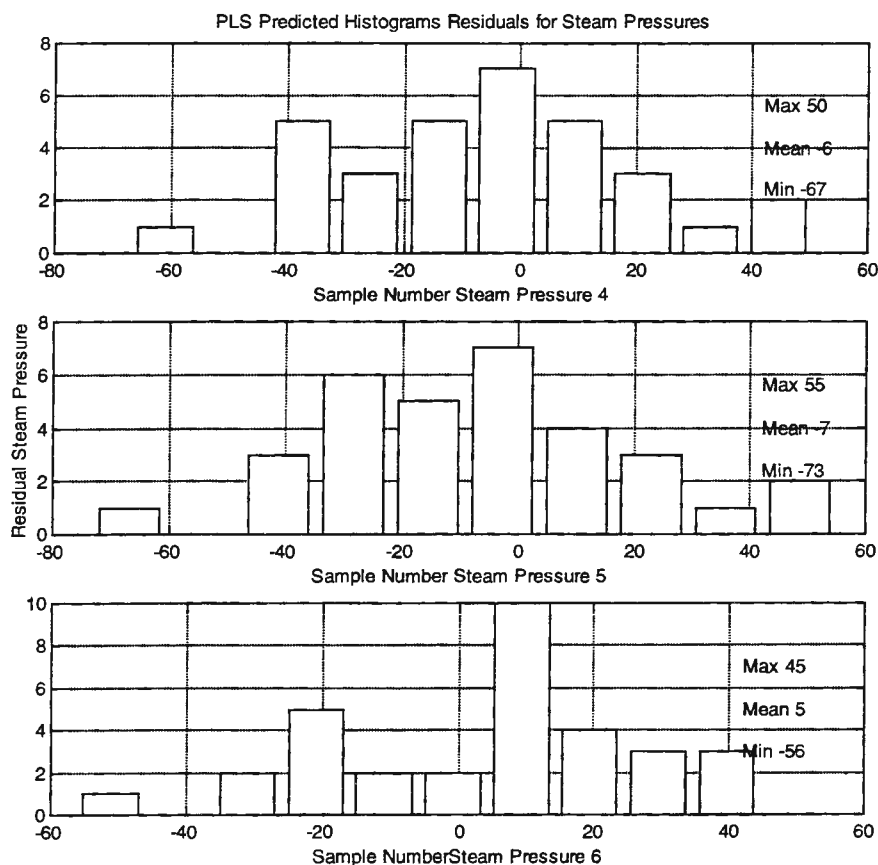
Figure 3. Histogram of Predicted Steam Pressures Residuals in the Global Model

Six principal components were used for the model. This was based on a thumb of rule that the model should explain over 85 % of variance. Most of the residuals of the predicted values of steam pressures stayed within +/-30 kPa. The requirement presented by the paper mill personnel was +/- 20 kPa. However, the results are considered good because grade changes were run manually It is assumed that the requirements can be fulfilled if automatic changes were used because a more systematic procedure could then be applied for the grade change. On the other hand the samples that deviate the most were a new type of GC that were not included in the actual modeling data set. This kind of error could also be minimized in reality because only one set of steam pressures need to be calculated at a time and new type of changes could be included in the modeling data set if needed.

Figure 4. Classification by Dry Basis Weight and Dry Production Change

Figure 5. Grade Changes classified by Principal Components

It was assumed that predictive properties of the PLS model could be improved by using local models. The local models were done first tested by classifying data into three groups by dry basis weight and dry production rate (fig. 4). Then PLS models were identified from each classification groups. Similar procedure was also repeated by using principal components for the classification (fig. 5) but data was now classified into five groups. Three principal components were includes in the local models

Conclusions

Local and global PLS models were constructed for a paper machine. A simulator that calculates target values for steam pressures for grade changes was designed.

The simulator that was based on a global model was chosen the best alternative because of its simplicity and easiness of database management, although it is difficult to compare the models due to different structures. Local models did not improve the simulation accuracy.

The future research consists of improving the predictive properties of the models and developing dynamic models for GCs. Also the nonlinear nature of the process will be included in the models.

REFERENCES

Bezdek J. C., "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.

Geladi P. and Kowalski B.R., "Partial least squares regression: A tutorial", Anal. Chim. Acta, 185, 1, (1986), p. 1-17.

Höskuldsson A., "PLS regression method", Journal of Chemometrics, 6, 2, (1988), p. 211-334.

Huhtelin T., "Total production management - How to implement it", Automation Technology Review 1996, Editor Ranta J., Published by VTT Automation, Espoo Finland, p. 6-15.

Ihalainen H. and Ritala R., "Optimal Grade Changes", Control Systems '96 Preprints, Chateaux Halifax, Halifax, Nova Scotia, Canada, Apr 30-May 2, 1996, p. 213-216.

Jackson J.E., "A User's Guide to Principal Components", John Wiley and Sons Inc., New York, 1991.

Kallela J., "Paradigms in automation. User participation to automation design", Espoo 1996, Technical Research Center of Finland, VTT Julkaisuja - Publikationer 817, 148 p. + app 74 p.

MacGregor J.F. and Kourti T., "Statistical process control of multivariate processes", Control Eng. Practice, Vol 3, No. 3, (1995), p. 403-414.

Mardia K.V., Kent J.T. and Bibby J.M., "Multivariate Analysis", Academic Press, London, 1989.

Ranta J., Ollus M. and Leppänen A., "Information Technology and Structural Change in the Paper and Pulp Industry", Computers in Industry vol. 20, 1992, p. 255-269.

Välisuo H., "Managing grade changes and other transitions in paper-board production", Automation 97, Finnish Society of Automation, Helsinki, Finland, 23-25.9.1997.

Viitamäki P., "Grade Change on a Paper Machine", Automation Days -93, Finnish Society of Automation, Helsinki, Finland, 11-13.5.1993a, p. 241-243, (in Finnish).

Viitamäki P., "Grade Change on a Paper Machine", Proceedings of the 12th International Conference on Production Research, Lappeenranta, Finland, 16-20 August, 1993b, p. 587-588.

Viitamäki P., Välisuo H. and Ventä O., "Information Technology Based Grade Change in the Process Industry", Automation Days, Finnish Society of Automation, Helsinki, Finland, 3-5.5.1995, p. 355-357, (in Finnish).

Wold S., Geladi P., Esbensen K. and Ohman J., "Multi-way principal components and PLS analysis", Journal of Chemometrics 1, 1987, p. 41-56.

ALICES : an Advanced Object-Oriented Software Workshop for Simulators

G. Rouault[1], D. Pieroux[2], and U. Van Houte[2]
[1]CORYS T.E.S.S. France, [2]TRACTEBEL Belgium

0.   Abstract

Reducing simulator development costs while improving model quality, user-friendliness and teaching capabilities, is a major target for many years in the simulation industry. It has led to the development of specific software tools which have been improved progressively following the new features and capabilities offered by the software industry.

Unlike most of these software tools, ALICES (which is a French acronym for "Interactive Software Workshop for the Design of Simulators") is not an upgrade of a previous generation of tools, like putting a graphical front-end to a classical code generator, but a really new development. Its design specification is based on previous experience with different tools as well as on new capabilities of software technology, mainly in Object-Oriented Design. This allowed us to make a real technological "jump" in the simulation industry, beyond the constraints of some traditional approaches.

The main objectives behind the development of ALICES were the following :

- Minimizing the simulator development time and costs : a simulator development consists mainly in developing software. One way to reduce costs is to facilitate reuse of existing software by developing standard components, and by defining interface standards.
- Insuring that the produced simulator can be maintained and updated at a minimal cost : a simulator must evolve along with the simulated process, ant it is then necessary to update periodically the simulator. The cost of an adequate maintenance is highly dependent of the quality of the software workshop.
- Covering the whole simulator development process :
  - from the data package to the acceptance tests and for maintenance and upgrade activities;
  - with the whole development team, even if it is dispatched at different working sites;
  - respecting the Quality Assurance rules and procedures (CORYS T.E.S.S. and TRACTEBEL are ISO-9001 certified).

The development of ALICES was also done to comply with the following two main constraints :

- Building an open system : ALICES will evolve and need to interact with foreign systems; new tools will easily be added.
- Building a modular system : ALICES will also be used for simulator upgrade and update. It will be possible to integrate ALICES progressively into an existing simulator maintenance and evolution process.

1. Introduction

With the rapid power increase of computers and their ever decreasing price, the industrial simulation world is facing two major challenges.

On the one hand, progress is expected in model quality, in user-friendliness and in teaching ability. With each step of progress in one of these fields, simulators become even more interesting educational solutions to new categories of potential customers.

On the other hand, the building of better simulators requires various types of expertise, especially in computer and software technology, in process simulation, in ergonomy and in graphic drawing. As a result, highly skilled personnel is needed and simulator costs are nowadays strongly dependent of these miscellaneous software issues. As an example, software development represents around 80% of the development costs of a middle size training simulator in the electrical industry.

For many years, special software tools have been designed for easing the development of simulators, and thus reducing the development time. They achieved their goal by facilitating a few time consuming tasks of the development process.

Though this was already a big progress, a lot of work remained to be done, especially in integrating the various software components and testing them properly. Following generations of simulation software workshops tried to improve the solution to these problems.

Unlike these tools, ALICES is not an upgrade of previous tools but has been developed from scratch as a coherent, integrated and appropriate answer to the building of modern simulation systems. It is a new generation engineering software workshop that includes all what is necessary to build and maintain various types of simulators.

Indeed, apart from offering sophisticated and modular modelling tools, ALICES takes care of important simulator development aspects such as management of input data, product testing, progressive and modular integration, documentation, project deve lopment and co-ordination.

ALICES has been developed by a Consortium of European companies, among which CORYS T.E.S.S. (France) and TRACTEBEL (Belgium), two companies which are world leaders in the simulation business. ALICES has been awarded the European EUREKA label.

In this paper we first expose the way ALICES works, its objectives and constraints; we then present different important aspects in a simulator development and how there were treated. Some technical information is then provided.

2. ALICES objectives and constraints

The main objectives behind the development of ALICES were the following:

Minimising the simulator development time and costs: as already mentioned, a simulator development consists mainly in developing software. One way to reduce costs is to use a generic simulator core coupled with software components describing the simulated real world objects. Moreover, since many of these components can be designed with generality in mind, they can be saved into general standard component libraries and be ready to be used in many projects .

Insuring that the produced simulator can be maintained and updated at a minimal cost: a simulator must evolve along with the simulated process, and it is then necessary to update the

simulator as the process is upgraded. The cost of an adequate maintenance is highly dependent of the quality of the software workshop.

The development of ALICES was also done to comply with the following two main constraints :

Building an open system: ALICES will evolve and need to interact with foreign systems: new tools have to be added easily. Moreover, the ALICES architecture allows it to be linked with already existing object components that have not been specifically developed for ALICES.

Building a modular system: ALICES will also be used for simulator upgrade and update. ALICES structure is such that affective recovering of other simulator software pieces is possible. This is the key to allow a smooth transition from a n old generation simulation system to ALICES.

3. Important aspects of ALICES

To fulfil the requirements given in the preceding section, special attention has been given to the following aspects, which are essential to an appropriate simulator development and maintenance. We present here the solution that has been chosen within the software workshop ALICES.

Input data

It is very important to trace the original process data, to know which value is being used, what kind of previous calculations, hypotheses or tuning have been performed on the data. ALICES offers in this area an integrated relational data base facility for keeping and handling the original process data through integrated calculation parameters sheets. These are the link between the data used by the simulator and the original input process data. Thus, it is very easy to know how the value of each parameter is computed.

It is also possible to identify which data have been tuned to match the simulated process with the real one.

With the help of the built-in documentation facilities, it is thereby easy to keep trace of all the data and the tunings used. Such a knowledge is essential for a proper simulator maintenance and evolution.

Tests

Adequate test facilities are primordial to guarantee an efficient simulator development. The sooner and the deeper a model is tested, the lesser it contains potential problems that could appears only at the simulator development final stage, resulting in time and cost expensive debugging sessions. It is also important to know what has been tested. ALICES provides in this field various tests facilities, including the possibility of writing and replaying test scenarios.

Tests can be performed as soon as a group of objects is assembled. It is not necessary to waste time in compilations, it is just necessary to define the model that is going to be tested, which is also necessary for assuring an easy trace of the tests performed.

Documentation

Documentation is an essential aspect when building and maintaining a simulator. ALICES offers all the facilities necessary for creating, printing and consulting on-line documentation. Whatever is developed with ALICES can be easily and interactively documented .

Project development

A proper organisation is an essential key to succeed in a simulator development. ALICES provides project development tools, allowing the configuration and follow-up of various types of

projects, control on the rights of use and general organisation of the software development to be performed. These tools have been devised to cope with a proper development methodology. For example, it is easy to allocate to each user the tools necessary for his work, so that starters with the workshop do not get overloaded will all the existing possibilities .

Co-ordination

In any software project involving more than one developer, co-ordination is a critical item. Much time can be wasted whenever people working together are not communicating properly.

This problem has generated a special attention to the development of ALICES by helping developers to interface properly their respective applications. The use of gigantic databases is avoided, and most of the interface labels between systems built by different people can be generated automatically, with the help of the process input data. This greatly reduces potentially serious co-ordination mistakes.

4.   How does ALICES work?

Let us consider first the efficient three level approach used to analyse and design a complex process in the industrial world. Such a process can be described in terms of basic objects strongly interacting together inside subsystems, while the subsystem assembly forms the entire system.

The main point with this approach is that a single object or a subsystem can be described, analysed and simulated much more easily that the whole system. Besides, a direct advantage of this approach is to ease the system upgradings, since an upgrading concerns often only a small subset of one or a few well-identified subsystems, the other ones being not affected. ALICES organisation features naturally this three level approach (Fig. 1).

Level 1 is used to create the basic objects of the simulation. The dynamics of these objets is described by discrete time transitions from one objet state to another or by ordinary differential equations (ODEs). Moreover, each object can be made fairly general by incorporating parameters in its definition. This allows the building of general object libraries reusable within different simulation projects.

Two major features have been incorporated in ALICES to simplify the modeller's task.

Firstly, each object needs to communicate with other simulated objects. In ALICES, this is achieved in two ways, depending on the fact that the objects that need to communicate belongs or not to the same subsystem. Inside a subsystem, the objects are usually tightly coupled. This is expressed by graphical connections via special input/output points called "connectypes". Each connectype is defined to exchange a given information in a given direction (from or to the object) and the exchanged information is directly available within the object. Moreover, for a connection to be valid, the exchanged data through two connectypes must match in type (what is exchanged ?) and in direction (from an object to another object). This strongly reduces the communicating interface problems between two objects that have been created by two different persons.

For objects belonging to different subsystems, another communication facility is available. Each object has the ability to publish every quantity it computes under a public name. Through this public name, other objects can access the published quantity. Here again, to avoid mismatches, each public name is checked to be generated by only one object.

Secondly, the other major feature that simplifies the modeller's work is the possibility to express the ODEs describing an object state evolution in terms of secondary variables instead of the object state variables. All that has to be done is to express in a very natural way the relation

existing between each secondary variable and the object state variables it is function of. Moreover, the Jacobian matrix of the secondary variables relative to the state variables is computed automatically. This enables the object dynamics ODEs writing to be simpler compared to what it would be if it was expressed as state variable functions. As a result, the objects are created faster and are less error prone.

Once an object exists, its behaviour can be tested by creating a simple system made of this object connected to passive or active boundary objects. After it has succeeded its unitary tests, the object is ready to be assembled with other checked objects into subsystems. The object unitary test and subsystem realisations feature ALICES Level 2.

At level 2, the objects created at level 1 are graphically dragged onto a virtual working document and connected together via their connectypes. The object parameter values are also set. Here again, a special care has been taken to simplify the modelling work and to avoid errors since the parameter values can be read or computed from external databases. Level 2 is also responsible for giving a public name to quantities that need to be exported to or imported from other subsystems.

The subsystems are assembled together and run at level 3. The subsystem set can be the whole system; however, for test purposes, it can consist of only a subset of these subsystems.

Level 3 gives the user the ability to follow the evolution, textually or graphically, of every quantity used in the simulation. The users can also modify on the fly these quantities.

Level 3 has also been designed to allow external code integration with the simulator. It should be noted that compilation occurs only at level 1, in order to produce the software components we talk about in section 2. These components need then just to be dynamically linked with the simulator core at level 3 to get an executable simulator. That feature greatly speeds up all simulator modifications.

5. Technical background

ALICES was developed in the industry standard C++ language. The object oriented development method, originating from G. Booch's work, was chosen to guarantee coherence during the building and the successive evolution phases of the workshop. ALICES runs under the Solaris operation system on Sun SparcStation. The subsystems can be shared over different processors and/or over different workstations. Special care has been given to insure that ALICES is portable to other UNIX environments and to Windows NT.

6. Conclusions

ALICES has been designed in order to satisfy the following important points of view :

The point of view of the project: time and costs of simulator development and maintenance need to be minimised.

The point of view of the designer: modularity and open systems are required to insure easy interface and integration with already existing systems.

The point of view of the user: flexibility is essential, in order to adapt the software workshop to the size of the development; user friendliness and coherent environment are a must in order to facilitate the use of the software workshop.

As an open system, ALICES can also be used for building other types of simulators, such as design simulators.

The software workshop ALICES is currently used to develop the multifunctional simulators at TRACTEBEL.
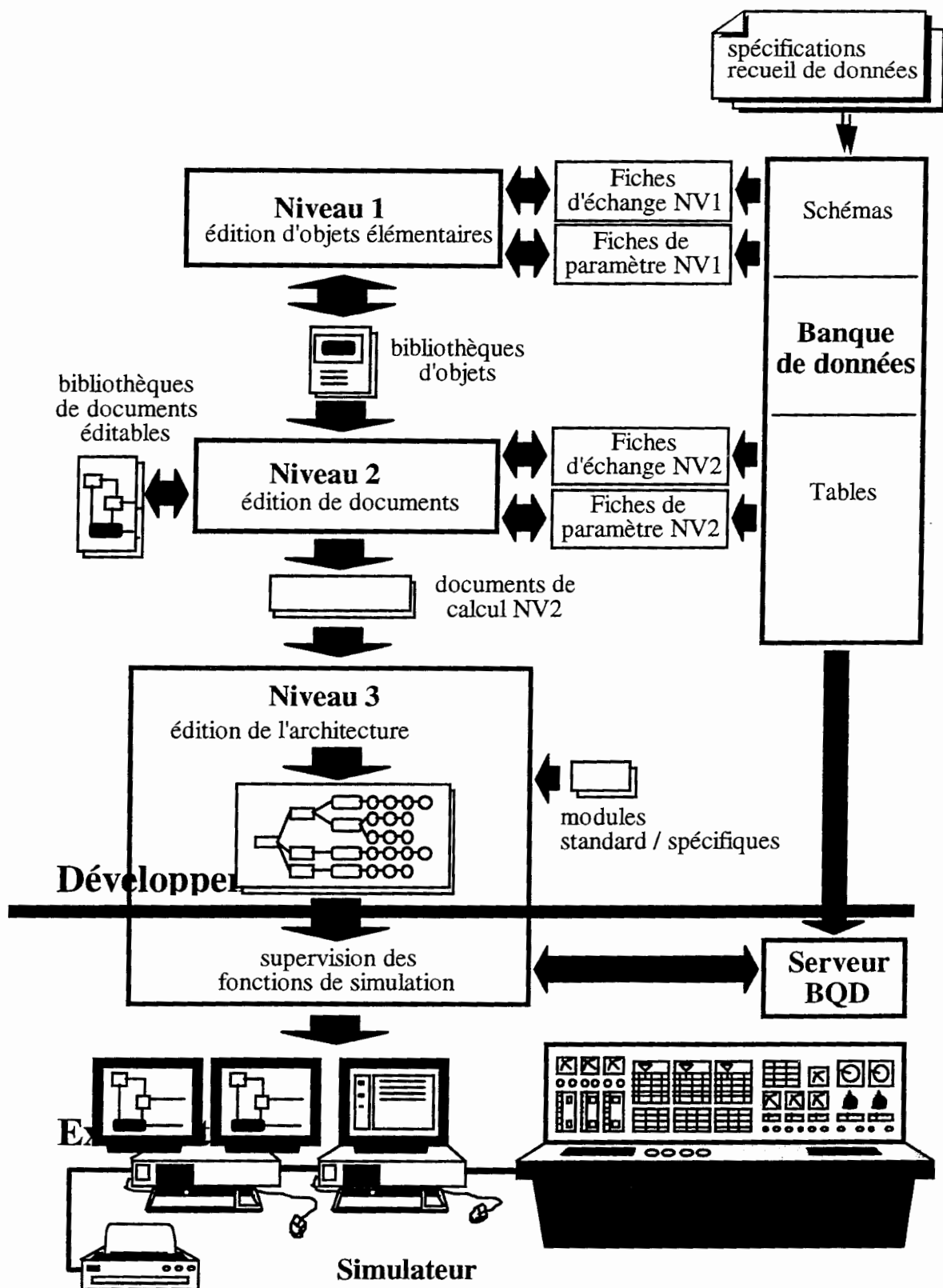
Figure 1 : the architecture of a simulator development with ALICES

VALIDATION OF ACCIDENT TRANSIENTS IN TRAINING SIMULATORS

Laurent BRUE, Benoit FALLON

CORYS T.E.S.S., 74 rue des Martyrs, 38027 GRENOBLE CEDEX 01, FRANCE

1. Presentation

Simulators have become of a major use in the training of operators in various domains, such as nuclear, thermal and diesel power plants operations.

For several years, CORYS has developed a complete software workshop in order to integrate all the systems modelled in simulators. Among these tools, DRAC is a thermal-hydraulic two-phase flow code used for modelling of normal and accidental processes.

As the reliability of the computing codes and models used in these simulators has a major importance in the training of operational incidents and, in particular, of accident transients, it is clear that the validation of the results on characteristic transients must be provided. But measurements recorded on site cannot be supplied for all accidental events. Comparative validation based on available plant data can only concern a specific range of operational transients and certain incidents, while the simulation of main accidents must be based on physical models validated and qualified by alternative means.

In order to estimate the accuracy of the results obtained with DRAC, comparisons with the "best estimate" code RELAP have been analysed. Works performed during the development of two simulators are presented in this paper.

2. DRAC : two-phase thermal hydraulic code

DRAC is a two-phase flow thermal hydraulic code developed and used in CORYS. For simulation purposes, the basic 5-equation model is structured and written to facilitate its integration into the simulator environment. The scope of simulation is required to cover all normal and accidental operations in the simulated systems.

The accidental sequences for breaks were examined in order to determine the various physical situations which may be encountered in them. The main points are given below :

- two-phase flows may be encountered in all the primary coolant system components. Vapour can be generated by the power given off in the core, by heat release from walls, or by self-vaporization in the depressurization phase,
- two-phase flows with subcooled liquid are encountered on activation of the safety injections,
- two-phase flows with superheated vapour are also encountered in the event of partial core uncovering and rod temperature excursion,
- mechanical non-equilibria between phases are encountered in all elements. The velocity deviations of the two phases must therefore be described. Particular attention is paid to the appearance of stratification in the horizontal legs and the countercurrent flow.
- operating conditions in single-phase and two-phase natural circulation are accounted for,
- the conditions known as "reflux boiling" are also described with the associated countercurrent flows,
- the critical flowrate at the break is calculated for upstream conditions ranging from subcooled liquid to superheated vapour, via all two-phase conditions.

The constitutive package includes an exhaustive set of closure relationships for :

- wall-fluid heat transfer : different heat transfer processes are modelled : forced convection to liquid, nucleate boiling, onset of critical heat flux, transition boiling, film boiling and forced convection to vapour. For critical heat flux a correlation with mass flow rate, pressure and void fraction is used, including different types of transitions (departure from nucleate boiling and dryout),
- interfacial heat transfer : these heat transfers are managed by heat fluxes between the interface and liquid and vapour respectively. Cases of superheated liquid or subcooled vapour are taken into account.
- velocity slip : an algebraic expression is used covering the different geometries and different flow patterns.

To calculate the core reflooding phase of large-break Loss Of Coolant Accident (LOCA) transients, a simplified approach to the complex rewetting phenomenon is adopted. The quench front velocity is calculated as a function of pressure, wall temperature, mass flux and liquid subcooling at the quench front.

Nodalization and resolution
The system is broken down into volumes, axial modules (pipes) and junctions which make up the various system elements. An element (separator, core, tube, etc,) can be made up of one or more modules depending on its complexity and on the simulation scope. Pumps and valves are modelled by junctions introducing a pressure difference as a function of flowrate.
The DRAC axial module has been developed to enable accurate representation of pipes with a large L/D ratio due to a more complete momentum equation.
Evolution of the main variables is achieved by resolution of a system of conservation equations with partial derivatives which describe the transfers between the various phases, volumes, pipes and junctions. The resolution of the non-linear algebraic system, which results from implicit time discretisation is done by Newton iterations. It is crucial to implement optimised matrix inversion methods, in order to guarantee real time computation speed.
The models need user specified parameters, which are accessible through a structured data base system.

Use
During the last years, DRAC has been used for the modelling of thermal-hydraulic systems in various simulators, such as :
-       VVER 1000/320 NPP simulators,
-       RBMK 1500 simulator,
-       250 MW thermal power plants.
Results and comparisons with RELAP5 calculations, concerning PWR-600 and VVER 440-213 power plants, are presented hereafter.

3.   KRSKO validation - Simulation of Loss of Coolant Accident

The DRAC software was used for the modelling of the Reactor Coolant System (RCS), Steam Generators and Steam Lines up to the Steam Header in a multifunctional simulator developed by CORYS for KRSKO Nuclear Power Plant. This 2 loop PWR-600 type plant is situated in Slovenia. The scope of simulation of the simulator extends from cold shutdown with the primary circuit open (midloop operation) to nominal state of the power plant. It includes all important transients which may occur on the systems simulated by DRAC: break (up to 8 inches) on any hot or cold leg of the primary circuit, steam generator tube rupture (up to 3 tubes), steam line rupture (inside or outside containment).
Before integration of the DRAC module to the simulator, unit tests were performed in order to evaluate the accuracy of the behaviour of the DRAC model alone. Results of simulation concerning LOCA could be compared to the ones obtained with RELAP5.
RELAP5 has been adopted by the Faculty of Electrical Engineering and computing in Zagreb and a model has been developed in order to analyse accidental transients in KRSKO NPP. The model has been qualified on the steady-state level.
Results of the simulation of an 8-inch LOCA on a cold leg are presented hereafter.
The following table presents the values of main parameters at the nominal state of the plant :

Pressurizer pressure	15.5 MPa
Steam Generators pressure	5.7 MPa
Hot Leg Temperature	598 K
Cold Leg Temperature	560 K
Core thermal power	1876 MW

Both in the DRAC and RELAP5 models, scenarios are computed to take into account the main events; according to the evolution of the main parameters, scram, turbine trip (causing closure of isolation valves on steam lines), safety injection signals, feedwater isolation of steam generators or primary pumps trip are likely to occur.

The main differences between DRAC and RELAP5 models are the following ones :

-        Nodalization is much more precise in RELAP5 than in DRAC : 230 nodes instead of 46 (DRAC nodalization is linked to the real time constraint).

-        Containment is not included in DRAC, due to the fact than it is simulated by another module in the simulator. For unit tests, containment pressure is defined as a correlation from primary circuit pressure.

-        DRAC includes the Steam Header; RELAP5 does not.
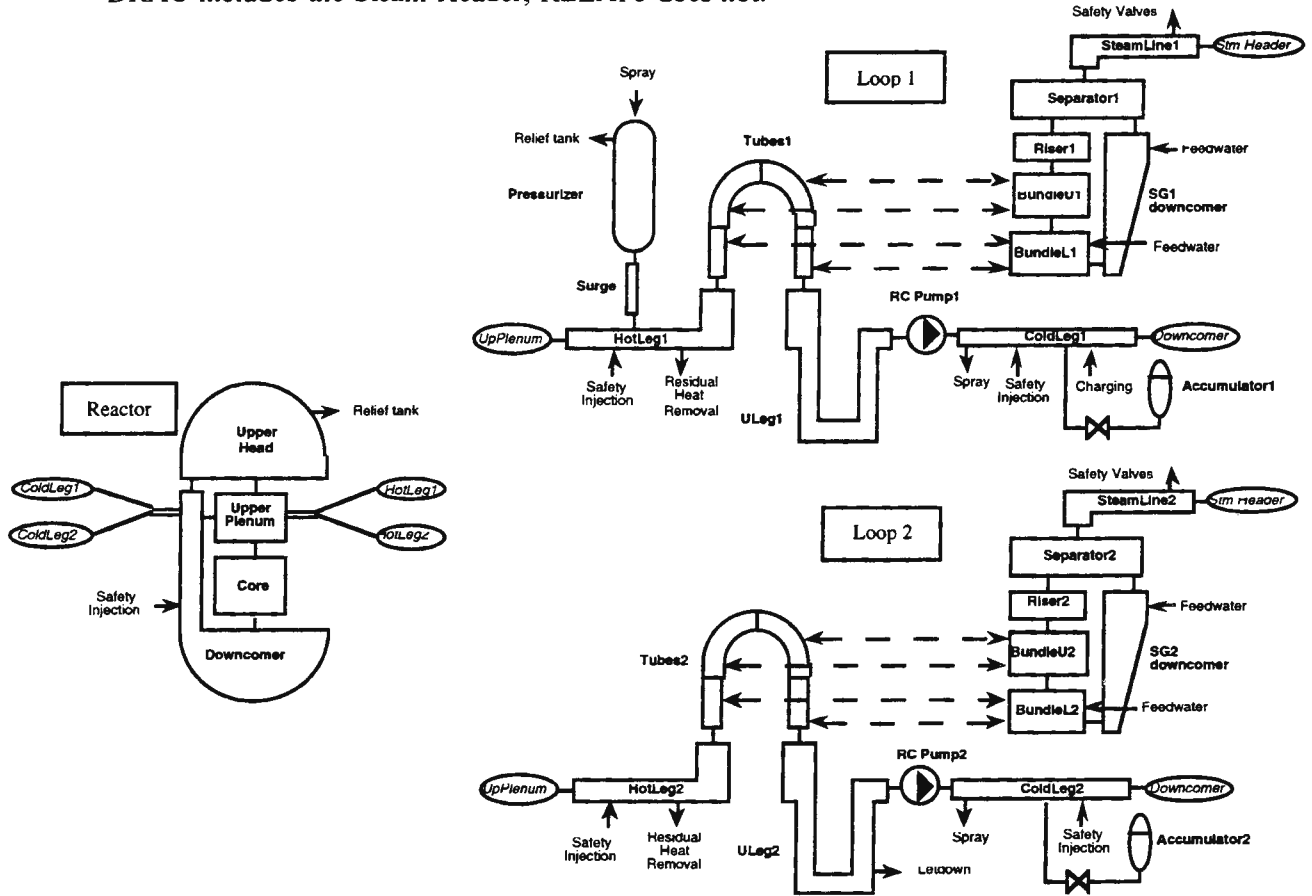
Figure 3.1 : overview of the systems simulated with DRAC and RELAP5

Figures 3.2 to 3.4 display the evolution of parameters after occurrence of an 8-inch break at time t = 0 s (scales are exactely the same on both sets of curves).
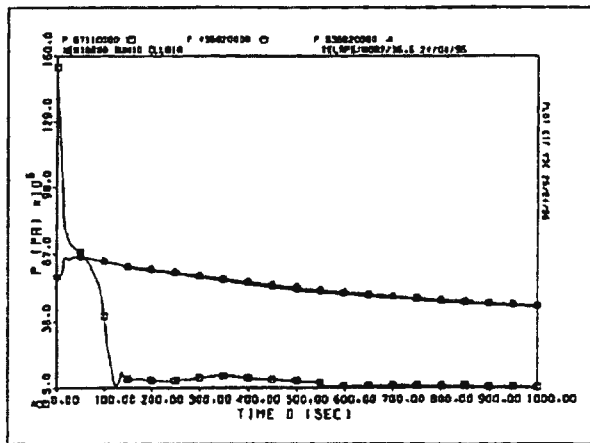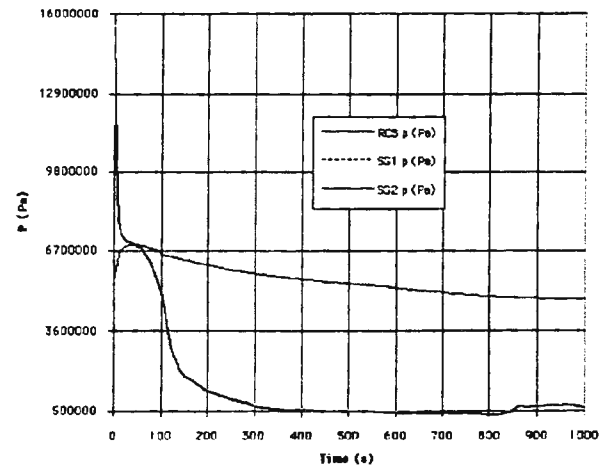
Figure 3.2 : primary circuit and SGs pressures computed by RELAP5 (left [1]) and DRAC (right)
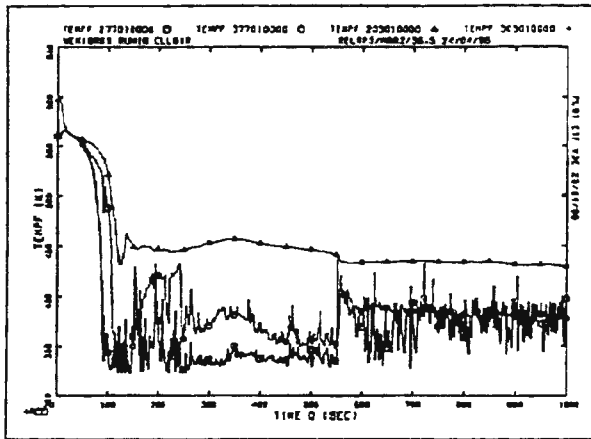
Figure 65   CLLB1A - Hot and Cold Leg Fluid Temperatures
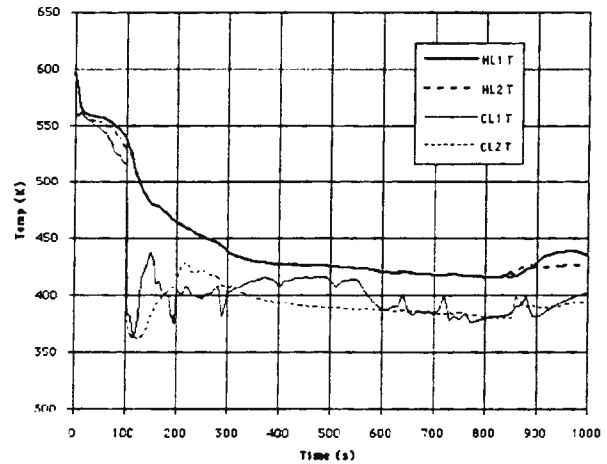
Figure 3.3 : hot and cold leg temperatures computed by RELAP5 (left [1]) and DRAC (right)
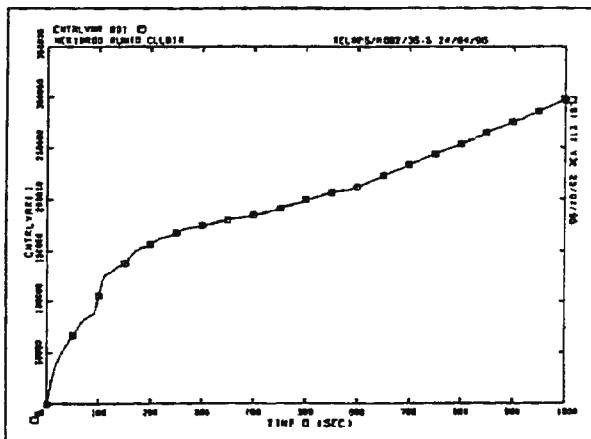
Figure 66   CLLB1A - Integrated Break Mass Flow (kg)
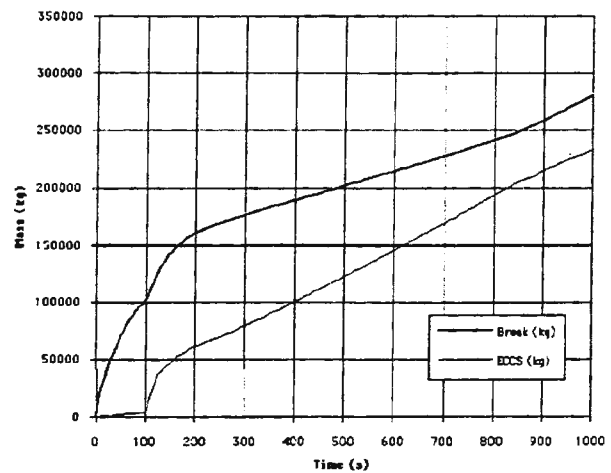
Figure 3.4 : integrated break mass flowrate computed by RELAP5 (left [1]) and DRAC (right)

Break opening induces a fast depressurisation of the primary system. Start of the high pressure safety injection is not sufficient to compensate the break flowrate and both accumulators and low pressure safety injection are prompted. However pressurizer becomes empty in few seconds.

Safety injection water temperature being low (accumulators temperature is 50°C) both hot and cold legs temperatures decrease quickly after primary pumps trip and actuation of the safety injection system.

The size of the LOCA break is large enough to remove residual heat from the core ; therefore heat power exchanged with the secondary system through the steam generators decreases very quickly ; increase of pressure in the secondary side of the steam generators after occurrence of feedwater trip and turbine trip is not sufficient to make relief valves on steam lines open.

It is remarkable to notice that DRAC and RELAP5 simulations lead to the same behaviour of the main parameters. Time characteristics of the events are the same in both simulations. The main difference concerns primary pressure decrease after accumulator injection, which is faster in RELAP5 (from 100 s to 300 s).

It must be noticed that the evolution of the levels in steam generators are not displayed here : the definition and the working range of the sensors being different between RELAP5 computation and requirements for the simulator, comparison of the results is not justified. But both computations present a fast drop of level when the break occurs ; the subsequent long term increase of the level is caused by the running of the auxiliary feedwater system.

This comparison with the results of the qualified "best estimate" code RELAP5 is totally acceptable for the simulation code DRAC.

Integration into the simulator

The integration of the model into the simulator enables to connect it with other modules and to have a better definition of the following elements :

-        the models of external thermal-hydraulic systems, such as safety injection, main and auxiliary feedwater, main steam system and containment are no more defined as boundary conditions or correlations, but as models using physical equations for mass and energy balances.

-        the interlock and control system integrates additional elements such as heaters activation or separate controls of each device of the external systems.

Moreover, didactical sensors and images are displayed in the simulator and enable to analyse physical phenomena which occur during the simulation.

Figure 3.5 presents a didactical image of the steam generators on which measurements of several physical parameters enable to follow the thermal-hydraulic evolution during the transient phases.

Such images are of prime importance in the training process since they permit to visualise the phenomena and focus the training on the understanding of the underlying physics of the processes.
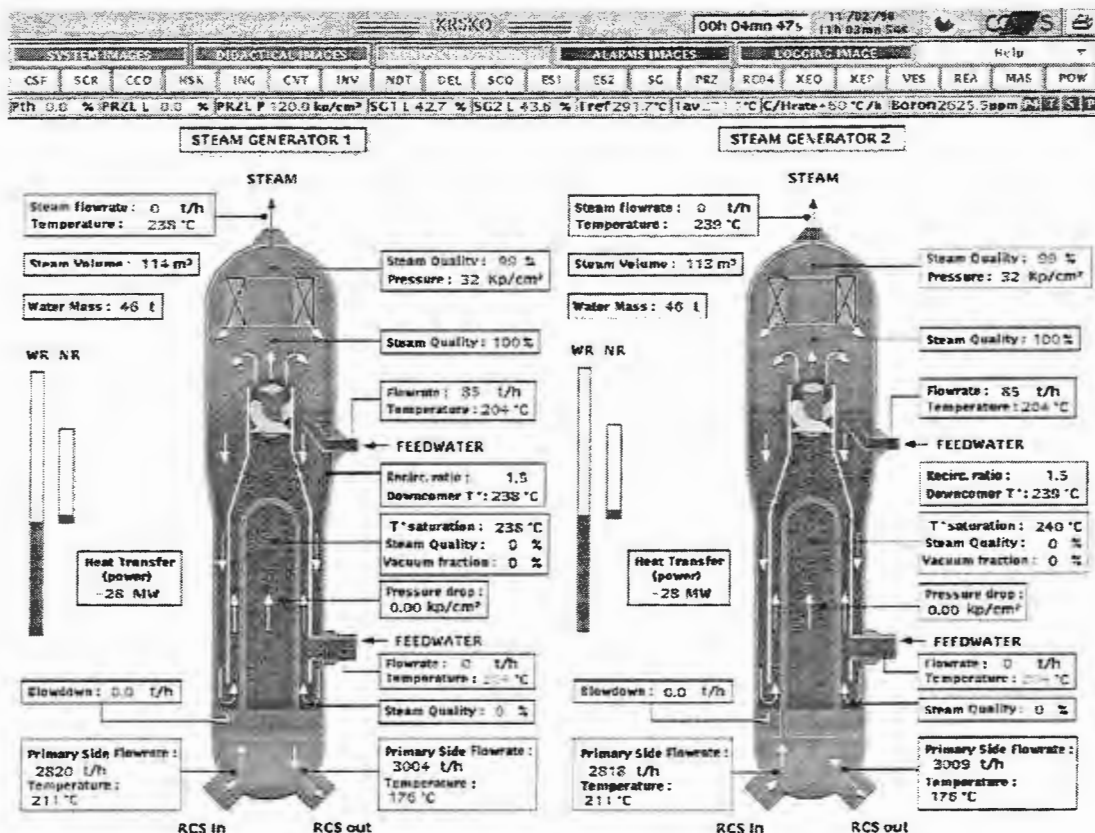
Figure 3.5 : didactical image of the steam generators

4.   EVVEREST validation - Primary header leak accident

The EVVEREST project includes 8 extended-scope multifunctional simulators of 6-loop VVER-440 type Nuclear Power Plants of 3 basic sub-types :

            V 179 : Novovoronesh Nuclear Power Plant,
            V 230 : Bohunice V1, Kola 2 and Kozloduy Nuclear Power Plants,
            V 213 : Dukovany, Bohunice V2, Kola 4 and Rovno Nuclear Power Plants.

The DRAC code is used to simulate the Nuclear Steam Supply System (NSSS) of these VVER-440 units, including the reactor vessel, the 6 loops and the primary and secondary sides of the steam generators up to the steam headers.

The DRAC EVVEREST model was first qualified on the 100 % steady state with regard to the plant physical parameters (geometry, thermo-hydraulic parameters, control set-points,...). Starting from this state, the following set of validation transients was computed and analysed : shutdown of the 6 reactor coolant pumps, pressurizer safety valve stuck open, primary header leak, small- and intermediate-break LOCAs (diameter from 32 mm to 250 mm), large break LOCA, loss of feedwater, steam line break, steam generator pipe rupture.

The nodalization scheme was built on a compromise between physical accuracy of the model and the simulator real time constraint; a total number of 158 nodes is divided as follows :

Element	Number of nodes	Element	Number of nodes
Core	35	Pressurizer	4
Reactor Vessel	21	6 SG primary	48
6 loops	18	6 SG secondary	32

Hereafter, validation results are discussed for the case of the steam generator cold collector leak accident and compared to the available RELAP5 calculation ; the break (equivalent diameter 107 mm) is activated on the cold collector of the steam generator on loop 1.

The comparison of calculation results is illustrated in the following figures (from 4.1 to 4.4).
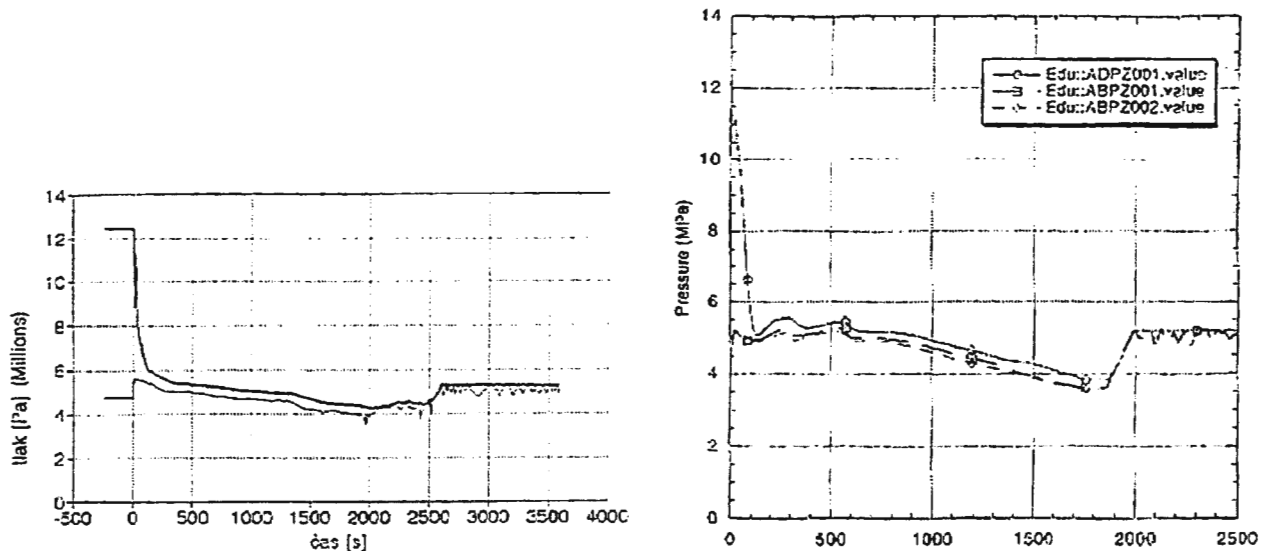
Figure 4.1 : primary circuit and SGs pressures computed by RELAP5 (left [2]) and DRAC (right)

To be able to reproduce with precision the initial and boundary conditions of the reference calculation, a stand-alone DRAC version was used. The scenario was in accordance with the RELAP 5 calculation as follows:

    thermal core power: SCRAM followed by a standard decay heat curve,
    shutdown of the 6 RCP at 0 s,
    turbine trip after 5 s,
    stop of feedwater pumps at 0 s,
    start-up of injection systems (hydro-accumulators and high pressure).

Dynamic equipments like heaters, pressurizer safety valves, steam generator safety valves, steam dumps... are included in the model. No models for injection systems have been developed within these stand-alone calculations. Boundary conditions for mass and enthalpy flowrates were imposed (in the complete simulator, these injection systems are modelled with a dedicated 3 equations code). Simulation time step is kept constant and equal to 250 ms during the whole calculation.
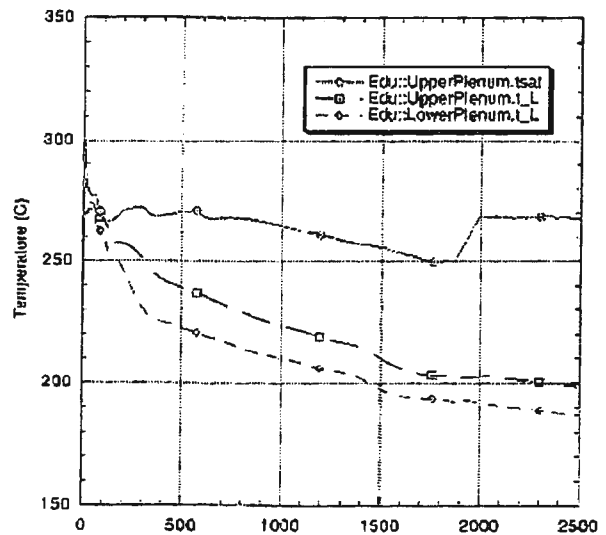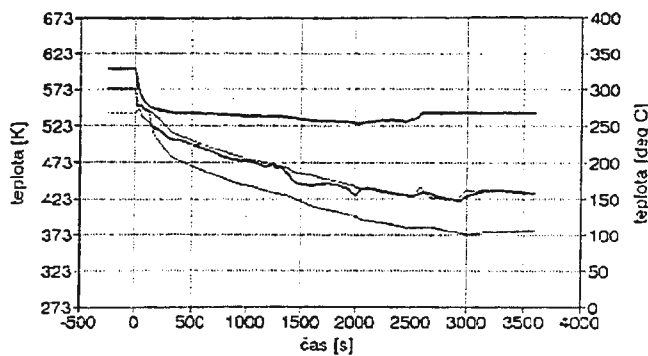
Figure 4.2 : T° in lower and upper plenums computed by RELAP5 (left [2]) and DRAC (right)
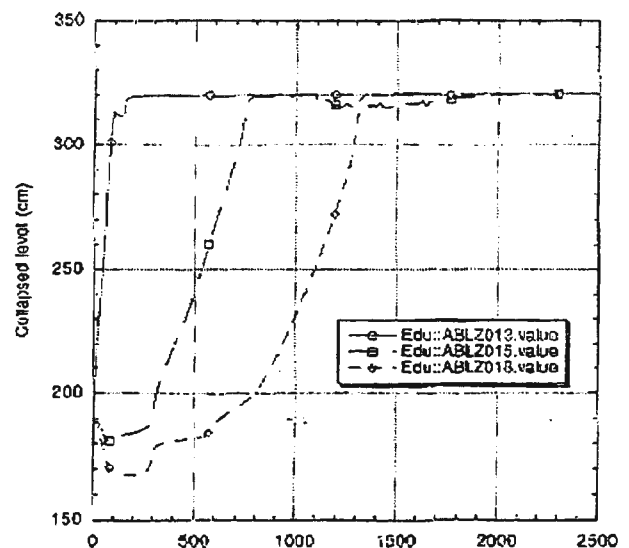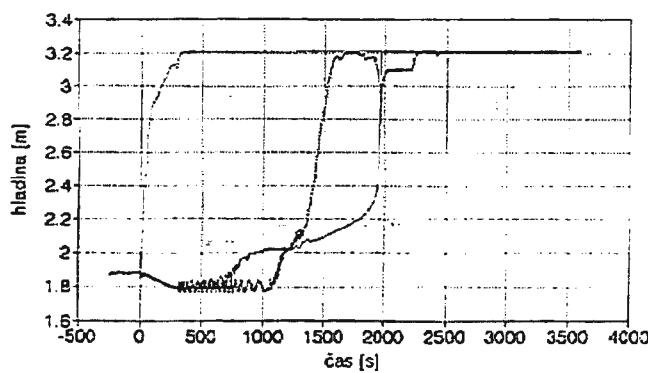
Figure 4.3 : level in steam generators computed by RELAP5 (left [2]) and DRAC (right)
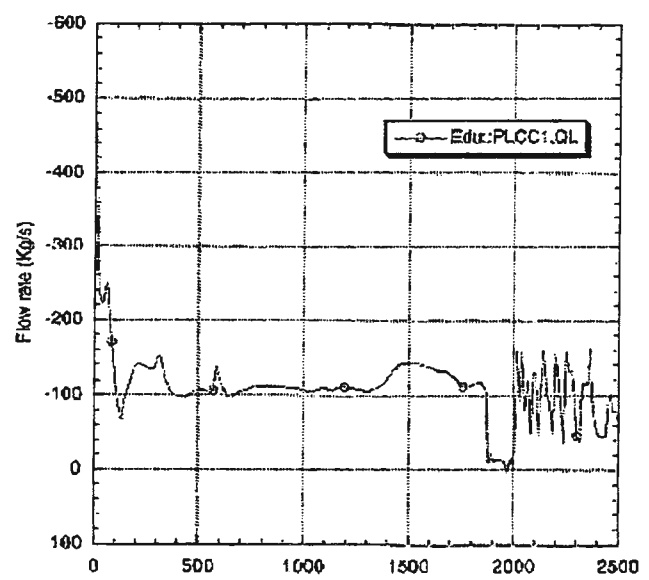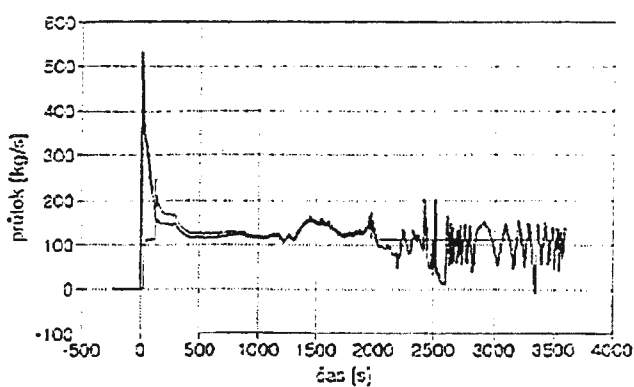
Figure 4.4 : break flowrate computed by RELAP5 (left [2]) and DRAC (right)

162

Phase 1: Primary system depressurization (0-150s)

Break opening induces a rapid primary pressure decrease until the pressure balance between primary and secondary side is achieved. On the secondary side, the break flow from the damaged collector leads to rapid water fill-up of SG1. Water starts to fill up the main steam header.

Start of the high pressure safety injection is not sufficient to compensate break flow rate. Thus, the pressurizer level rapidly decreases and the pressurizer voids at about 100 s.

In the non-affected SGs, turbine and feedwater trip cause a pressure increase until opening of the atmospheric dump valves at 5.3 MPa. SG levels slightly decrease. After that, the secondary pressure is controlled at 5.0 MPa.

Phase 2: Cooling effect on primary side (150-1900 s)

Both high pressure and hydro-accumulators injections have a cooling effect on the primary side. Core temperature decreases progressively from about 270 °C to less than 150 °C. This induces a slight primary pressure decrease. Break flow remains constant at about 100 kg/s.

The secondary circuit progressively fills up from the damaged SG, until the main steam headers are completely full of water (at about 2500 s).

Phase 3: Pressurizer refill (1900-2500 s)

The secondary side is full of water. High pressure injection induces a pressure increase on primary side. Pressure increases rapidly until the opening of the atmospheric SG dump valves at 5.3 MPa. Pressurizer starts to fill up and core temperature continues to decrease due to high pressure safety injections.

Conclusions

DRAC stand-alone predictions are in excellent agreement with the RELAP calculation. All main physical behaviours are well reproduced.

The observed differences with the reference RELAP calculation can be explained by:

- simplified nodalisation in the DRAC calculation
- small differences in the boundary conditions due to their incomplete description in the reference calculation
- in the reference run, reactivity phenomena (which increases primary pressure) are taken into account at about 2200 seconds. The stand-alone version of DRAC cannot take into account reactivity feedback effects.

For training simulators, real time simulation is essential. DRAC convergence during the rather violent transient is fully acceptable (7 per time step iterations during the first 100 seconds and 2 iterations per time step in the remaining part of the transient). Real time performance on the simulator computer platform (a 6-CPU SUN E4000 server) is achieved.

A total of 9 validation tests with respect to reference RELAP calculation was carried out with satisfactory results. Apart from that, the physical behaviour in all normal operational situations is checked and approved by the NPP experts during the validation and acceptance phase of the simulator.

5. Conclusion

The presented results show that DRAC models provide accurate simulation of accidental transients in thermal hydraulic systems of the PWR and VVER-440 reactor systems: comparison with results obtained with the "best estimate" code RELAP5 demonstrates that DRAC is a reliable tool for the simulation of accident transients with a precision fully acceptable for training simulators.

The validation on specific transients enables to integrate both KRSKO and EVVEREST simulators with confidence in the behaviour of the models on major accidental events proposed on these simulators.

The results show that the real time constraint is compatible with the use of high fidelity models in a 5-equation thermal-hydraulic code.

Further work is necessary to achieve systematic validation of the code DRAC for the full range of accident transients.

References

[1]   I Basic - Summary of NEK Reactor Coolant System Leaks (0.5"; 1";4" and 8") Best Estimate ELAP5/mod2 Analysis ; Report number NEK ESD TR 03/96, Rev.0, May 1996

[2]   J. Krhounkova   -   Bezpecnostni analysa havarie s uvolnemin vika kolektoru PG JE s VVER440/213 (1995)

Tracking Simulator Model of Nuclear Power Plant

Pasi Laakso, Finntech Ltd.
Tekniikantie 12
02150 Espoo
Pasi.Laakso@sah.vtt.fi

1   INTRODUCTION

Main purpose of this report is to present shortly the contents of the Master's Thesis[1] where one possible concept of a tracking simulator was considered. In the thesis a tracking simulator was developed based on controlling certain state variables chosen by using expert knowledge on nuclear safety analysis. Thesis was supported by the IVO Power Engineering Ltd. and Technical Research Centre of Finland.

In this thesis the tracking simulator was a simulation model, driven alongside a 'real plant' in real time and the state of which was kept as close as possible to the state of the 'real plant'. In the thesis the 'real plant' was also a simulation model.

The results of tracking can be used to improve the quality of measurements and therefore to enhance the knowledge of the real state of the plant. A special requirement for the realised tracking simulator was that it should be able to generate an accurate initial state for a simulator, which predicts the behaviour of the plant in the future.

The basic principle in setting up the tracking simulator was to choose a small number of 'real plant' variables which were assumed to define the thermal hydraulic state of the plant. These variables were called the main variables. The values of the main variables were the target, towards which the values of the corresponding variables of the tracking simulator were driven. To achieve this goal the main variables of the tracking simulator were either controlled directly or guided by modifying the simulation model parameters affecting them. Furthermore the control actions made by the operators and plant automation system were taken into account when implementing the tracking simulator.

The underlying idea of setting up a test bench was the fact that many industrial processes can be modelled very accurately and in real time using present day simulation tools. Also, the automation systems can be connected directly to the simulation models. In this test case both the tracking simulator and 'real plant' model were realised by using Apros simulation software, which offers the necessary models and communication facilities. The modelled process was the primary side of a VVER-440 nuclear power plant. The 'real plant' model was based on a detailed six-equation thermal hydraulic model, while the tracking simulator had a simpler homogeneous thermal hydraulic model with heavily simplified nodalization. This set-up was considered to represent closely enough the situation between a real process and a simulation model. Automation system was identical in both of the models except for

---

[1] Pasi Laakso, 1997 "Ydinvoimalaitoksen Seurantasimulaattori", Master's Thesis (in Finnish). Helsinki University of Technology.

the changes made to implement the tracking features. The predicting simulator was identical to the tracking simulator with tracking features switched off.

In this test case the proposed concept was found out to work and therefore it forms a basis for the development of the APROS based tracking simulator in the future.

2 TRACKING SIMULATOR

The role of a tracking simulator is to calculate

- initial state for the predicting simulator
- unmeasurable state variables
- measurable state variables, which are compared to the corresponding variables of the real plant.

The realised tracking simulator was mainly meant to calculate the initial state for a predicting simulator but it can be used for all of the above purposes. Tracking simulator can also be used to determine how well the simulated process is modelled by the simulator program. The tasks and implementation of a tracking simulator are close to the tasks and implementation of a signal validation and state identification systems.
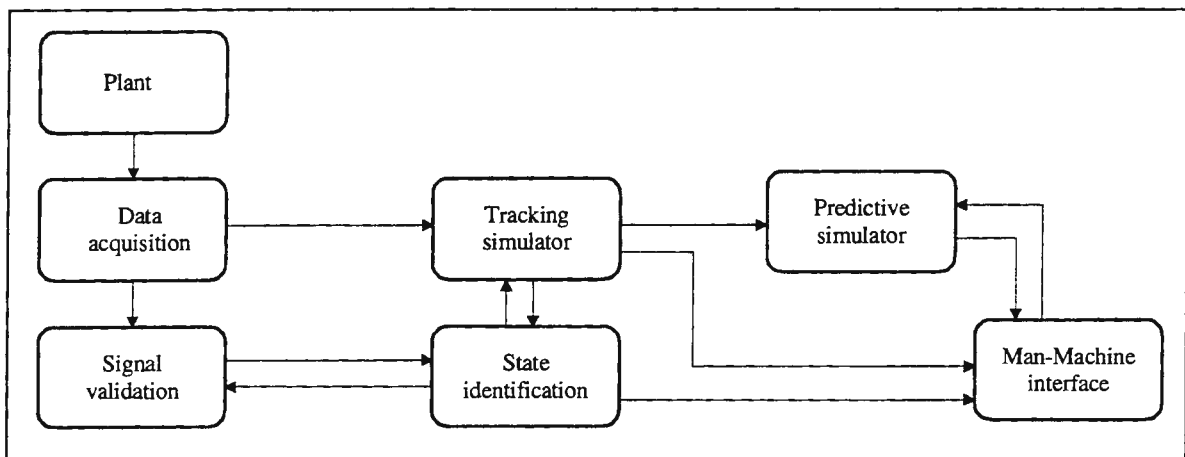
Figure 1 Data flows in a computerised plant management system.

Tracking simulator is supposed to be one part of the computerised management of the process plants, which helps the plant personnel to monitor and control the state of the plant. Figure 1 illustrates one possibility of the interactions between a tracking simulator and other parts of the system. The figure is based on the research made in the CAMS project in Halden, Norway[2]. The data acquisition system transfers measurements from the main process and from the automation system to the tracking simulator which calculates its state by using these measurements. Tracking simulator can also use 'improved' data got from the state identification system. The state calculated in the tracking simulator is compared to real plant state in the state identification system and forwarded, if needed, to the predictive simulator to be used as an initial state. The unmeasurable state variables, which have been

---

[2] Fantoni P., Iguchi Y., Meyer G., Sörenssen A., Van Dyck C. "CAMS prototype extension: integration of Data Acquisition, Signal Validation, Tracking Simulator, Predictive Simulator, State Identification and Probabilistic Safety Assessment" OECD Halden Reactor Project, HWR-440. 44 s.

calculated in the tracking simulator can be presented to the operators through the monitors of the operator station.

In this work the tracking had two possible operating modes. In the catch-up mode the state of the tracking simulator was controlled strongly to drive its state close to the state of the 'real plant'. In the tracking mode the tracking simulator was already close to the state of the 'real plant' and the purpose of tracking was to keep this situation. The use of a predictive simulator could also be called one mode, because the predicting simulator model is based on the tracking simulator model and thus the accuracy of the prediction is closely related to the accuracy of the tracking simulator model.

3 MAIN VARIABLES AND TRACKING STRATEGIES

The most important phase in realising the tracking simulator was to choose the main variables, which adequately illustrate the state of the plant. When these variables had been chosen next step was to control the calculation of the tracking simulator so that these variables in the tracking simulator closely enough follow those of the 'real plant'. The task is challenging when the used tracking simulator model has thousands of parameters and variables which affect to calculation. Furthermore the interactions between the main variables must taken into account. The chosen main variables are presented in the Table 1. The main variables were obtained by using the expertise gained in many years of nuclear power plant safety analysis..

Table 1 The main variables of the tracking simulator and the tracking principles used with these variables. In all of the tracked variables the set point of the control system comes from the plant and the measurement from the tracking simulator. Control actions affect the tracking simulator.

Main Variable	Used tracking principles or features in the tracking simulator
Pressure of the secondary circuit	The positions of the turbine valves are controlled. The power controller of the tracking simulator has been by-passed.
Mass balance of the secondary circuit	There is no special tracking feature for this purpose. Modelled liquid level controllers of the steam generators control the mass balance.
Temperature of the cold leg	The temperature difference over the steam generators is controlled and some extra heat is either extracted from or added to primary circuit.
Flow state of the primary circuit	The flow parameters in the primary circuit are controlled.
Reactor power	Position of the control rods of the reactor are controlled. The secondary pressure controller has been by-passed.
Mass balance of the primary circuit	The liquid level of the presssurizer is brought directly from the plant to the tracking simulator.
Pressure of the primary circuit	In the upper part of a pressurizer is a pipe and boundary condition, which controls the pressure in the head of the pressurizer.

The state of the automation system of the plant can not be used directly in the tracking simulator, because of the fact that the models of the regulated units are not identical. For example, when the output of the controller in the plant orders a valve to be in some position and the corresponding valve in the tracking simulator is set to the same position

the flow through the valves might differ. The same applies in lesser respects to regulating units that have only few allowable states, in which they spend most of the time. At least the amount of work needed to tune the parameters of regulated units is large. This means that the automation system of the plant must be modelled into the tracking simulator and the set points and automation system interlockings had to be transferred from the plant to the tracking simulator.

4 SIMULATION RESULTS

The functions of the tracking simulator were tested by making simulation runs where the tracking simulator was set to catch-up, track or predict the behaviour of the 'real plant'. Four of these test cases were presented in the thesis and the results will be reviewed briefly in this chapter and in appendices 1 to 4, where some curves of the simulation results are presented.

In the first case (Appendix 1) the initial state of the tracking simulator was far from the state of the plant. The plant was operating at 75 % power level (Figure 2) while the tracking simulator was at full power. This test gives an idea, how the tracking simulator behaves during the catch-up mode. In the test the tracking simulator was run as fast as possible and the main variables were controlled strongly. For example the heat extraction (Figure 3) from the primary circuit is at its largest at 40 seconds (32 MW). The whole catch-up phase was quickly over. Within 500 seconds the tracking simulator had reached the state of the plant and stabilised. Even if the catch-up mode was quite fast it could be accelerated. The reactor inlet temperature (Figure 4) shows that the tracking features affecting to the cold leg temperature could be improved. At first the temperature rises and just at 30 s it starts to drop. The main reason for the imperfect behaviour is the control of the temperature difference over the steam generators (Figure 5), which decreases in the beginning the efficiency of the heat transfer between the primary and the secondary circuits (Figure 6).

In the second case (Appendix 2) the situation generated in the first case was continued. A transient was made in the plant, which the tracking simulator was set to follow. The transient was a change in the set point of the power produced. The set point was changed from 75 % to full power (Figure 9). The transient begun at the moment 0 seconds, when the set point change was made to the automation system of the plant. The plant power was controlled by the turbine valves (Figure 7), which started to open. This caused the pressure of the secondary circuit to drop (Figure 8). The secondary pressure of the plant was controlled by the control rods of the reactor. Therefore the pressure drop caused the reactor power to rise gradually to enable full power. The tracking simulator managed quite well. The main variables of the tracking simulator followed closely the state of the plant. Some differences can be noticed, but they are relatively small. For example, the reactor inlet temperature in Figure 10 is tracked closely. The largest difference between the plant and the tracking simulator is not more than 0.3 °C.

In the third case (Appendix 3) the situation generated in the second case was again continued and the prediction capability of the tracking simulator was studied. The predictive simulator was given the snapfile generated by the tracking simulator at 0 seconds. The transfer from tracking to predicting was successful and took place without a large transient, which might have been caused by the fact that tracking features were turned off at the same time. Now the prediction simulator tries to follow the plant, where one of

the reactor coolant pumps stops at 50 s. In Figure 11 the effect of this can be seen in a flow through reactor. The behaviour of the predictive simulator is compared to the behaviour of the plant. Furthermore the behaviour of the tracking simulator, which follows the plant during the transient is added to the figures and compared to other curves. The curves presented in Appendix 3 reveal that the tracking/predicting simulator model was quite inaccurate and by using the tracking features it was possible to keep the state of the simulator closer to that of the plant than without using them. The variables, where the former can be checked are the reactor power (Figure 12), secondary pressure (Figure 13) and liquid level of the pressurizer (Figure 14).

Fourth simulation run (Appendix 4) is an example of the situation, where the tracking simulator does not know the real state of the plant and thus the tracking features control forcefully the state of the tracking simulator. The case simulated was a small break loss of coolant accident (SBLOCA). Some conclusions could be made based on the control actions of the tracking simulator. For example the mass flow through the leak was close to the steam flow through the pipe, which controlled the pressure of the pressurizer (Figure 15). The liquid level of the pressurizer (Figure 16), primary pressure (Figure 17) and mass balance of the primary circuit (Figure 18) are presented to give an idea of the development speed of the accident. These figures also reveal the boundary condition-like behaviour of the pressurizer because of tracking features.

5   CONCLUSIONS

In this summary report one tracking simulator concept and the simulation results based on this concept have been presented. The chosen method of implementation was found out to work and therefore to give a basis for further development, which might consist of:

- accelerating the simulation speed of the predictive simulator and therefore the speed of tracking simulator in catch-up mode.
- the reconsideration of the implementation of the tracking features.
- testing the operation of tracking simulator in cases where some measurements are faulty or measurements are disturbed by noise.

The main problem of the reported concept is that the more the calculation process is affected on, the better the result of tracking is, however, the results available from the tracking simulation lose value, when a comparison to a real process is made, and the transfer from tracking to prediction comes more difficult. Ideally, the tracking simulator model would be accurate enough to enable tracking so that only a part of the automation system state has to be transfered from the plant to the tracking simulator and the tracking features affect only a little on the progress of calculation in normal operation conditions.

The method used in the catch-up mode can be utilised by most users of Apros. The initialisation of the large simulation model is a time-consuming task, which can be made easier by using the tracking features presented.
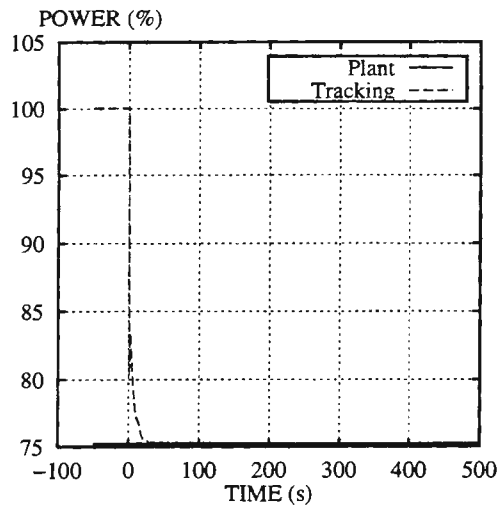
POWER (%)

Figure 2 Relative power of the reactor (catch up mode).

POWER (MW)
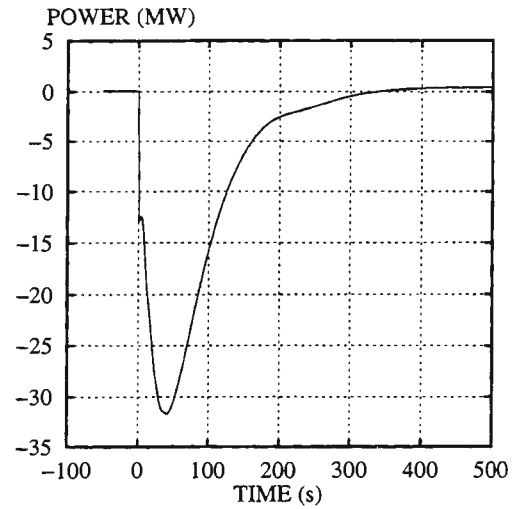
Figure 3 Heat extraction from or add to primary circuit (catch up mode).
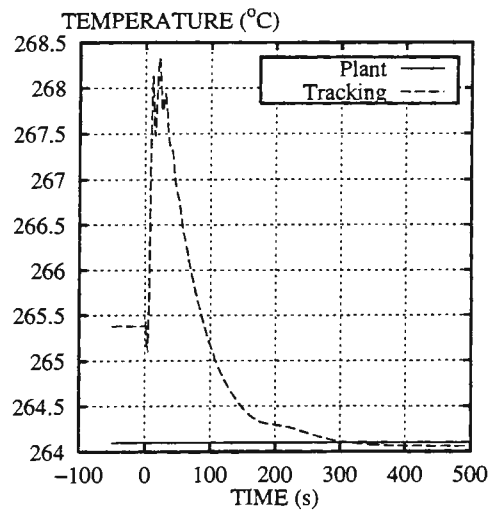
TEMPERATURE (°C)

Figure 4 Reactor inlet Temperature (catch up mode).

TEMEPERATURE DIFFERENCE (°C)
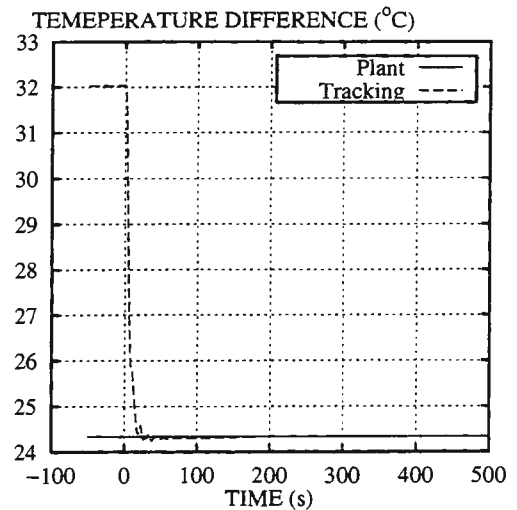
Figure 5 Temperature difference over one steam generator (catch up mode).
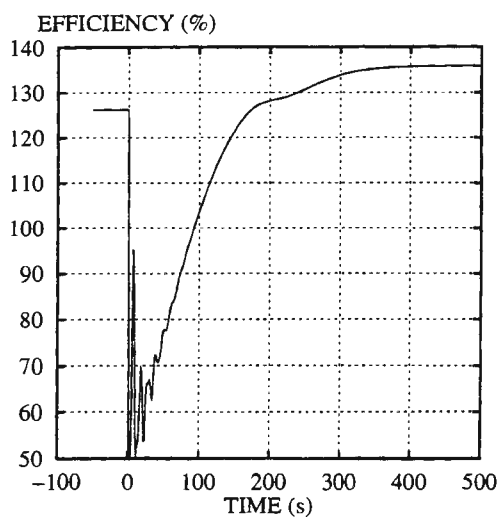
EFFICIENCY (%)

Figure 6 The efficiency of the heat transfer in one steam generator (catch up mode).
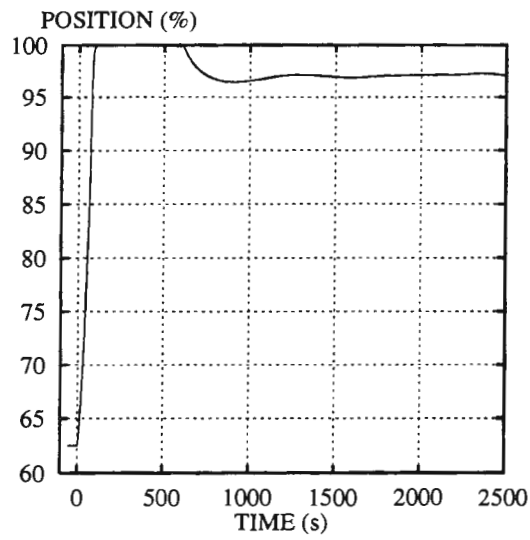
Figure 7 Position of one turbine valve (tracking mode).
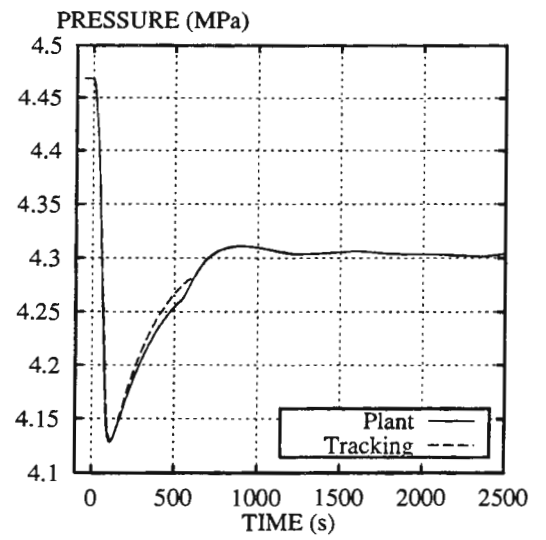
Figure 8 Pressure of the secondary circuit (tracking mode).
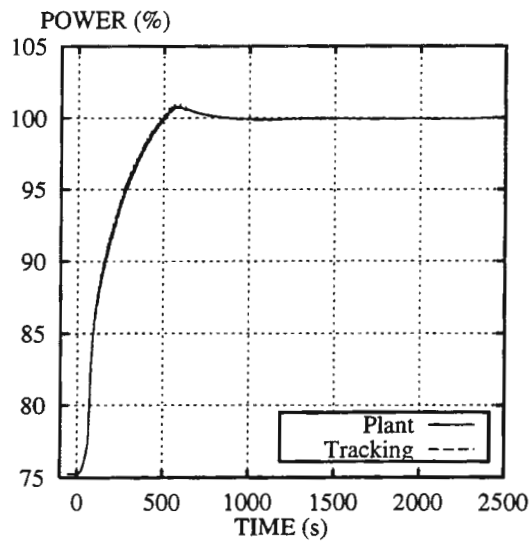
Figure 9 Relative power of the reactor (tracking mode).
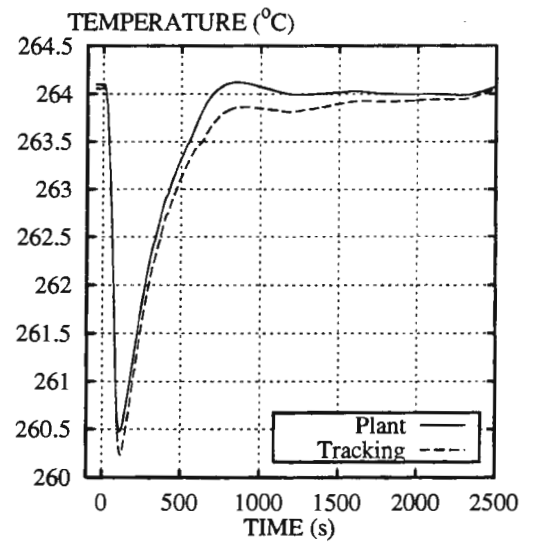
Figure 10 Reactor inlet temperature (tracking mode).
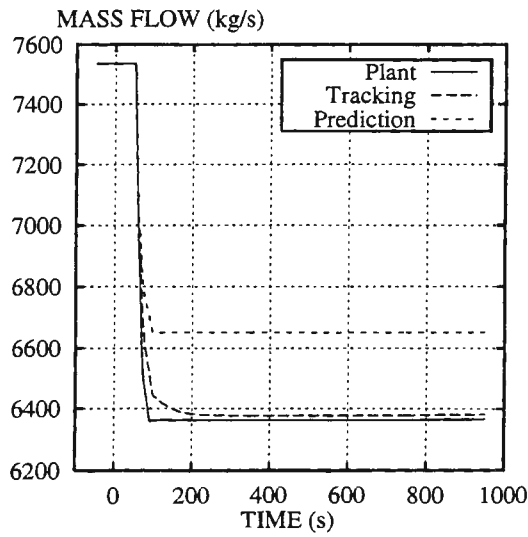
MASS FLOW (kg/s)

Figure 11 Mass flow through the reactor (prediction).
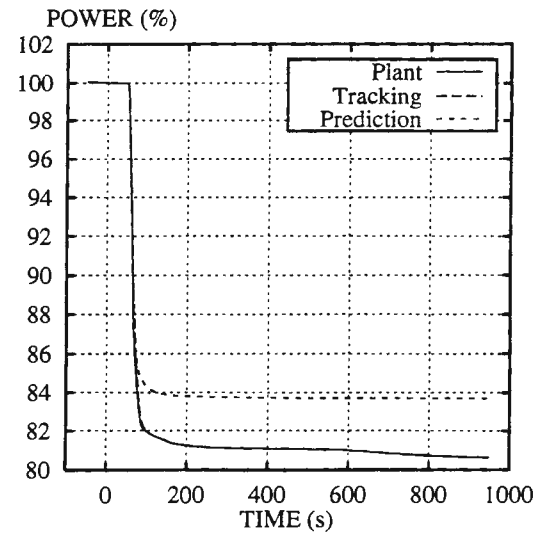
POWER (%)

Figure 12 Relative power of the reactor (prediction).
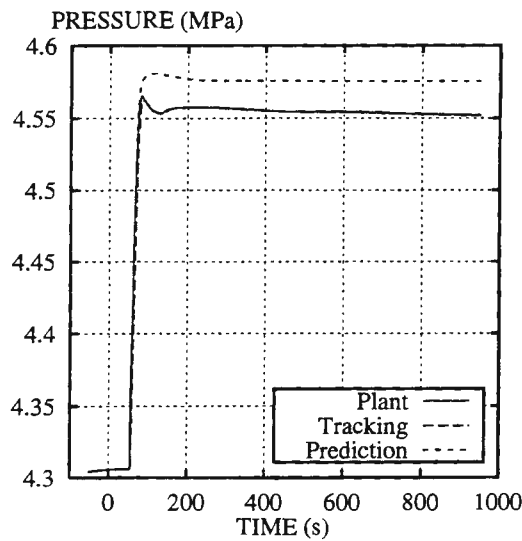
PRESSURE (MPa)

Figure 13 Pressure of the secondary circuit (prediction).
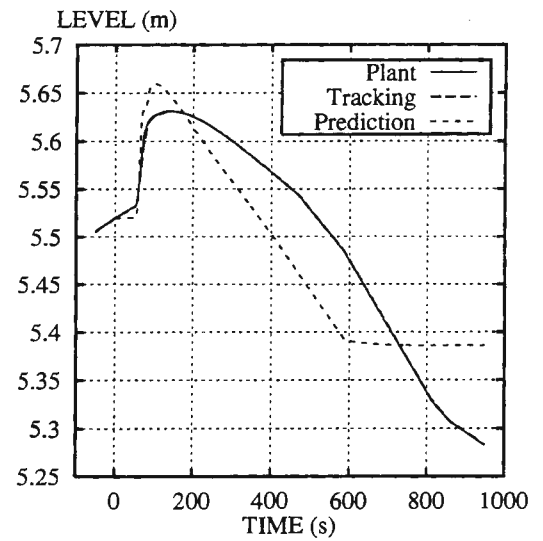
LEVEL (m)

Figure 14 Liquid level of the pressurizer (prediction).

MASS FLOW (kg/s)
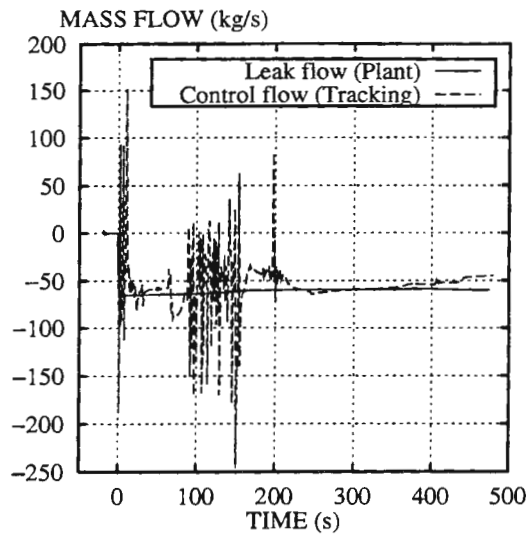
Figure 15 Leak flow in the plant and the steam flow that controls the primary pressure in the tracking simulator (SBLOCA).
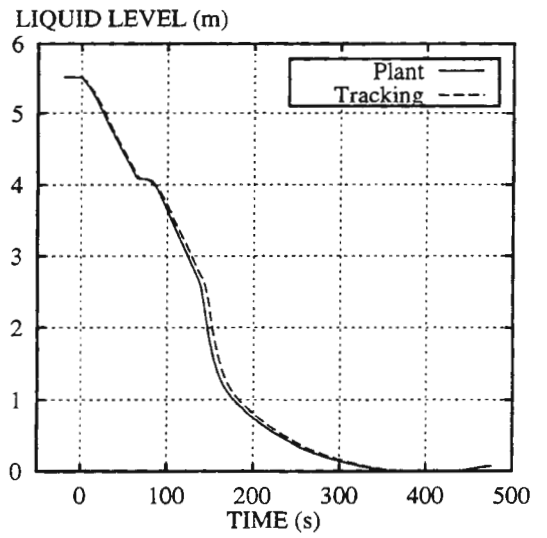
LIQUID LEVEL (m)

Figure 16 Liquid level of the pressurizer (SBLOCA).
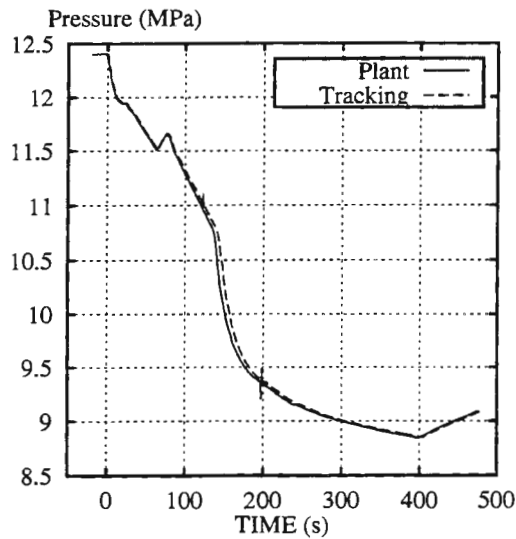
Pressure (MPa)

Figure 17 Pressure of the primary circuit (SBLOCA).

MASS (t)
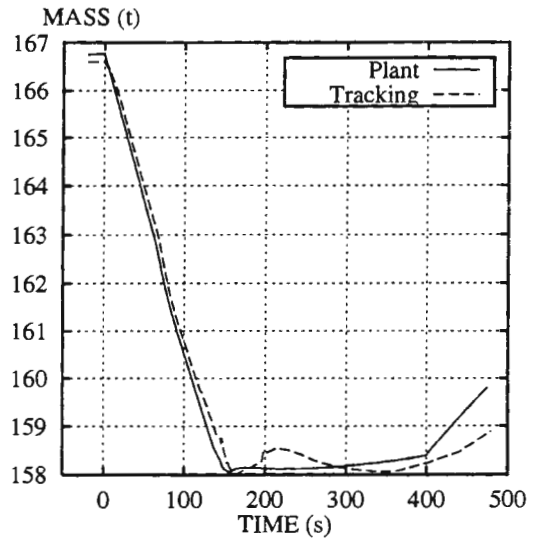
Figure 18 Mass balance of the primary circuit (SBLOCA).

THERMAL HYDRAULIC MODEL CAPABILITY IN THE APROS CODE ENVIRONMENT

Jaakko Miettinen, Markku Hänninen
VTT ENERGY
P.o. Box 1604, Fin-02044 VTT, Finland
Tel. +358 9 4561, Fax +358 9 456 5000
jaakko.miettinen@vtt.fi, markku.hanninen@vtt.fi

ABSTRACT

The capabilities of the thermohydraulic solutions in APROS are discussed. The thermohydraulic model of the APROS plant analyzer includes three level of solutions, homogeneous 3-equation model, 5-equation drift flux model and 6-equation two-fluid model. In practical modelling different approaches are selected for different type of the process system sections. The 3-equation model is used process networks conducting mostly single phase fluid and for turbines and condensers. The 5-equation model and 6-equation model are alternative models for describing two-phase thermohydraulics with phase separation. This type of description is important on the primary and secondary sides of a nuclear power plant, but also in some conventional process equipment. The 5-equation model is practical for real time applications in training simulators and the 6-equation model for the safety analyses and detailed process analysis. Both models have been validated matching the events expected during a PWR and BWR transients and accidents. In code applications many accident areas have been covered.

1. Introduction

APROS simulation environment has been developed in a co-operation project between Imatran Voima (IVO) and the Technical Research Centre of Finland (VTT). The simulator has an efficient and user friendly graphical interface and it can be used in the process and automation design, safety analysis and training for the nuclear, fossil and chemical processes. The sections for the nuclear power plant simulation includes 1D- and 3D-neutronics, thermohydraulics, automation, electrical network and containment simulation packages. The thermohydraulic model includes three level of solutions, 3-equation homogeneous model, 5-equation drift flux model and 6-equation two-fluid model. In this paper the thermohydraulics for main systems, 5-equation and 6-equation solutions are discussed.

The development of 5-equation and 6-equation thermohydraulic solutions started in the beginning of 1980'ies as separate projects. The development of the 5-equation model originated from practical need for a fast running thermohydraulic model for scoping studies of small break LOCA accidents. The 5-equation model is based on a non-iterative algorithm of five conservation equations with a single momentum equation for the mixture. The phase separation is solved by using the drift flux model.

The development of the 6-model started from the need to understand the characteristics of the two-fluid solutions at the time, when available RELAP5-versions for small break LOCA

studies were under development. The 6-equation model is based on a iterative algorithm of six conservation equations.

The development of the homogeneous model has proceeded as parallel to the development of the 6-equation model. Some experiences had been achieved from a pipe network solutions for single phase flow in a simulator project. The homogeneous model of APROS was an improved model considering homogeneous, equal velocity two phase flow. The same model can be then used for the single phase process networks, boiling and condensing heat exchangers, turbine and condenser. The homogeneous model has been applied extensively in simulation of gas mixture and other fluids than water.

The implementation of three solutions into same platform has allowed the systematic comparison of different approaches with respect to the physical content details, separate effects test cases, integral test cases and plant application results.

2. Thermal hydraulic model

In the present chapter the characteristics of the different TH-models, especially 5- and 6-equation models have been discussed. The basic conservation equations, basic formalisms for wall boundary conditions and for constitutive equations of both models are presented in Table 1. The basic equations for the 6-equation solution are (1), (2), (3), (4), (5) and (6) /1/. The 5-equation solution been discussed is based on equations (1), (2), (7), (5) and (6) /2/. For phase separation the drift flux model applies the formalism presented in the equation (8). The phase separation in the 6-equation model is based on the balance between wall friction to liquid, wall friction to gas and interfacial friction, i.e. the equations (9) and (10). In 5-equation model the friction is needed only between the wall and mixture and thus only equation (11) is needed. The basic formalism for the wall heat flux is according to the equations (12) and for the interfacial heat flux according to equations (13). The mass transfer due to interfacial heat transfer is defined as a energy balance over the interface. The wall heat flux partitioning to boiling and liquid heatup is a function of wall heat flux, liquid subcooling, flow rate and channel diameter.

On the basic equation level quite little differences exists between different solutions. Most important is the difference by describing the phase separation by using interfacial friction or drift flux correlation. But the numerical solution makes both solutions quite different. The 5-equation solution proceeds straight forward from pressure and velocity solution to solving phase separation and finally to solving phase enthalpies. The 6-equation solution includes corresponding steps but the solution is iterated so far the convergence has been reached.

The physical reality in the basic equations is defined by the constitutive models. The selection of constitutive models is presented in Table 2. Due to different development histories there are differences in constitutive models more than it would be necessary. The two-fluid formalism presumes that in 6-equation model the wall friction is divided into two fractions, one is the friction for liquid and another the friction for gas. The interfacial friction model is very essential, because it defines phase separation.

Mass conservation equations

$$\frac{\partial(A\alpha\rho_g)}{\partial t} + \frac{\partial(A\alpha\rho_g u_g)}{\partial z} = A\gamma + As_g \tag{1}$$

$$\frac{\partial(A(1-\alpha)\rho_l)}{\partial t} + \frac{\partial(A(1-\alpha)\rho_l u_l)}{\partial z} = -A\gamma + As_l \tag{2}$$

Momentum conservation equations, two fluid model

$$\frac{\partial(A\alpha\rho_g u_g)}{\partial t} + \frac{\partial(A\alpha\rho_g u_g^2)}{\partial z} = -A\alpha(\frac{\partial p}{\partial z} - \frac{\partial p_e}{\partial z} - \frac{\partial p_{fk}}{\partial z} - \frac{\partial p_i}{\partial z} - \rho_g g\cos\theta) \tag{3}$$

$$\frac{\partial(A(1-\alpha)\rho_l u_l)}{\partial t} + \frac{\partial(A(1-\alpha)\rho_l u_l^2)}{\partial z} = -A(1-\alpha)(\frac{\partial p}{\partial z} - \frac{\partial p_e}{\partial z} - \frac{\partial p_{fl}}{\partial z} + \frac{\partial p_i}{\partial z} - \rho_l g\cos\theta) \tag{4}$$

Energy conservation equations

$$\frac{\partial(A\alpha\rho_g h_g)}{\partial t} + \frac{\partial(A\alpha\rho_g u_g h_g)}{\partial z} = -A\alpha\frac{\partial p}{\partial t} + Aq_{wg} + Aq_{ig} + A\gamma h_{gs} + As_g h_{S,g} \tag{5}$$

$$\frac{\partial(A(1-\alpha)\rho_l h_l)}{\partial t} + \frac{\partial(A(1-\alpha)\rho_l u_l h_l)}{\partial z} = -A(1-\alpha)\frac{\partial p}{\partial t} + Aq_{wl} + Aq_{il} - A\gamma h_{ls} + As_l h_{S,l} \tag{6}$$

Momentum conservation equation, drift flux model

$$\frac{\partial(A\rho_m u_m)}{\partial t} + \frac{\partial(A_l\rho_m u_m^2)}{\partial z} = -A(\frac{\partial p}{\partial z} - \frac{\partial p_e}{\partial z} - \frac{\partial p_{fm}}{\partial z} - \rho_m g\cos\theta) \tag{7}$$

Drift flux model for phase separation

$$u_g = C_o j + v_{gj} \tag{8}$$

Wall friction for both phases (k=l,g), two fluid model

$$\frac{\partial p}{\partial z} = \frac{1}{2}C_k C_{fk}\rho_k u_k |u_k| \quad , \quad C_{fk} = C_{fk}(\text{Re}_k) \quad ; \quad with \quad \text{Re}_k = \frac{\alpha_k\rho_k u_k D_h}{\mu_k} \tag{9}$$

Interfacial friction, two fluid model

$$\frac{\partial p}{\partial z} = \frac{1}{2}C_i\alpha(1-\alpha)\rho_m(u_g - u_l)|u_g - u_l| \tag{10}$$

Wall friction, drift flux model

$$\frac{\partial p}{\partial z} = \frac{1}{2}(\frac{f}{D_e} + X)(\alpha\rho_g u_g|u_g| + (1-\alpha)\rho_l u_l|u_l|) \tag{11}$$

Wall heat flux to gas   Wall heat flux to liquid Wall heat flux to boiling

$$q_{wg}^{"} = h_{wg}(T_w - T_g) \quad , \quad q_{wl}^{"} = h_{wl}(T_w - T_l) , \quad q_{wb}^{"} = h_{wb}(T_w - T_s)^n \tag{12}$$

Interface to liquid heat flux (i=sat) Interface to gas heat flux (i=sat)

$$q_{il}^{"} = h_{il}(T_i - T_l) \quad , \quad q_{ig}^{"} = h_{ig}(T_i - T_g) \tag{13}$$

Boiling along the interface

$$\gamma = -\frac{q_{il} + q_{ig}}{h_{gs} - h_{ls}} \tag{14}$$

Table 1. Conservation equations, friction and phase separation models

For the wall heat transfer the different correlations have been used, but for typical light water reactor cases the differences in the wall heat flux calculation are quite small. The interfacial heat transfer models differ much more and the reason is that the 5-equation model was derived as non-iterative solution, which should tolerate large timesteps and the efficiency of correlation was limited. The condensation correlation gives a reasonable condensation rate

and the condensation is not flow rate dependent. In the 6-equation model the condensation is strongly dependent on the flow rate.

Physical phenomena	5-equation model	6-equation model
Wall friction	Blasius equation for mixture	Blasius eq. for both phases, factoring: void fraction and flow regime
Interfacial friction	Not modelled for the drift flux model	Separately: bubbly, annular, droplet, stratified, weighting function
Net vaporization	A linearized ramp function from subcooled liquid to saturation point	Separate gas interface, liq.interface, energy balance for interface
Pre-DNB heat transfer	Dittus-Boelter, Chen as simplified for boiling	Dittus-Boelter, Thom
Critical heat flux for wall heat transfer	Zuber-Griffith, VVER: Smolin, Bezrukow	Zuber-Griffith, Biasi, VVER: Smolin, Bezrukow
Post-DNB wall heat transfer	Dittus-Boelter to gas	Dittus-Boelter, Berenson
Special heat transfer models	No quench model yet	Heat transfer enhancement near quenching front
Interfacial condensation	Droplet type condensation or through stratified water level	Shah mode for liquid, Lee-Ryley model for gas
Interfacial flashing	Linear function of liquid mass and liquid superheat	Exponent function of void
Critical flow limitation	Sound velocity limitation or Moody model applied for the junction	With dense nodalization inherently or limitation to sound velocity
Pump characteristics	Four quadrant curves for head and torque for flow and pump speed.	Four quadrant curves for head and torque for flow and pump speed.
Phase separation	Drift flux model derived from EPRI correlation or full separation	Via interfacial friction correlation or full separation
Material property solution	Rational function fittings, two- or one parameter functions	Table interpolation as a function of pressure and enthalpy

Table 2. Constitutive thermohydraulic models for 5- and 6-equation solutions

There are some areas, where finding of constitutive model have been found problematic. The available correlations in the literature are typically valid on a limited parameter range. The selected correlations should be valid over the whole parameter range, from condenser pressure up to supercritical pressure, from ice temperature to the fuel melting temperature, from zero flow to sound velocity.

The problematic areas may be listed as follows:

- Applying sound velocity for choking flow of the two-phase mixture.
- Calculation of net evaporation for the wall heat flux.
- Deriving all range to material correlations.
- Pump head and torque characteristics for two phase flow.
- Quench front model during large break LOCA reflooding.

The numerical solution for the 5-equation model is a predictor-corrector type noniterative solution. The solution for the 6-equation is iterative. In both models the sparse matrix inversion is used for the solving the pressure, void fraction and energy equations. In Table 3 the main features related to the 5- and 6-equation numerical solution are presented. Although

one model is non-iterative and another iterative, both solution include similar type of solution steps.

Numerical feature	5-equation model	6-equation model
Numerical solution principle	Predictor - correction solution without iteration	Iteration until the final convergence
Solution scheme	1. Predict heat transfer and mass transfer 2. Solve pressures 3. Update heat and mass tr. 4. Calculate mixture velocity 5. Calculate drift flux separation 6. Solve mass flows 7. Integr. Mass conservation 8. Solve phase enthalpies 9. Update material properties 10. Proceed to the next timestep	1. Solve pressures 2. Calculate phase velocities 3. Update linear densities 4. Solve void fractions 5. Solve heat structures 6. Calculate heat flows 7. Solve phase enthalpies 8. Update material properties 9. Check convergence, if not, return to beginning
Matrix inversions	Pressure Void fraction Heat structure temperature Gas enthalpy Liquid enthalpy	Pressure Void fraction Heat structure temperature Gas enthalpy Liquid enthalpy
Discretized scheme	Staggered grid for space discretization Implicit in time advancement	Staggered grid for space discretization Implicit in time advancement
Timestep in the calculation	0.01 ... 0.2 s, fixed time step, short timesteps in rapid transients	0.01 ... 0.1, automatic time step control
Source term linearizations	Linearizations in respect pressure, liquid temperature, gas temperature and wall temperature.	Linearizations in respect pressure, void fractions, velocities, surface temperatures

Table 3: Numerical solutions in 5-equation and 6-equation thermohydraulics

Even if the APROS thermal-hydraulic models calculates the flows, enthalpies and pressures numerically using finite difference scheme, the definitions of the simulated process can be made by using process component modules. An user gives data to the process components that exist in the real power plants - conventional or nuclear. The typical process components are pipes, valves, pumps, tanks and heat exchangers. At the same time when a user specifies the necessary data of a process component, he defines how in detail the process component is to be simulated, i.e. how many finite difference elements are used in simulation. The APROS initialization programs create automatically the calculation nodalization on the basis of process component input data and process component connections.

3. Validation thermohydraulics

The validation program of thermohydraulics includes analyses of separate effects tests, integral tests and real plant transients. The separate effects tests are focused on specific physical phenomena. Their advance are detailed measurements around the studied phenomena in such an accuracy, which is not possible integral experiments. The model development need

typically a lot of measurement data on specific physical phenomena and that is why the separate effect tests are popular for the code development. When only limited phenomena are studied, the boundary conditions are clearly defined and the experiment may be modelled accurately.

Two types of separate effects test may be mentioned for nuclear power plant related phenomena. Some separate effects tests are focused on the specific physical phenomenon like heat transfer or critical flow. Another class of tests is focused on a special behaviour of a specific process component. The core uncovery caused by the loop seals and counter-current-flow-limitation at core exit are examples of this type of phenomena, studied in many experimental projects.

Phenomena to be analyzed	APROS validation cases matching the group	5-eq.	6-eq.
Basic phenomena −evaporation − flashing −condensation −friction −pressure drop	Edwards pipe, Battelle Edwards pipe Lotus air/water annular flow Edwards pipe, Battelle, ISP-6	x x x	x x x x
Critical flow	Edwards pipe, Marviken, ISP-27	x	x
Vertical phase separation	Battelle, ISP-6	x	x
Horizontal stratified flow	BETHSY ISP-38	x	x
Phase separation at branches	BETHSY, ISP-38		
Entrainment/de-entrainment	Lotus air/water annular flow	x	x
Liquid-vapour mixing, condensation			
Stratified condenssation	NOKO EC bundle	x	x
Spray effects	Loviisa pressurizer	x	x
Countercurrent flow	IVO CCFL	x	x
Multidimensional void/temp.			
Heat transfer: − pre DND − critical heat flux − post DNB	FRIGG, Christensen Becker Becker	 x x	x x x
Quench front propagation	ERSEC - 7, REWET-II	x	x
Lower plenum flashing			
Guide tube flashing			
Pump one-and two-phase char.	Loviisa pump trip, Loft	x	x
Separator and dryer behaviour			
Accumulator behaviour	Loft 2-5, Loviisa		x
Loop seal filling and clearance	IVO loop seal tests, Pactel	x	x
ECC bypass / downcomer penetrations			
Parallel channel instability			
Boron mixing and transport			
Lower plenum entrainment			
Noncondensible gas effecs			

Table 4. Requirements for the core validation

The OECD/NEA-CSNI Separate Effects Test Matrix for Thermal Hydraulic Code Validation /3/ defines the best separate effect tests for studying different physical phenomena. The matrix

includes typically 3 - 5 recommended separate effects tests for the code validation. In Table 4 matching of the APROS validation cases belonging to different categories are presented.

The Edwards pipe test case was a horizontal pipe which was closed in the beginning of the test and contained water under high pressure. The purpose of the Battelle blowdown experiment was to examine the steam and mixture blowdown from the top of the pressure vessel. The aim of Marviken critical flow tests was to obtain the data for critical flows in large diameter pipes. In Beckers experiment the heat transfer crisis and post-dryout heat transfer was studied in a single tube. These experiments are clearly focused on basic physical phenomena.

The IVO loop seal experiments is an example of separate effects tests related on a plant specific problematics. The water remaining in the loop seal during a loss of coolant accident may lead to the reactor core uncovery. The IVO loop seal experiments were conducted for the real loop seal geometry of the VVER-1000 type of PWR at atmospheric pressure by injecting compressed air. In the experiments the phase separation and remaining liquid mass is essential.

Another group of validation cases are s.c. intergral tests with scaled thermohydraulic loops PWR or BWR type light water reactor plants. The integral test cases include medium size LOCA in the LOFT facility (L3-5), small break LOCA in the LOFT facility (L3-6), small break LOCA in BETHSY facility and midloop operation test in BETHSY facility. The integral test experiments in the BETHSY facility for the 2 inch cold leg break (ISP-27) in a PWR and midloop operation test (ISP-38) have been calculated with both 5-equation and 6-equation models.

4. Requirements to thermohydraulic models in different plant incidents

In Table 5 the plant specific features for different transients are collected for the plant simulation. Before incidents are calculated, the analyst should be aware on if the thermohydraulic model is good enough taking into account specific thermohydraulic features in the incidents. Some important phenomena may be validated by using separate effects tests. On the other hand there are many phenomena, which need validation by integral tests. The natural circulation modes with single phase circulation, two-phase circulation and reflux condensing model are examples from this kind of phenomena.

5. Plant applications for VVER type of PWR plants

The APROS 5-equation model has been mainly used in simulator applications. The 6-equation models have been used as a nuclear plant analyzer and for safety analyses. One idea has been that the models for the analysis application and training application cannot be distinguished any more. The detailed models for auxiliary systems, automation and electrical systems are needed in both cases. The simulator model may be used for analysis applications with comparison against system codes by changing the 5-equation solution with that of 6-equation. The modification requires only few hours. Vice versa the simulator model may be generated by changing 6-equation solution to 5-equation solution.

Incident	Simulation requirements
50 - 200 % PWR LOCA blowdown	Fast depressurization, pressure wave propagation, delayed flashing, critical heat flux in the core, two-dimensional downcomer flow.
50 - 200 % PWR LOCA emergency cooling	Quench front propagation, droplet entrainment, energy non-equilibrium in post dryout, radiation heat transfer
0.1 - 10 % PWR LOCA (small break LOCA)	Natural circulation modes (single phase, two-phase, reflux condensing mode), system depressurization
Steam line break (PWR)	Primary side overcooling, core recriticality, pressure vessel cold stress
Reactor scram, turbine trip, main coolant pump trip (PWR)	Heat transfer crisis, actuation of diffrent signals, hot channel and hot rod behaviour
Control rod ejection, control rod withdrawal (PWR)	Neutronics 3-D -behaviour due to reactivity changes.
Loss of offsite power	Decay heat removal of the reactor core
ATWS after loss-of feedwater (PWR)	Neutronics and thermohydraulics interactions, axial shape of neutron flux, natural circulation modes
Reactivity accident by boron dilution or cold water plug (PWR)	Non-diffuse numerics for calculating the propagation of the boric acid and cold water front
Steam line or feedwater line LOCA	Starting the high pressure injection, core cooling by spray or bottom flooding.
Operational transient as recirculation pump trip(BWR)	Heat transfer crisis in the core, dynamics of the reactor scram
Steam line isolation (BWR)	Power spiking (may be 200-400 % peaking power, before feedback from the void affects) in connection with pressure spiking and core void collapse, pressure wave propagation in the steam line, steam relief capacity, direct heat absorption into coolant, effect of wet steam on presure behaviour
Loss of off-site power (BWR)	Securing emergency feedwater injection. After core heatup melting of control rods and possibility for the criticality.

Table 5. Simulation requiremens in different type of accidents

VVER -related applications for APROS have included /4/
- Large Break LOCA cases, comparisons against DRUFAN and FLUT analyses
- REWET tests for reflooding
- Small break LOCA, comparisons against RELAP -analyses
- Primary to secondary leak analyses
- Real plant incident including erronous opening of the turbine bypass valve
- Main coolant pump trip analyses
- Turbine trip analyses
- Natural circulation calculation, comparison with RELAP analyses
- ATWS LOOP, comparison against HEXTRAN -SMABRE analyses.

6. Plant applications for ASEA type BWR plants

A transient analysis model of the Olkiluoto BWR has been created with APROS. The model was initially developed for calculation of three verification cases. The main systems: reactor vessel, steam lines, feed water lines and relief system have been modelled using both the 5-

and 6-equation models. Auxiliary systems have been described with the homogeneous model. The primary system model starts from the feed water pumps and ends at the high pressure turbine and steam dump control valves.

Three transients have been analysed with the Olkiluoto model. These are:

- Double ended steam line break with simultaneous loss of electricity
- Loss of feed water, ATWS
- Simultaneous closure of all steam line isolation valves

7. Conventional plant applications

Even if the flow models have been originally developed for nuclear applications, APROS has been extensively used for simulation of different type of conventional power plants. The power plant components like water walls, economisers, super heaters and drums can be simulated with the similar models as nuclear power plants. The APROS homogeneous model includes the general concentration solution that can be utilized to simulate the flue gas mixture produced by burning of different type of fuel. The heat released in burning is calculated on the basis of burning reactions. Both radiation and convection heat transfer from the gas to boiler components can be calculated. For the simulation of the combined cycle power plants APROS model library includes models for the gas turbine and compressor. The continuous development of the homogeneous model has enlarged the capability of APROS in the simulation of the conventional power plants.

8. Conclusions

The homogeneous, five equation and six equation thermohydraulics solution in APROS offer versatile possibilities for nuclear power plant and process component simulation. The homogeneous model is best for process systems without phase separation, the five equation solution for fast running real time simulation and the six equation solution for the engineering studies and safety analyses. The same development platform for both models makes the model comparison efficient, because practically the same input models can be used for both solutions. The five equation model is a best for real time simulator applications due to fast calculation.

References
/ 1/ The One-Dimensional Two-Fluid Model of APROS, M. Hänninen, J. Ylijoki, HOHTI-2/92.
/ 3/ Fast Running Five Equation Thgermohydraulics of APROS, J.Miettinen, ROIMA-5/98.
/ 3/ OECD/NEA-CSNI Separate Effects Test matrix for Thermal-Hydraulic Code Validation, N.Aksan, F. D'Auria, H.Glaeser, R. Pochard, C.Richards and A. Sjoeberg, 1993, CSNI-Report (Draft).
/ 4/ Experiences of APROS in Nuclear Power Plant Safety Analysis, H.Kantee, S.Savolainen, H.Kontio, CSNI Specialist Mtg. on Simulators and Plant Analyzers, Helsinki 1998.

AN OVERVIEW OF THE MODELING LANGUAGE MODELICA

Sven Erik Mattsson[1], Hilding Elmqvist[2]

[1]Dept of Automatic Control, Lund University, Box 118, SE-221 00 Lund, Sweden, SvenErik@control.lth.se
[2]Dynasim AB, Research Park Ideon, SE-223 70 Lund, Sweden, Elmqvist@Dynasim.se

Abstract.   A new language called Modelica™ for physical modeling has been developed in an international effort. The main objective is to make it easy to exchange models and model libraries. The design approach builds on non-causal modeling with true ordinary differential and algebraic equations and the use of object-oriented constructs to facilitate reuse of modeling knowledge.

Introduction

Mathematical modeling and simulation are emerging as key technologies in engineering. Relevant computerized tools, suitable for integration with traditional design methods are essential to meet future needs of efficient engineering. There is a large amount of simulation software on the market. However, languages and model representations are proprietary and developed for certain tools. Most general-purpose tools are based on the same modeling methodology, input-output blocks, as in the previous standardization effort, CSSL, from 1967 [6], when it was necessary to put more concern on computational aspects than on user aspects. The modeller has to perform the tedious work of transforming conservation laws and constitutive equation into an explicit ordinary differential equation, ODE, system. Domain-oriented packages are with few exceptions, only strong in one domain and are not capable of modeling components in other domains reasonably. This is a major disadvantage since technical systems are becoming more and more heterogeneous with components from many engineering domains.

Techniques for general-purpose physical modeling have been developed during the last decades, but did not receive much attention from the simulation market. The modern approaches build on non-causal modeling with true equations and the use of object-oriented constructs to facilitate reuse of modeling knowledge. There are already several modeling languages with such a support available from universities and small companies. There is also significant experience of using these languages in various applications.

In October 1996 an international effort started to unify the concepts of these languages in order to introduce common basic syntax and semantics and to design a new unified modeling language. The language is called Modelica[1]. The main objective is to make it easy to exchange models and model libraries and to allow users to benefit from the advances in object-oriented modeling methodology. In February 1997 the Modelica design effort became a Technical Committee within the Federation of European Simulation Societies, EUROSIM. The Modelica effort started in the continuous time domain since there is a common mathematical framework in the form of differential-algebraic equation (DAE) systems and there are several existing modeling languages based on similar ideas. Modelica version 1.0, was finished in September 1997. It is based on DAE systems with some discrete-event features to handle discontinuities and sampled systems.

The Modelica Design Group: *Manuel Alfonseca*, Universidad Autonoma de Madrid, Spain, *Bernhard Bachmann*, ABB Corporate Research, Baden-Dättwil, Switzerland, *Fabrice Boudaud* and *Alexandre Jeandel*, Gaz de France, *Jan Broenink*, University of Twente, The Netherlands, *Dag Brück* and *Hilding Elmqvist* (chairman), Dynasim AB, Lund, Sweden, *Thilo Ernst*, GMD-FIRST, Berlin, Germany, *Rüdiger Franke*, Technical University of Ilmenau, Germany, *Peter Fritzson*, Linköping University, Sweden, *Kaj Juslin*, VTT, Finland, *Matthias Klose*, Technical University of Berlin, Germany, *Sven Erik Mattsson*, Lund University, Sweden, *Pieter Mosterman* and *Martin Otter*, DLR Oberpfaffenhofen, Germany, *Per Sahlin*, BrisData AB, Stockholm, Sweden, *André Schneider* and *Peter Schwarz*, Fraunhofer Institute for Integrated Circuits, Dresden, Germany, *Hubertus Tummescheit*, GMD FIRST, Berlin, Germany, *Hans Vangheluwe*, University of Gent, Belgium.

---

[1]Modelica™ is a trade mark of the Modelica Design Group

Modelica Fundamentals

In order to give an introduction to Modelica we will consider modeling of a simple electrical circuit as defined in Figure 1. The system can be broken up into a set of connected electrical standard components. We have a voltage source, two resistors, an inductor, a capacitor and a ground point. Models of these components are typically available in model libraries. Using a graphical model editor we can define a model by drawing an object diagram as shown in Figure 1, by positioning icons that represent the models of the components and drawing connections.

A Modelica model of the circuit is given in Figure 2. It is a composite model which specifies the topology of the system to be modeled in terms of components and connections between the components. The statement "`Resistor R1 (R=10);`" declares a component R1 of class `Resistor` and sets the default value of the resistance R to 10.

Variables and connectors

Connections specify interactions between components. A connector must contain all quantities needed to describe the interaction. For electrical components we need the quantities voltage and current. A type for voltage quantities is declared as **type** `Voltage = Real(Unit = "V");` where `Real` is the name of a predefined type. A real variable has a set of attributes such as unit of measure, minimum value, maximum value and initial value.

To simplify the use of Modelica and to support compatibility, there is an extensive standard library of type definitions which always is available with a Modelica translator. The type definitions in this base library are based on ISO 1000 and its naming conventions for physical quantities. Several ISO names are long, which make them awkward in practical modeling work. For this reason, shorter alias-names are provided if necessary. The use of the name `ElectricPotential` repeatedly in a model becomes cumbersome and therefore the alternative `Voltage` is provided.

Defining a set of connector classes is a good start when developing model libraries for a new application domain. It promotes compatibility of the component models. A connector class, `Pin`, with voltage and current for electrical connections is defined in Figure 3.

A connection, `connect(Pin1, Pin2)`, with Pin1 and Pin2 of connector class `Pin`, connects the two pins such that they form one node. This implies two equations, namely `Pin1.v = Pin2.v` and `Pin1.i + Pin2.i = 0`. The first equation indicates that the voltages on both branches connected together are the same, and the second corresponds to Kirchhoff's current law saying that the current sums to zero at a node. Similar laws apply to flow rates in a piping network and to forces and torques in a mechanical system. The sum-to-zero equations are generated when the prefix `flow` is used in the connector declarations. In Modelica it is assumed that the value is positive when the current or the flow is into the component.

```
model Circuit
    Resistor  R1 (R=10),   R2 (R=100);
    Capacitor C  (C=0.01);
    Inductor  L  (L=0.1);
    VsourceAC AC;
    Ground    G;
equation
    connect(AC.p, R1.p);  // Capacitor circuit
    connect(R1.n, C.p);
    connect(C.n,  AC.n);
    connect(R1.p, R2.p);  // Inductor circuit
    connect(R2.n, L.p);
    connect(L.n,  C.n);
    connect(AC.n, G.p);
end Circuit;
```
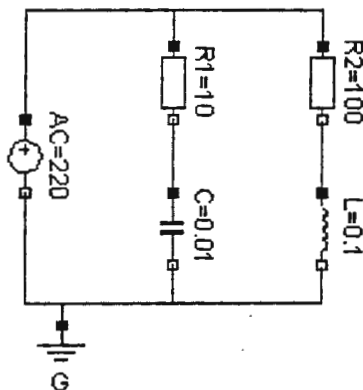
Figure 1 A simple electrical circuit.   **Figure 2** A Modelica model of the circuit in Figure 1.

```
connector Pin
   Voltage v;
   flow Current i;
end Pin;
```

Figure 3   A connector representing a pin.

```
partial model TwoPin
   Pin p, n;
   Voltage v;
equation
   v = p.v - n.v;   p.i + n.i = 0;
end TwoPin;
```

Figure 4   Shell model with two electrical pins.

Partial models and inheritance

A very important feature in order to build reusable descriptions is to define and reuse *partial models*. A common property of many electrical components is that they have two pins. This means that it is useful to define a "shell" model class TwoPin, that has two pins, p and n, and a quantity, v, that defines the voltage drop across the component. See Figure 4. The equations define common relations between quantities of a simple electrical component. In order to be useful, a constitutive equation must be added. The keyword partial indicates that this model class is incomplete.

To define a model for a resistor, start from TwoPin and add a parameter for the resistance and Ohm's law to define the behavior. A model for a capacitor is defined similarly.

```
model Resistor "Ideal resistor"
   extends TwoPin;
   parameter Resistance R;
equation
   R*p.i = v;
end Resistor;
```

```
model Capacitor "Ideal capacitor"
   extends TwoPin;
   parameter Capacitance C;
equation
   C*der(v) = p.i;
end Capacitor;
```

A string between the name of a class and its body is treated as a comment attribute. Tools may display this documentation in special ways. The keyword parameter specifies that the quantity is constant during a simulation experiment, but can change values between experiments. A parameter is a quantity which makes it simple for a user to modify the behavior of a model. The expression der(v) means the time derivative of v.

Modelica's facility with partial and extends is similar to inheritance in other languages. Multiple inheritance, i.e., several extends statements, is supported. The type system of Modelica is greatly influenced by type theory [1], in particular the notion of subtyping (the structural relationship that determines type compatibility) which is different from subclassing (the mechanism for inheritance). The main benefit is added flexibility in the composition of types, while still maintaining a rigorous type system. Inheritance is not used for classification and type checking in Modelica. An extends clause can be used for creating a subtype relationship by inheriting all components of the base class, but it is not the only means to create it. Instead, a class A is defined to be a subtype of class B, if class A contains all the public components of B. In other words, B contains a subset of the components declared in A. As an example, consider a varistor model

```
model Varistor
   extends TwoPin;
   parameter Resistance R;
   parameter Voltage    Uc "Cut-off voltage";
equation
   v = Uc*atanh(R/Uc*p.i);
end Varistor;
```

It is not possible to extend this model from the ideal resistor model Resistor, because the equation of the Resistor class needs to be replaced by a new equation. Still, Varistor is a subtype of Resistor because it contains all the public components of Resistor. This subtype relationship is especially used for class parameterization as explained in the next section, which discusses a more powerful parameterization, not only involving values like time constants and matrices but also classes.

Class Parameterization

Consider the model `Circuit` in Figure 2. Assume that we would like to exchange the resistor models with the varistor model, `Varistor`, while retaining the parameter values given for `R1.R` and `R2.R` and the circuit topology. This can be accomplished by redeclaring `R1` and `R2`:

```
model Circ2 = Circuit(redeclare Varistor R1, redeclare Varistor R2);
```

It is possible to replace the ideal resistor model, since `Varistor` is a subtype of `Resistor`. A value can be given to the parameter `Uc` in the redeclaration: `redeclareVaristor R1(Uc=100);`

This is a very strong modification of the circuit model and there is the issue of possible invalidation of the model. The keyword `redeclare` clearly marks such modifications. Furthermore, the modeller of `Circuit` is able to state that such modifications are not allowed by declaring a component as final: `final Resistor R2(R=100);` It is also possible to state that a parameter is frozen to a certain value, i.e., is not a parameter anymore: `Resistor R2(final R=100);`

To use another resistor model in `Circuit` we needed to know that there were two replaceable resistors and their names. To avoid this problem and prepare for replacement of a set of models, one can define a replaceable class, `ResistorModel` in the circuit model:

```
model Circuit2
replaceable model ResistorModel = Resistor;
protected
  ResistorModel R1(R=10), R2(R=100);
  // then as Circuit.
```

The replaceable model `ResistorModel` is declared to be of type `Resistor`. This means that it will be enforced that the actual class will be a subtype of `Resistor`, i.e., have compatible connectors and parameters. It must have `Pins p` and `n` and a parameter `R`. Default for `ResistorModel`, i.e., when no actual redeclaration is made, is `Resistor`. Setting `Resistormodel` to `Varistor` is done as

```
model MyCircuit2 = Circuit2(redeclare model ResistorModel = Varistor);
```

Replaceable classes are useful for medium parametrization. It makes it possible to separate medium properties from device properties in libraries for process components such as pumps, heat exchangers, chemical reactors etc. [4]. Another example is a controlled plant where some PID controllers are replaced with auto tuning controllers. It is of course possible to just replace those controllers in a graphical user environment, i.e., to create a new model. The problem with this solution is that two models must be maintained. Modelica has the capability to instead just substitute the model class of certain components using a language construct at the highest hierarchical level, so only one version of the rest of the model is needed.

Matrices

Modeling of, for example, multi-body systems, control systems and approximations to partial differential equations is done conveniently by utilizing *matrix equations*. Multi-dimensional matrices and the usual matrix operators and matrix functions are thus supported in Modelica. It is also possible to have arrays of components and to define regular connection patterns. A typical usage is the modeling of a distillation column which consists of a set of trays connected in series.

A matrix variable can be declared by appending dimensions after the name: `Real S[3, 3];` It is also possible to declare matrix types: `type Transformation = Real[3, 3];`.

To describe interaction between rigidly connected bodies in a 3D free-body diagram, define

```
connector MbsCut
  Transformation S  "Rotation matrix for frame A relative to the inertial frame";
  Position3      r0 "Vector from the origin of the inertial frame to A's origin";
  flow Force3    f  "Resultant cut-force acting at A's origin";
  flow Torque3   t  "Resultant cut-torque with respect to A's origin";
end MbsCut;
```

A model for a rigid bar can be defined as

```
model Bar "Massless bar with two mechanical cuts."
  MbsCut a b;
  parameter Position3 r[3] = [0, 0, 0]  "Position vector from a to b";
equation
  b.S = a.S;      b.r0  = a.r0 + a.S*r;   // Kinematic relations between a and b
  0 = a.f + b.f;         :                // Force balance
  0 = a.t + b.t - cross(r, a.f);          // Torque balance
end Bar;
```

Other Modeling Features

Modelica supports *hybrid modeling*. Realistic physical models often contain discontinuities, discrete events or changes of structure. Examples are relays, switches, friction, impact, sampled data systems etc. Modelica has introduced special language constructs allowing a simulator to introduce efficient handling of such events.

Algorithms and functions are supported in Modelica for modeling parts of a system in procedural programming style. Constructs for including *graphical annotations* are available in order that also icons and model diagrams become portable. An extensive *Modelica base library* containing standard variable and connector types promotes reuse by standardizing on interfaces.

Conclusions

A short overview of Modelica has been given. The expressive modeling power of Modelica is large. Model classes and their instantiation form the basis of hierarchical modeling, connectors and connections correspond to physical connections of components. At the lowest level, equations are used to describe the relation between the quantities of the model.

Modelica has been used to model various kinds of systems. Modeling of automatic gearboxes for the purpose of real-time simulation is described in [5]. Such models are non-trivial because of the varying structure during gear shift utilizing clutches, free wheels and brakes. Modeling of heat exchangers is discussed in [4]. Class parameters of Modelica are used for medium parametrization and regular component structures are used for discretization in space of the heat exchanger. Thermodynamical and flow oriented models are also discussed in [3]. A Modelica library supporting the bond graph modeling methodology is described in [2].

Several Modelica tools and model libraries are under development. There are discussions to extend Modelica to support, for example, partial differential equations and discrete event models.

More information, including modeling requirements, rationale and definition of Modelica and future developments is available on WWW at URL: http://www.Dynasim.se/Modelica/.

References

[1] M. ABADI and L. CARDELLI. *A Theory of Objects.* Springer-Verlag, 1996.

[2] J. F. BROENINK. "Bond-graph modeling in Modelica." In *Proceedings of the 1997 European Simulation Symposium (ESS'97)*, Passau, Germany, October 1997. The Society for Computer Simulation.

[3] T. ERNST, M. KLOSE, and H. TUMMESCHEIT. "Modelica and Smile — A case study applying object-oriented concepts to multi-facet modeling." In *Proceedings of the 1997 European Simulation Symposium (ESS'97)*, Passau, Germany, October 1997. The Society for Computer Simulation.

[4] S. E. MATTSSON. "On modeling of heat exchangers in Modelica." In *Proceedings of the 1997 European Simulation Symposium (ESS'97)*, Passau, Germany, October 1997. The Society for Computer Simulation.

[5] M. OTTER, C. SCHLEGEL, and H. ELMQVIST. "Modeling and realtime simulation of an automatic gearbox using Modelica." In *Proceedings of the 1997 European Simulation Symposium (ESS'97)*, Passau, Germany, October 1997. The Society for Computer Simulation.

[6] J. C. STRAUSS (ED.). "The SCi continuous system simulation language (CSSL)." *Simulation*, 9, pp. 281–303, 1967.

A Free Context Physics Knowledge Representation suitable for Object-Oriented Modelling Tools

Juan José Ramos[1], Miquel A. Piera[2], Ignasi Serra[3]

Department of System Engineering and Automatic Control, Autonomous University of Barcelona.
C-Building, ES-08193 Bellaterra. [1] juanjo@sunaut.uab.es , [2] miquel@sunaut.uab.es, [3] ignasi@sunaut.uab.es

Abstract

Increasing interest in simulation techniques demands of modelling tools to support full reusability of predefined models preserving system?model analogy. The constrains presented by equation-oriented formalisms are analysed in this work. A new modelling language supporting reusability by means of physical causality analysis is introduced.

Introduction

Nowadays there is an increasing industrial interest in the use of modelling and simulation tools. However, there are many industrial branches where these techniques are scarcely used. A reason could be that the development of mathematical models for complex systems is difficult and time consuming. To extend the use of simulation techniques, the effort has been mainly focused in creating different environments that simplify the modelling task. To deal with large and complex systems, it has been widely stated the need of modelling tools supporting libraries of non-causal models that can be linked together in the same way as the physical units are assembled in the system. Object-Oriented Modelling paradigm emerges as a methodology based on the ideas from Object Oriented Programming of the reuse of modular code.

New generation object oriented modelling languages as Omola, Dymola and recently Modelica [1] are based on these ideas. Complex systems can be modelled as a structure of interconnected submodels. Nevertheless, the description of the overall system behaviour it is not always achieved just by the interconnection of the equations which describe each submodel behaviour. Common difficulties, which not allow a fully reuse of predefined models, arise when it is not possible to describe in advance the coupling phenomena resulting from subsystem interactions. Some roots of these problems can be found in the equation-oriented representation of physical behaviour in a model:

- *Representation of phenomena and interactions.* In present object oriented modelling languages, behaviour is encapsulated in the model using acausal equation formalism. They also supply with mechanisms to describe coupled models by means of connectors where the variables shared between a model and its environment are defined. Since generation of state-space models, or a more generic DAE causal sequence is required for simulation, the structure of all these equations is analysed in order to determine which variable has to be solved for in each equation.

  Different approaches can be used to achieve DAE formalism. For instance, in Bond Graph modelling methodology, subsystems are represented by objects describing energy storage, energy transformation and energy dissipation (vertices in bondgraphs), and physical interconnections are represented by terminals (bonds) describing power flow by means of couples of flow and effort variables. The computational causality can be derived by the well-known bondgraphs sequential causality assignment procedure [6]. In this translation procedure, from a modular and acausal equation formalism into a causal model, systems of equations resulting from submodel coupling often present non linear constraints over state variables (Structurally Singular Models) [2].

  Though recent advances in numerical solvers for DAE systems, the problems derived from the use of such solvers could be avoided when the singularities appear as the result of the interconnection of different components describing the same phenomenon, since for an expert it is possible to find a mathematical formulation where the singularity does not arise [5].

- *Medium property propagation.* In many types of systems, mainly in chemical processes, the medium (an entity) properties propagation is related with matter transport through system components, which often do not affect such properties. It is difficult to obtain a generic modular representation of property propagation since it depends on many factors. For instance, the change in a system of a single medium (water) for a composite medium (with different product concentrations) will drastically affect the overall requirements for the behaviour description of the system components, even in those components that are not medium dependent. The laws required to describe phenomena are medium dependent and the terminals used to represent subsystem interactions are not context independent. When systems interact through a medium that flows between them, it can appear an interaction, derived from the transport phenomenon, which would require the use of complicated terminals [5]. Somehow that breaks the object oriented encapsulation principle.

Despite the fact that solutions as medium separation from equipment [3], or Model Parameterisation [1,3] have been thoroughly investigated in object oriented modelling, equation oriented languages do not, nowadays, offer to model user a clear feedback to decide how to reuse and parameterise a model in a particular context.

In order to avoid these problems a new modelling language, *Physics Modelling Language* (PML), has been developed. The language has been designed to support modular data structures where the problem of representing the physical behaviour and the problem of finding a suitable formalism for the simulation procedure are separated. *Physics Modelling Tool* (PMT) is a graphical modelling environment used to define structured models preserving the topology of the system. It is being developed under X and Unix like operative systems. In second section it is presented an overview of the modelling methodology and the required knowledge representation. The third section describes the algorithm to solve the physical causality analysis problem. In fourth section it is presented the strategy for physical knowledge distribution.

Modular representation of system behaviour

Modelling should be a matter of having the most convenient tools to specify models. Representations of systems as circuit diagrams or flowsheets are familiar to most engineers. Those representations preserve the topological structure of a system. A language that pretends to support this modelling approach should provide with reuse of predefined models and with the consequent manipulation procedures required to achieve an adequate computational formalism (simulation model). Unfortunately, a model definition according to system topology does not always preserve a computational structure suitable for simulation.

Fully reuse of predefined models is supported whether it is possible to develop a model library without considering the reuse context. To automatically reuse a predefined model in a new context, two basic preconditions should be fulfilled: the interaction description mechanism must be independent from any assumption about the variables needed to interconnect models; second, the representation of the resulting coupling behaviour might be adaptable to each particular context. Actual OOM languages attempt to find the simulation model from the behaviour equations defined in submodels, together with the coupling equations derived from terminal connections. In general, it is not possible to find a mathematical description of all the physical relationships derived from subsystem interactions. The equations describe how a phenomenon can be computed, but they are not able to describe the phenomenon itself. For instance, reservoir system behaviour can be described in terms of expressions as Equation 1:

$$\frac{dm}{dt} = m_i - m_o \qquad (1)$$

For the model user view, such representation collects all the concepts around the physical behaviour: the involved entity (matter), the phenomenon (matter storage), the law which describes it (matter conservation principle) and the computable property of the matter (mass balance). It is always possible to go from the knowledge of physic laws to a mathematical representation, therefore phenomena can be formulated
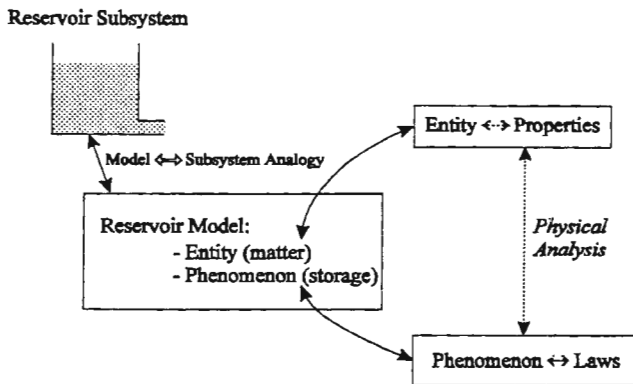
Figure 1 Model representation structure

according to each particular context. In the reservoir system, storage phenomenon could be adapted to the case where different matter types are involved. However, from the equation oriented language view, expressions as Equation 1 have not information enough to abstract this physical behaviour. Equations are static representation of behaviour; meanwhile reusability requires a behaviour representation adaptable to context.

The new modelling language, PML, is proposed to allow the representation of system behaviour and interactions in such a way that the equation oriented formalism can be dynamically stated once the model reuse context is analysed. A modelling environment, PMT, is being developed to support the translation procedures that find the suitable computational formalism from such physical behaviour representation. The proposed modelling methodology is based on the object oriented and modular model development paradigm, preserving the system ⇔ model structure analogy.

In PMT there are three main stages in the modelling-towards-simulation process: model definition, physical causality analysis and simulation-oriented model generation. A composite model is built up by connecting its components (predefined submodels) defining the *topological model* of the system. The topology of connections is analysed by the modelling tool in order to establish, from a physics point of view, the consequent phenomena and interactions. In last term, the functional model, where the adequate coupling behaviour is represented, is used to derive an equation-oriented formalism for numerical simulation. The language PML has been designed to support this translation procedure.

Representation of system behaviour in PML.

The language has been designed to support the knowledge needed to analyse complex model behaviour from a physics causality analysis view rather than from a computational causality analysis view. The main structure of a model is represented in Figure 1. In order to preserve system ⇔ model analogy, the *model* concept is tightly related to the system component. Therefore, end user may readily identify in the library of predefined models the duality system ⇔ model component. The behaviour representation in a model has been split in two main objects: *entities* and *phenomena*. The *entity* is the subject of the physical process carried out by a system component. The phenomena will represent those aspects of the physical process that are of interest for a particular simulation experiment. The properties of an entity are modified by the phenomena, and laws are used to define such dynamics. For example, a *tank* (model) is a system component where *matter* (entity) is *stored* (phenomenon). The knowledge of the phenomena represented in a model makes possible to find the adequate law to describe it: a *mass balance* (law) can be applied to state the property *mass of matter*. The syntax to represent this physical knowledge will be discussed in the last section.

189
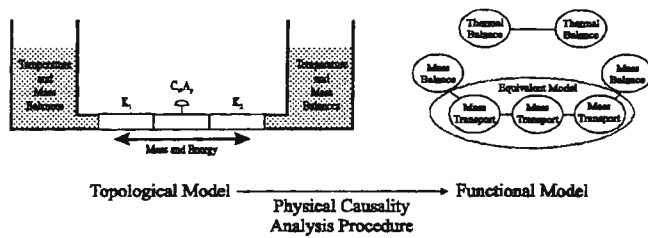
Figure 2 Physical causality analysis procedure.

This representation makes a first difference from object oriented modelling languages where behaviour is described in terms of mathematical equations. In the methodology depicted, the representation of physical knowledge can be stated afterwards as a set of equations, when the computational model is required. A second difference is the simulation model generation, where only those phenomena of interest will be considered, deriving efficient models suitable for the simulation goal. For instance, in a single tank with water model, only mass dynamics will be considered if the goal for the simulation model is to test a level controller.

Representation of system interaction in PML.

The interaction between connected submodels is defined in terms of entity exchange (for instance, matter or energy). This new definition of the model interface is independent of the variables needed to establish the relationships between the behaviour equations that compute the physical phenomena. Therefore, a different interface to describe interactions between models is needed. The concept of *port* has been introduced as an interface defining an exchange of an entity (for instance, matter or energy) [5].

This approach is guided by the fact that the development of basic component models should not be subject to the analysis of the context in which they will be reused. Whether ports defining *entity* exchange are allowed, connections between submodels will preserve system ⇔ model analogy describing the analogous connections between subsystems, regardless of the variables needed to mathematically represent the interactions.

The procedure of physical causality analysis.

The capability of reusing predefined models can be improved when it is possible to obtain physics phenomena and interactions analysing the model topology. The procedure of physical causality analysis examines the *topological* model deriving physical interactions from interconnected ports together with the phenomena described at each submodel. The derived behaviour and interactions are represented in the *Functional Model* (see Fig.2); a graph where nodes represent phenomena and the structure of edge connections represents the entity exchange paths derived from model topology.

In a first stage of the procedure, matching each model component declaration with the knowledge defined in the libraries checks consistence of topological model. The procedure analyses model structure in a top-down sense retrieving the topology of submodel interconnections and how ports define entity exchange. Then, each submodel is analysed to determine the laws describing the required phenomena. Applying to physical knowledge, the used relationships are: submodels represent entities and phenomena; the laws ruling the phenomena are found and, consequently, the involved entity properties can be described by these laws. By iterating this research algorithm, all the properties representation that is modular to the submodel can be stated. In the example at Fig. 2, the matter mass can be stated from the mass balance related to storage phenomenon and the matter exchange defined by ports. The remaining behaviour representation, which is not modular to the submodel, is instantiated from the coupling phenomena derived from port interconnections with other submodels. In the example, energy exchange between tanks (not explicitly described by ports) is deduced from the mass transport phenomena defined by the pipe lines and matter exchange ports. The resulting

190

functional model is used to achieve a simulation-oriented formalism. This translation procedure can be found in previous works [4,5].

Physics Knowledge representation

The information needed to make the physical analysis of a model built from the interconnection of predefined submodels (preserving system topology) is defined in the opened modelling libraries as instances of five main base classes: *phenomena, laws, models, entities* and properties.

Phenomena

Phenomena are defined inside models to declare model behaviour. The purpose of this declaration is to represent a phenomenon that occurs over an entity affecting to its properties. Defined as a class in the modelling libraries makes the link between the phenomenon and the law used to describe it. For instance, a mass balance rules the phenomenon of storing matter:

```
(phenomenon [stores(matter)] [mass_balance(matter)->mass(matter)]);
```

Laws

Describe how a property affected by a phenomenon can be quantified. Laws are instantiated when is necessary to represent the effects of a phenomenon declared in a model.

```
(law [mass_balance(matter)][integral(sum({+(input(matter)),-(output(matter)),
                                      +(generated(matter)),-(dest(matter))})])]);
```

Models

A model object represents the components (equipment) of a system. In a model can be defined the interaction with its environment (by means of mass or energy ports), geometric aspects of the component, which phenomena take place, the entities involved and/or the properties of interest. Inheritance is supported; all models in a library are hierarchically defined from a base class named base::model. Structured modelling is also supported, a model can be *atomic*, i.e. no submodels are present, or *structured* or *aggregated*, i.e. defined by the interconnection of submodels. Model declaration syntax is as follows:

```
(  model superclass
 [ model class instance ]
 [{{list of entities},
   {list of properties},
   {list of phenomena}
 }]);
```

The two first items define the *"IS A"* inheritance relationship. In model behaviour part (third item) phenomena, entities and their properties are represented. For instance, suppose that class reservoir is a model, in order to define a model for a tank which stores matter, declaration would be:

```
(reservoir [tank] [stores(matter)]);
```

Entities and properties.

Are used to define the entities involved in a system, for instance any kind of matter, and their properties. When an entity is declared inside a model it means that it is desired to focus the model behaviour into a particular entity. For instance, if tank_A stores water, declaration would be

```
(tank [tank_A] [matter(water)]);
```

In the modelling libraries, `water` has to be defined from `matter`, which in turn has to be hierarchically defined from `entity` super class. That allows the environment to check consistence of models during physical causality analysis. For instance:

```
(base::entity [matter] [{mass,temperature}]);
(matter [water] [density]);
```

Properties, the third item in entity declaration, are associated to an entity in order to define those aspects from the entity that are of interest. When an entity property is specified inside a model means that this property must be solved during physical causality analysis. For instance,

```
(tank [tank_A] [{matter(water),temperature(water)}]);
```

declaration specifies that the temperature of water is needed, even the case that `tank_A` precedence tree has no class declaring any thermodynamical behaviour. In such situations, physical causality procedure must find the physics relationships to achieve the required behaviour description.

Conclusions

The constrains derived from the computational causality analysis, used to achieve the simulation model in present object oriented modelling languages, have been discussed in this work. Modelling difficulties arise when behaviour is represented using an static equation oriented formalism since coupling phenomena and property propagation can not be statically predefined in a modular way. The analysis of models from a physical point of view is presented as an alternative to achieve a behaviour representation that can be dynamically modularised. A new language PML to represent physics knowledge has been designed to support the physical causality analysis procedure. PML allows an object-oriented methodology to develop opened libraries of models. New composite models are built up preserving system topology by the interconnection of predefined submodels. The result of the physical causality analysis procedure is the functional model, where dynamics of interest are represented in a modular way. An important consequence is that functional model represents adequate property coupling behaviour and can be translated into a simulation model avoiding the duplication of the phenomena description that leads to numerical singularities. Finally, a brief description of physics knowledge representation and language syntax has been presented.

Acknowledgements

References

[1] H.Elmqvist et Al. *Modelica™- A Unified Object-Oriented Language for Physical System Modelling*. September, 1997. http://www.dynasym.se/Modelica/Modelica1.html.
[2] F.E. Cellier and H.Elmqvist. Automated Formula Manipulation Supports Object-Oriented Continuous-System Modeling. In *IEEE Control Systems*, volume 13. 1993.
[3] Bernt Nilsson. *Object-Oriented Modeling of Chemical Processes*. Ph.d thesis, May 1993.
[4] Juan José Ramos. *Una metodología de modelado orientado a objetos para la representación del conocimiento físico*. Master's thesis, Universitat Autònoma de Barcelona, 1995.
[5] Juan José Ramos, Miquel Àngel Piera, and Ignasi Serra. The use of physical knowledge to guide formula manipulation in system modelling. *Simulation Practice and Theory*, 6(3). Pages 243--254, April 1998.
[6] Jean U.Thoma. *Simulation by bondgraphs:introduction to a graphical method* . Springer, 1990.

A Process Industry Pseudo-Bond Graph Tool to Avoid Algebraic Loops and Derivative Causalities Arisen From Model Coupling.

Daniel Luque[1], Miquel Àngel Piera[2], Ignasi Serra[3]

Unitat d'Enginyeria de Sistemes i Automàtica. Universitat Autònoma de Barcelona
Facultat de Ciències. Edifici C. E-08193, Bellaterra. Barcelona. Catalonia (Spain)
E-mail: [1]luque@sunaut.uab.es, [2]miquel@sunaut.uab.es, [3]ignasi@sunaut.uab.es

Abstract

A Bond Graph modelling tool is introduced. The tool has been conceived to generate an efficient computational model of a process industry flowsheet by analysing physical and computational causality arisen when reusing predefined acausal submodels. End users interact with the modelling protoype just by supplying a topological description of the system. The Bond Graph modelling technique is used to obtain the functional model describing physical phenomena which appear due to system interaction. The modelling tool generates and analyses automatically the bond graph model of a topologically described process to detect and remove algebraic loops and structural singularities caused by model interconnection, dealing in this way with an efficient computational model.

1. Introduction

Current Object Oriented modelling languages such as *Dymola* (and in an early future, *Modelica*[TM] [3]) support subsystems description in modular structures (objects) which can be linked in a similar way as subsystems components do in reality. In order to obtain the simulation model of a complex system described by its topological model together with a library of acausal models, a modelling environment must sort out the computational causality problem. Most of the present modelling tools [2,6] handle this problem with symbolic manipulation of the acausal submodel equations trying to deal with an external compatible information flow through the overall model simulation equations. Unfortunately, when computational causality is solved by analysing the model interconnection (shared variables) instead of analysing the subsystem interaction (phenomena description), two well-known problems sometimes might arise: algebraic loops and structurally singular models.

Although there are numerical methods to sort out these problems, the simulation models generated in this way are far to be considered efficients. The fact that a modeller could solve these problems by analysing the overall system from a physical causality point of view encourages to develop a modelling tool which might avoid algebraic loops and the derivative causalities which appear due to model connection without constraining its use to a specific application field.

The inclusion of physical knowledge in a modelling environment (such as PMT [7]) provides a logical approach to determine system phenomena which can not be described in a particular submodel element since they appear due to subsystem interaction. As a counterweight to physical knowledge incorporation into a modelling environment a constraint of its use to a particular application field arises (e.g., PSPICE[8] electric domain, MMS [9] fossil and nuclear power plants). A modelling tool which incorporates physical knowledge to determine physical phenomena without constraining its application field is presented in this paper.

Section 2 introduces the functional model representation, which results from a physical causality analysis of the overall system. A Bond Graph based modelling is presented as a good methodology to deal with functional models. Section 3 describes the bond graph manipulation (reduction) carried out in order to detect and solve algebraic loops and structural singularities. Section 4 introduces the implemented tool. Model data structure description together with bond graph based modelling tool is illustrated in this section. Finally, section 6 presents the conclusions of the present work

---

2. Functional model

Present Object Oriented modelling environments supports physic components representation through objects abstraction, trying to ensure that the system⇔model analogy is maintained, and allowing, in this way, the users to define the topological model of the system just by interconnecting objects in the same way as real system components do. Modelling environments carry out mathematical model generation from topological information and submodel libraries.

Despite acausal models are formulated under a context independent formalism, coupling equations generated to describe system interations do not describe physical phenomena in a proper way. Phenomena are not inherent to one singular element. A multiple definition of a phenomenon may appear into system model when the phenomenon is defined into more than one submodel.

Functional models [5] provides information about system physical phenomena which appears when components are connected. Since phenomena knowledge can not be previously modelled inside individual components to deal with the functional models, a physical analysis must be performed over topological model to find out all the physical phenomena system.

Bond Graph

Although physical system interaction is specific to each system domain, it is necessary to carry out analysis in a higher order level abstraction to determine system phenomena without constraining the modelling tool to a particular domain. Bond Graph offers a good platform to describe system interaction independently of the domain as mass or energy flow[4]. Effort-, flow-variables have different meanings depending on the application domain: voltage-current in electric circuits, pressure-volume flow rate in hydraulics, etc. However, it is not necessary to know the meaning of these variables to deal with BG analysis. For example, augmentation procedures do not take care if effort-flow variable pairs are either *voltage drop-current* pair or *pressure drop-volume flow rate* pair. Hence, BG is a good modelling technique which can represent mass or energy exchange phenomena of functional models. Furthermore, bonds network of a Bond Graph describe the energy flowing between system elements, so physical phenomena which appeare as result of components connection is supported.

3. Bond Graph reduction to Functional Model

Sometimes, the BG augmentation property of computational causality determination can lead to *false* singularity problems when working over BG derived from interconnected submodels. *False* singularities are expressed in the sense of singularities caused by submodels interconnection which can be solved through symbolical manipulation, as it is shown below. To illustrate this problem, let us consider the following example (figure 1): suppose three tanks connected through four pipes.
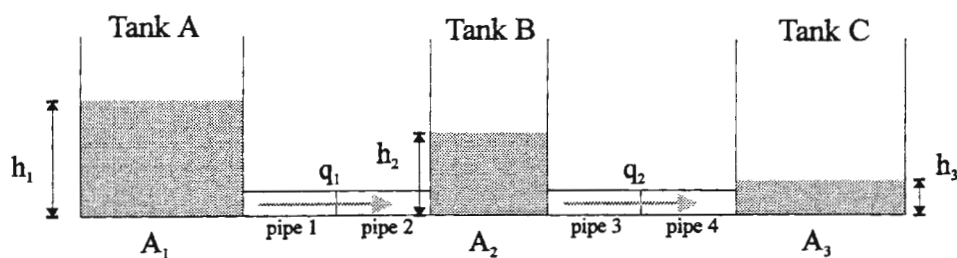
Figure 1. Three tanks connected by pipes.
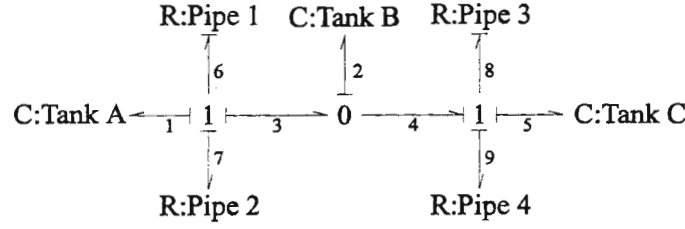
Figure 2. Bond graph of three tank system.

It is easy to find out that any of two pairs pipes (either pipe1-pipe2 or pipe3-pipe4) in serial connection may cause an algebraic loop when generating the computational model. Physical analysis of the BG shows that tank A and tank B interaction leads to a mass transport phenomenon, which in equation oriented modelling tools is described twice: once by pipe1 and again by pipe2 (the same problem arises for pipes 3 and 4, between tanks B and C). The problem arisen from submodel interconnection can also be observed over the augmented BG model (figure 2): in order to compute the effort-variable of R2 (pressure drop P at connecting point of pipe 1 and pipe 2), the flow-variable of R1 (volum flow rate q_1 which passes through pipe 1) is required which in turn requires effort-variable of R2 to be calculated (eq.1 and eq.2):

$$q_1 = Cv_1\sqrt{P_A - P} \qquad\qquad \text{(eq.1)}$$

$$P = \left(\frac{q_1}{Cv_2}\right)^2 + P_B \qquad\qquad \text{(eq.2)}$$

where P_A and P_B are pressure drop at tank A and Tank B, respectively. Cv_1 and Cv_2 are constants. The same dependency exists between R3 and R4. Note that when the modelling task is hand made by an expert the problem can be conveniently overcame by symbolic manipulation of acausal equation, describing the mass transport phenomenon as if there were only one pipe (eq.3). The resulting mixed pipe model collects the required characteristics of both pipes.

$$q_1 = \frac{(Cv_1{}^2 + Cv_2{}^2)}{(Cv_1 Cv_2)^2}\sqrt{P_A - P_B} \qquad\qquad \text{(eq.3)}$$

The BG model describes mass transport flow phenomena and energy exchange phenomena from which it is possible to determine computational causality. Computational causality assignment can lead to algebraic loops and singularities due to a multiple redefinition of the same phenomenon. Physical analysis is performed by the implemented tool in order to find out and solve algebraic loops and singularities which have been caused by the multiple definition of a phenomenon. The modelling tool tries to manipulate those problems in a similar way as a modeller would do: mixing (and reducing) the conflictive elements. The task of finding possible algebraic loops and structural singularities can be performed through the zero-order causal path definition introduced in [1].

Currently, the tool sorts out only a particular kind of algebraic loops: resistor elements which form an algebraic loop and are connected through either a 0- or a 1-junction. Other algebraic loop shapes are being studied together with structural singularities conflicts to be solved in an early future.

4. Implemented Tool

The tool has been conceived in order to help engineers generate efficient computational models of process industry systems, providing submodels libraries for model reusability automatically. Object Oriented Programming has been used when developing the tool, where Objects structures have been implemented to

195

carry out the different stages which represent the use of the tool. Figure 2 shows a scheme of tool functionality.
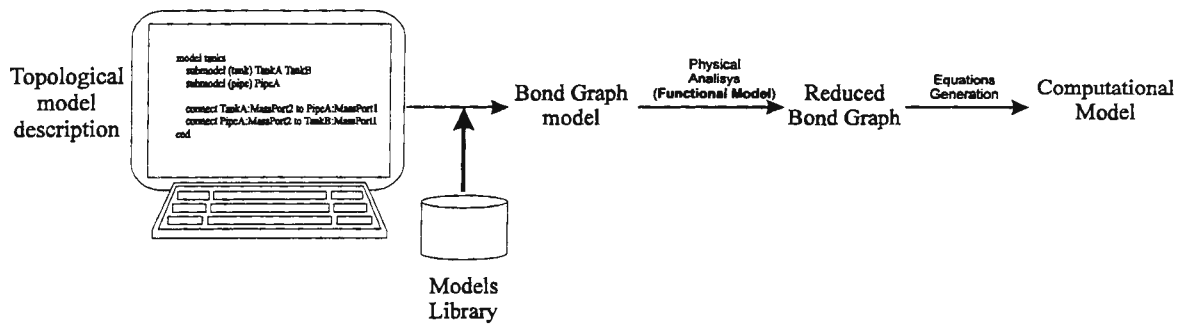
Figure 3. Tool functionality scheme.

Computational model generation evolves throughout a sequence of five steps:

- *First step*: The generation process begins by the specification of the topological model. This model definition involves submodels connection like the real system do.
- *Second step*: Submodels equation and BG information of the overall system is obtained from the topological models and submodels libraries. The BG is generated according to the connection of ports among submodels.
- *Third step*: BG model is analysed to determine physical phenomena. Algebraic loops and structural singularities associated to each phenomena are detected and (if possible) sorted out.
- *Fourth step*: A reduced BG is attained from physical analysis. Note that this reduced BG is generated only if some problems detected on third step are overcame.
- *Fifth step*: This final step involves computational causality determination and computational equations generation. In this step, a specific source program code could be generated for existing simulation packages.

Note that in case that a BG is reduced some system information is lost. For example, two pipes in serial connection describe the same mass transport phenomenon. When both pipes are mixed to remove the phenomenon the resultant mixed pipe does not take account some internal variables described in each pipe (pressure drop in the connection point between pipes, for example). In order to allow the access to the original system information when the model is simulated, the mixed components are held and required relations are build among the reduced BG elements and the mixed ones.

Library definition and Topological model specification

Library submodels are declared through text files using a predefined format. As an example, consider the model file of a tank with an incoming and an outgoing flow (describing only mass transport):

```
Model type tank
     Cut MassPort1 (Pi/Qi)
     Cut MassPort2 (Po/Qo)
     Parameters section rho g
     BGModel MassPort1->0:J0 0:J0->C:C0 0:J0->MassPort2
     BGEquations C0(e:Po;f:Qi-Qo)
     Der(m)=rho*(Qi-Qo)
     m=rho*V
     h=V/section
     Po=rho*g*h
End
```

The key word *CUT* declares the submodel ports, in this case mass ports where effort-, flow-variables are pressure (P) and volume flow rate (Q). *BGModel* defines the Bond Graph information of the submodel. Figure 3 shows the Bond Graph information described in the preceding component file. *BGModel* defines three bonds:

```
MassPort1->0:J0
0:J0->C:C0
0:J0->MassPort2
```

Note that the ports connection are specified. Bond elements are conveniently named: `C:C0` indicates a capacitive element with the identifier `C0` inside the submodel. Arrow direction indicates bond direction. Note that in this example the system component (tank) has been described by a unique equivalent bond graph element (a capacitiy C0), but this analogy is not preserved at all.
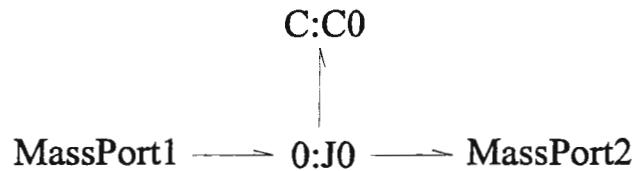
Figure 4. Bond Graph of a tank submodel.

Finally, constitutive relations among effort-, flow- variables of C-, R-, I-, SE-, SF- elements are defined in line *BGEquations*. In the example only capacitive constitutive relation is declared through:

```
C0(e:Po;f:Qi-Qo)
```

Note that the element identifier used in *BGModel* is referenced in *BGEquations*. Effort-variable of capacity `C0` is declared as the pressure drop at certain position in the tank. On the other hand the flow-variable is defined as the difference between incoming and outgoing volume flow rate.

The topological system description consists of information such as the one shown bellow through an example of three tanks connected by four pipes:

```
Model ThreeTanks
        Submodel (tank) TankA TankB TankC
        Submodel (pipe) Pipe1 Pipe2 Pipe3 Pipe4
        Connect TankA:MassPort2 to Pipe1:MassPort1
        Connect Pipe1:MassPort2 to Pipe2:MassPort1
        Connect Pipe2:MassPort2 to TankB:MassPort1
        Connect TankB:MassPort2 to Pipe3:MassPort1
        Connect Pipe3:MassPort2 to Pipe4:MassPort1
        Connect Pipe4:MassPort2 to TankC:MassPort1
end
```

This information is structured into two sections: the first section declares the submodels used to build the model and second section interconnects each component ports. Note that is not necessary to connect all the ports of a submodel. When the final BG is generated, the bonds associated to disconnected ports are not considered.

Bond Graph generation and study

The implemented package maintains a topological model information structure together with submodel library information from which BG model is created. It is important to note that is not necessary that a component has all its ports connected. So, when deriving the BG model from topological information, only the bonds that describe energy exchange among connected ports are considered.

Object Oriented programming has been used to manipulate as topological information as BG model. A *bondgraph* class has been created to handle BG information, which in turn contain a list of objects that represent BG elements. Furthermore, a hierarchy of classes has been created to represent each BG element: C-, I-, R-, SE-, SF-, TF- and GY- elements, 0- and 1-juncitons and bonds itself. Each element is conveniently connected, so that the element knows: to which elements are connected, which is the causality assignment, the power direction, etc.

Implementation aspects

This tool is being developed on *Matlab 5*, which offers object oriented programming capability. Furthermore, the *Maple* kernel has been used for symbolic manipulation of equations and for graph treatment. Thus, *Matlab 5* has allowed to create the objects structure necessary to maintain all the model information (topological information and bond graph model) meanwhile *Maple* has helped in detecting algebraic loops and structural singularities, and in symbolic manipulation of equations when mixing conflictive elements.

5. Conclusions

A modelling environment based on Bond Graph modelling technique has been presented. The developed tool incorporates physical analysis of BG models to determine the phenomena present in the system. This physical analysis allows to reach a functional model which can be used to detect and sort out problems arisen from submodel coupling, such as algebraic loops and structural singularities through BG model reduction (mixing conflictive elements).

Currently some kind of algebraic loops shapes can be overcome but the power of the tool in this sense is to be augmented in an early future. Submodels definition is also under improvement, so *BGEquations* information should be eliminated from model specification.

6. Acknowledgement

References

[1] Dijk, Johannes van. *On The Role Of Bond Graph Causality In Modelling Mechatronic Systems*. Thesis. Enshede, The Netherlands. 1994.
[2] H. Elmqvist. Object-Oriented Modeling and Automatic Formula Manipulation in Dymola. In SIMS'93, Scandinavian Simlation Society, July 1993.
[3] H. Elmqvist, F. Boudaud, J. Broenink, D. Brück, T. Ernst, P. Fritzson, A. Jeandel, K. Juslin, M. Klose, S. E. Mattsson, M. Otter, P. Sahlin, H. Tummescheit, H. Vangheluwe. *Modelica – An Unified Object-Oriented Language for Physical Systems Modeling*. http://www.Dynasim.se/Modelica/Modelica1.html, September 1997.
[4] Karnopp, D and Rosenberg, R. *Analysis And Simulation Of Multiport Systems. The Bond Graph Approach To Physical System Dynamics*. The MIT Press, Cambridge, 1968.
[5] J.J. Ramos, M.A. Piera and I. Serra. *A Modelling Tool to Guide Computational Causality Assignment Through Physical Causality Analysis*. In Eurosim, Vienna, Sept. 1995.
[6] Sven Erik Mattsson and Mats Andersson. *Omola – An Object-Oriented Modeling Language*. In M. Jamshidi and C. J. Herget, editors, Recent Advances in Computer Aided Control Systems Engineering, pages 291-310, Elsevier Science Publisers, 1993.
[7] M.A. Piera and César de Prada. *Modeling Tools In The Continuous Simulation Field*. Simulation 66:3, 179-189, Simulation Councils Inc., March 1996.
[8] MicroSim Corp. *Pspice User's Manual, Irvine*, California. 1987.
[9] EPRI. *Modular Modeling System (MMS): A Code For The Dynamic Simulation of Fossil and Nuclear Power Plants*. Report: CS/NP-3016-CCM, Electric Power Research Institute, Palo Alto, California. 1983

MODELING AND OPTIMAL DESIGN OF A CENTRAL SOLAR HEATING PLANT WITH HEAT STORAGE IN THE GROUND USING MODELICA

R. Franke

Technical University of Ilmenau, Kirchhoffstr. 1, D-98693 Ilmenau, rf@rz.tu-ilmenau.de

Abstract

The paper discusses the modeling of a central solar heating plant with seasonal storage in the ground using the new object-oriented physical systems modeling language Modelica. Main emphasis is put on the hierarchical decomposition of the system model and on the re-engineering of an existing Fortran code for the ground store model. The object-oriented physical system model is compiled to a mathematical description in the form of ordinary differential equations (ODE). The ODE model is used to formulate and solve nonlinear optimization problems. We show design optimization results obtained for given weather conditions in Switzerland. The heating system is designed to cover the annual load of a housing area for space heating of about 500 MWh to 95% by solar. The optimization results have been validated with the help of initial-value simulations using TRNSYS, a special-purpose simulation software for thermal energy systems.

1 Introduction

Seasonal heat storage in the ground is an important opportunity for a more efficient use of energy. The idea is to store thermal energy when it is available and to use it when it is needed. Many different thermal processes and their mutual interaction have to be considered, if a ground heat store is integrated with energy systems. Advanced design tools are required in order to obtain cost-effectiveness. Related system studies are normally done with the help of detailed computer simulations.

A widely used special-purpose modular simulation software for solar energy systems is TRNSYS. Often hundreds of initial-value simulations with varied parameter values of the system components have to be performed in order to optimize a system design, see e.g. [7]. The total number of needed calculations strongly increases with the number of free parameters. That is why the approach is no longer practicable for complicated optimization problems. On the other hand, the successful application of mathematical optimization procedures to complex system models is burdened with two things: a mathematical optimization problem has to be formulated and an appropriate solver is needed. The use of the new object-oriented physical systems modeling language Modelica [1] and of an advanced optimization solver [3] simplify the application of mathematical optimization to detailed models of dynamic systems considerably.

This paper discusses, how the features of the general-purpose physical systems modeling language Modelica can be exploited to implement a complex heating system model efficiently. The model is used to solve a design optimization problem. As an example we treat a design case from [7]. In this way we are able to compare our results with the comprehensive TRNSYS simulations done there.

2 The System Model

Figure 1 shows the structure of the system, whose model implementation in the object-oriented language Modelica is discussed in this section. The system can be divided into three subsystems: a
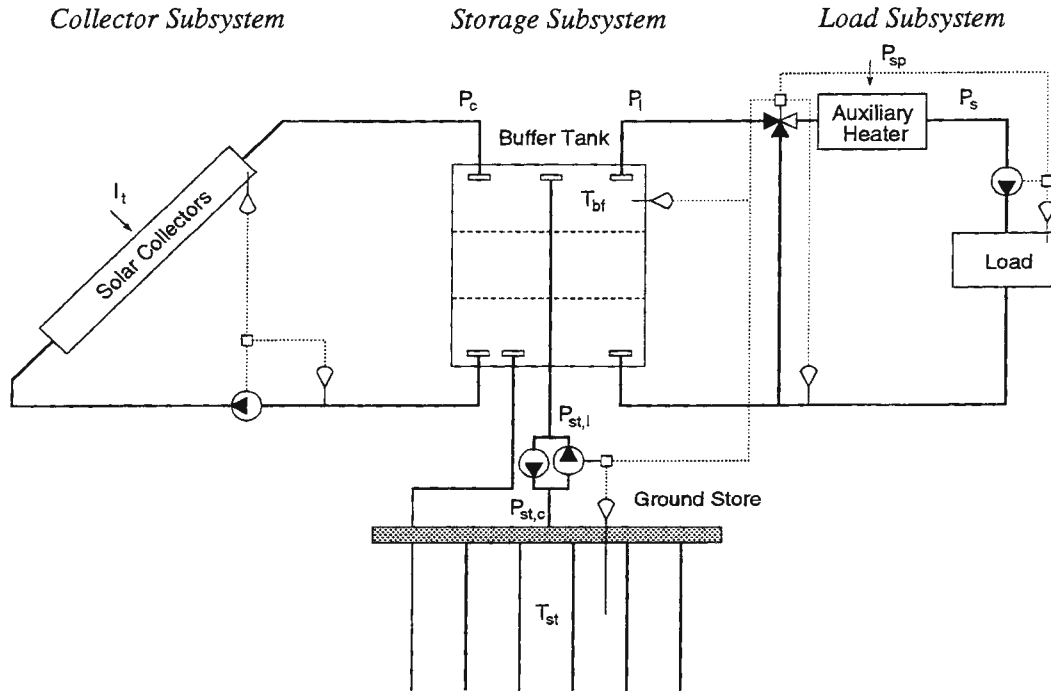
Figure 1: Structure of a central solar heating plant with seasonal duct store.

collector subsystem, a storage subsystem, and a load subsystem. Each subsystem is further subdivided, e.g. the store into a buffer tank and a duct ground store. The system is intended to collect and store solar energy during summer time and to use it for space heating in a housing area during winter time. An auxiliary heater supplements the solar energy in order to fully cover the load.

2.1 Interfaces

First we define interfaces and an abstract system model. The according Modelica code is listed in Figure 2. We are mainly interested in the energy balance of the system. Accordingly a HeatingConnector for the interconnection of several components is modeled by using variables and parameters that are needed to describe the transport of energy with a heat carrier fluid. That is the fluid temperature T, the mass flow rate q_{m}, and the specific heat capacity of the fluid c_p. The used types are predefined in the standard package Modelica.SIunit.

Many components have two connectors, one inflow and one outflow. Examples are the collector model and the load model. A HeatingTwoPort serves as common base class. The storage model has four ports: two for the collector loop and two for the load loop. Using the base classes LoadModelShell, CollectorModelShell, and StorageModelShell for the three main subsystems, the abstract solar heating model SolarHeatingModelShell can be defined. The specialization of the system components is discussed in the following subsections.

2.2 The Duct Ground Store

The model described here is based on the DST model [6], which is currently available as Fortran and Pascal version and which has a size of about 100 KB source code. Using the high-level features of Modelica, an efficient re-engineering of the model has been possible. The new implementation is divided into several subcomponents to consider convective and conductive thermal processes in the ducts and in the store, respectively.

```
connector HeatingConnector                partial model SolarHeatingModelShell
  Temperature           T;                   LoadModelShell lm;
  flow MassFlowRate     qm;                   CollectorModelShell cm;
  SpecificHeatCapacity  cp;                   StorageModelShell sm;
end HeatingConnector;
                                           equation
partial model HeatingTwoPort                 // load loop
  HeatingConnector a, b;                      connect(sm.bl, lm.a);
end HeatingTwoPort;                           connect(lm.b, sm.al);

partial model LoadModelShell                 // collector loop
  = HeatingTwoPort;                           connect(sm.bc, cm.a);
                                              connect(cm.b, sm.ac);
partial model CollectorModelShell          end SolarHeatingModelShell;
  = HeatingTwoPort;

partial model StorageModelShell
  HeatingConnector al, bl "load loop";
  HeatingConnector ac, bc "coll loop";
end StorageModelShell;
```

Figure 2: Connector and model interface definitions for a solar heating system model

In a duct store, an array of boreholes is drilled vertically into the ground. Water is used as heat carrier fluid and is circulated through these ducts in order to exchange heat. In the model, the store is divided into a number of sections (6 in our case). One representative duct component is defined for each section.

The main model complexity is caused by the spatially distributed thermal process in the ground, which can be described for constant thermal properties by the heat equation

$$\frac{1}{a}\frac{\partial T}{\partial t} = \nabla^2 T \tag{1}$$

with T the ground temperature and a the thermal diffusivity. The method of finite differences is a famous tool to treat partial differential equations (PDE) of the form (1) numerically by calculating the temperature field at several grid points. However, the number of temperature variables introduced in this way can be very high. In order to keep the model complexity as small as possible, we first divide the total thermal process in the ground into two subprocesses: a small-scale local thermal process around each duct and a large-scale global thermal process in the overall store. Each subprocess is treated separately. The total thermal process is given by superposition of the subprocesses.

One component for the local thermal process is defined for each section of the storage model and is attached to the corresponding duct component. Considering the radial temperature distribution around the duct during heat injection or extraction, the model can be implemented by a radial, one-dimensional mesh of ground cells (e.g. 10 cells per component). But after a comparable short time of operation (some days) it can be seen that the shape of the local temperature field will become steady. Furthermore the small-scale dynamics in the ground is damped by the buffer tank. Neglecting the time variation, the dynamic model for the local thermal process can be replaced by an analytical description of the local temperature field under steady-flux conditions. Accordingly the heat transfer is assumed here to occur via time-invariant thermal resistances between the ducts and the ground store volume. The resistances are given by a logarithmic function of the duct spacing, see [5].

The large-scale thermal process in the ground is modeled by a rectangular mesh of a few hundred ground cells. In system simulations we are mainly interested in the interaction between the defined

storage sections and the resting system. The actual temperature distribution in the ground, as approximated by the mesh, is of less importance. That is why we are applying model order reduction to the state-space model describing the temperature field in the ground. In the example discussed here, the number of temperature variables could be reduced from 220 to 15 by applying a singular perturbation approximation, see e.g. [8].

2.3 The Water Tank

The buffer tank is modeled by three uniformly mixed water layers. Each layer is described by one temperature variable and an energy balance equation taking into account water inflows, water outflows, and conductive heat transfers with neighboring layers and with the environment. The natural convection inside the tank is modeled by nonlinear conductances between neighboring layers using an arc tangent function. Variations of the buffer tank volume influence heat capacities and heat losses of the layers.

2.4 The Solar Collector Array and the Heat Load

In order to simplify this study, we use TRNSYS as preprocessor together with the solar collector subsystem and the load model developed in [7].

The solar collector subsystem is simulated for a number of constant fluid temperatures over the year. Afterwards the specific gained energy and the circulated water amount are approximated for periods of 3 hours by cubic polynomials using the fluid temperature as free variable. The total collector area is used as multiplier.

The heat load as well as the fluid forward and return temperatures in the load model are calculated as functions of the ambient temperature.

3 System Optimization

With the help of the object-oriented modeling tool we automatically compile a mathematical system model of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}[t, \mathbf{x}(t), \mathbf{z}(t), \mathbf{p}], \quad \mathbf{x}(t_0) = \bar{\mathbf{x}}(t_0). \tag{2}$$

The state variables $\mathbf{x} \in \mathbb{R}^{n_x}$ are 24 temperatures and 6 energy quantities. An additional state is used to integrate violations of a given limit by the buffer temperature. The disturbances $\mathbf{z} \in \mathbb{R}^{m_z}$ are the weather conditions during the year. The time-invariant parameters $\mathbf{p} \in \mathbb{R}^{m_p}$ are the collector area, the buffer tank volume, the duct store volume, and the duct spacing. The model is translated into C-code as it can be imported by the applied optimization tool.

The optimization problem is to find the minimum-cost system design that fulfills a given solar fraction, i.e. that supplements a given fraction of the total heat load by solar, during the operational period $[t_0, t_1]$ of one year. The system costs are described by the cost function

$$J[\mathbf{x}(t_1), \mathbf{p}] \to \min_{\mathbf{p}}, \quad J : \mathbb{R}^{n_x} \times \mathbb{R}^{m_p} \mapsto \mathbb{R}^1, \tag{3}$$

which includes prices for the solar collector array, the buffer tank considering cost decreases for larger volumes, and the duct store considering e.g. drilling costs, land area costs and piping costs. Furthermore we added a term to consider the energy needed for the thermal build-up process of the ground store during the first years of operation. We are interested in a solution that minimizes J subject to the system model behavior (2) and additional constraints of the form

$$\mathbf{c}_\mathrm{l} \leq \mathbf{c}[\mathbf{x}(t_0), \mathbf{x}(t_1), \mathbf{p}] \leq \mathbf{c}_\mathrm{u}, \quad \mathbf{c} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{m_p} \mapsto \mathbb{R}^m. \tag{4}$$

One final state constraint results from the solar fraction. Other constraints are restrictions on the design parameters and on the state trajectories, e.g. to avoid boiling. Moreover we do not know initial values for the 24 storage temperatures $\bar{x}_T(t_0)$ at the beginning of the year. They are the result of the thermal build-up process. That is why the initial temperatures are not fixed, but they are defined with the help of the equalities

$$\bar{x}_T(t_0) = x_T(t_1).$$ (5)

The solution we are specifying in this way is called the steady-state operation of the system.

The nonlinear, constrained optimization problem (2)–(4) is solved numerically with the sequential quadratic programming (SQP) software Omuses [3]. The optimization algorithm requires sensitivities of the system equations (2), the criterion (3), and the constraints (4) with respect to the parameters and the unknown initial states. They are calculated internally by the solver by using automatic differentiation. The system equations (2) and their extension by sensitivity equations are solved numerically with the procedure RKsuite, which is available in the Netlib.

4 Case Study
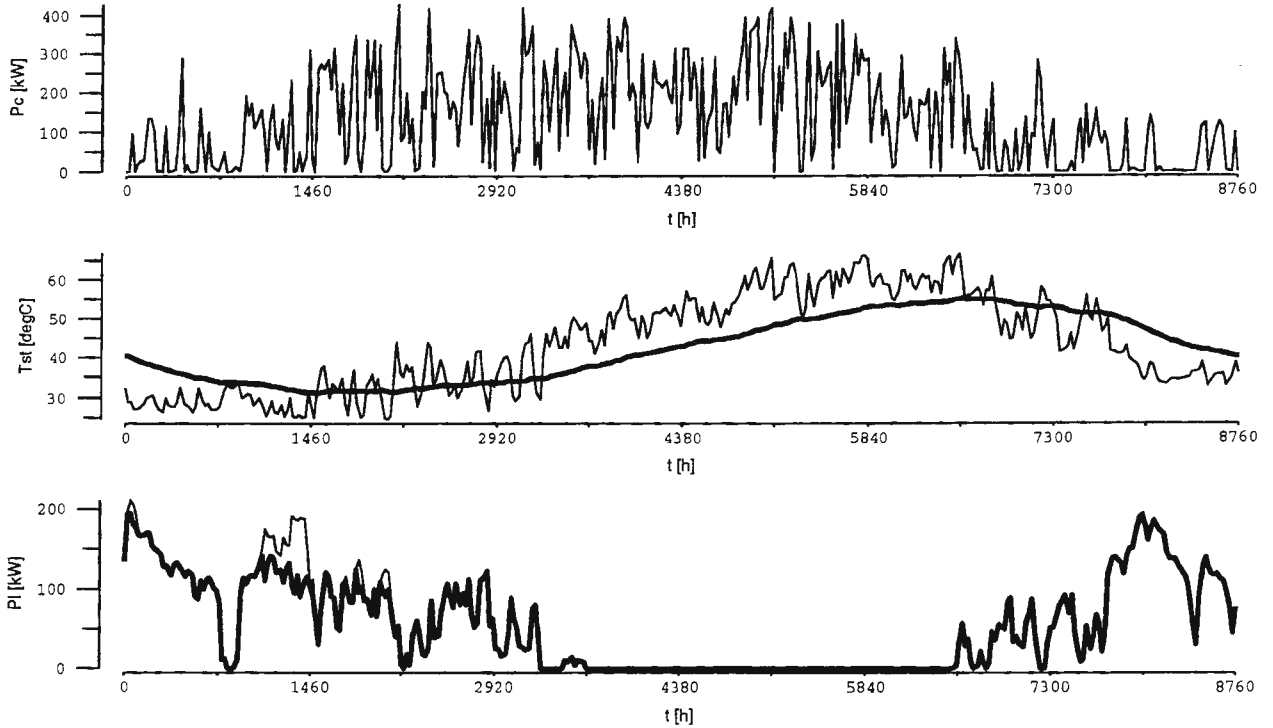
Figure 3: Simulated daily mean values for a solar fraction of 95% in steady state operation. The diagrams show the collected solar heat P_c, the average temperatures T_{bf} of the buffer tank and T_{st} of the ground store (thick), as well as the heat load P_s and the solar contribution P_l (thick).

We apply the methodology to the low temperature case of the study [7]. The system has a total annual heat load of 500 MWh corresponding to 120 housing units. The heat is only used for floor space heating. Weather data and prices for Geneva, Switzerland are assumed.

In [4] we have discussed optimization results for a solar fraction ranging from 40% up to 100% assuming 25 years of operation and an annuity factor of 0.1. There we used a version of the model formulated in the modeling language Omola. However, the compiled ODE-model is identical to the

one used here. According to the results, there is almost no cost increase for solar fractions up to 95%. The solar cost is less than 325 CHF per MWh.

Figure 3 shows the steady state behavior of the system for a solar fraction of 95%. The seasonal heat store has to bridge the gap between availability of solar energy during summer time and the need for space heating during winter time. It can be seen, how the large ground store and the small buffer tank inter-operate in order to fulfill this task efficiently. The supplementary heat is mainly needed in February. The optimal system parameters have been validated with the help of TRNSYS simulations.

5 Conclusions

Modern algorithms can considerably improve the efficiency of computer simulations. Here we successfully solved a design optimization problem for a sophisticated system model. Other areas of application of the method are nonlinear parameter estimation and optimal control, see e.g. [2].

The modeling done in this study is concentrated on seasonal heat storage. The use of the general-purpose high-level modeling language Modelica and the exploitation of automated symbolic processing of model equations simplify the model development and allow the efficient application of recently developed mathematical algorithms to complex dynamical system models.

The main goal of the re-engineering done for an existing Fortran code of the ground store model has been to reduce the model complexity in system simulations. New features introduced in the Modelica version are the description of small-scale thermal processes by analytical expressions and the application of model order reduction to the state space model of the large-scale thermal process in the ground. Such modifications can be expressed in Modelica by replacing submodels with alternative implementations. As the treatment of PDEs is not yet covered by Modelica, a model generator program has been developed that generates a numerical mesh and performs the model order reduction.

References

[1] Hilding Elmqvist and Sven Erik Mattsson. Modelica, the next generation modeling language — An international design effort. In *Proceedings of the 1st World Congress on System Simulation*, Singapore, September 1997.

[2] R. Franke. Object-oriented modeling of solar heating systems. *Solar Energy*, 60(3/4):171–180, 1997.

[3] R. Franke and E. Arnold. Applying new numerical algorithms to the solution of discrete-time optimal control problems. In K. Warwick and M. Kárný, editors, *Computer-Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*, pages 105–118. Birkhäuser, 1997.

[4] R. Franke and G. Hellström. Optimization of solar heating systems with seasonal storage in the ground. In *Megastock '97, 7th Int. Conference on Thermal Energy Storage*, volume 1, pages 527–532, Sapporo, Japan, June 1997.

[5] G. Hellström. *Ground heat storage*. PhD thesis, Lund University, Sweden, 1991.

[6] G. Hellström, L. Mazzarella, and D. Pahud. Duct ground heat storage model. Lund-DST. TRNSYS 13.1 version 1996. Techn. report, Dept. of Mathematical Physics, Lund University, Sweden, 1996.

[7] D. Pahud. Simulation of central solar heating plants using a duct store: an application for Switzerland. Techn. report, Dept. of Mathematical Physics, Lund University, Sweden, July 1996.

[8] R.V. Patel, A.J. Laub, and P.M. van Dooren, editors. *Numerical Linear Algebra Techniques for Systems and Control*. IEEE Press, 1994.

Simulation of Radiowaves Propagation Using Neural Networks[1]

Michal Steuer and Miroslav Snorek

Czech Technical University in Prague, Faculty of Electrical Engineering,
Department of Computer Science and Engineering, Karlovo Namesti 13, Praha 2, The Czech Republic,
Steuer@sun.felk.cvut.cz

Abstract: This article first summarises some basic problems in the field of calculation of radiowaves propagation for land mobile services. New approach to the solution of this problem using artificial neural networks is proposed here. Two examined networks are shortly introduced at first. The adaptation of these networks to the solved problem is shown. Practical results of the usage of these networks are described and analysed.

Keywords: Neural Networks, Electromagnetic Field Propagation, Land Mobile Services, Backpropagation, GMDH.

Introduction

Simulation of the propagation of electromagnetic field is a very complex task which is demanded to be solved for many applications. The interest in finding the best possible solution of this problem grows with the boom of cellular networks which are used to build mobile telephony networks or radio networks for emergency services.

a) predicted by GMDH neuronal network b) clean map for comparison   c) predicted by Backpropagation n. n.
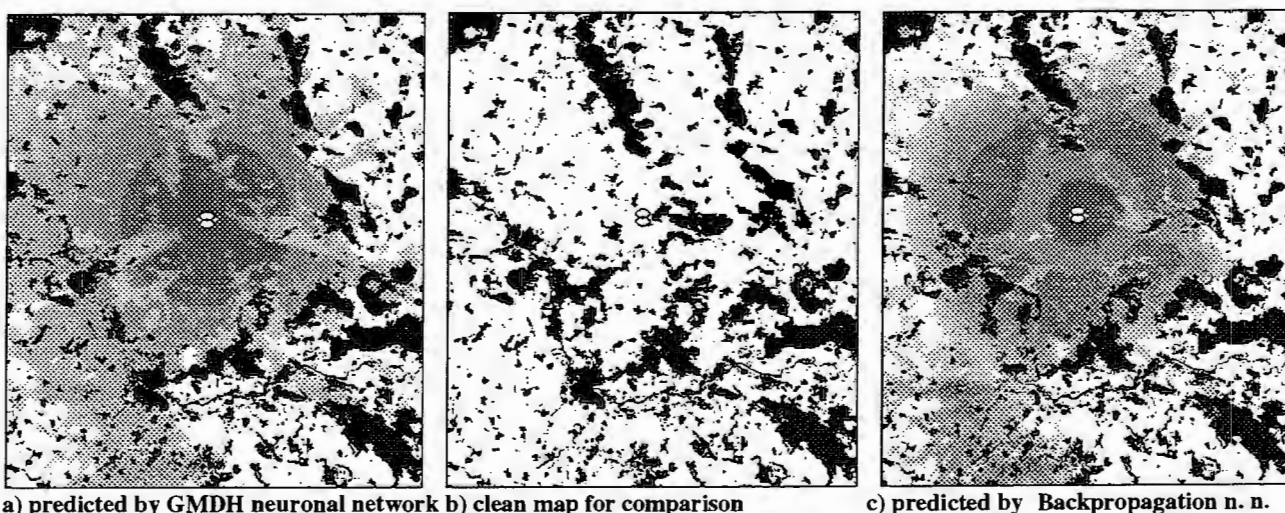
Figure 1: Strength of electromagnetic field predicted by both the neural networks, and a clean map for comparison. Towns, forests and rivers are drawn by the darkest colours. The strength of the field is distinguished by shades of grey drawn in a layer above the map. So the background map shines through only if the field strength is lower than a weakest threshold. The transmitter was situated on a 40 meters high tower and the power was radiated by an omnidirectional antenna.

A cellular network consists of several base stations (BS) located on fixed positions, and many mobile stations (MS) moving among them. To can a MS operate, it needs a good radio connection to at least one BS. While building a radio network, the designers need to know which area will be serviced (or covered) from a given BS. The serviced area is determined by a minimal strength of electromagnetic field on which a MS can operate, and by radio parameters of the BS like radiated power, antenna type and position. In all this article we only investigate the path from BS to MS, that is so called *downlink*. So BS is supposed to be only a transmitter and MS is supposed to be only a receiver. Most of the services use duplex communication but for simplicity we can say that both the paths are reciprocal. A typical serviced area is depicted in figures 1a and

---

1c. The background of the figure is a map and the serviced area is drawn in shades of grey. So the places where the background map is visible are supposed to be uncovered, without any possibility to make the radio connection to the BS. The prediction of radiowaves propagation, by which the serviced area in the figure is determined, was performed by a neural network described later.

At first sight, the simulation of propagation seems to be very easy task. We simply need to calculate the loss of the path between BS and MS for all possible positions of MS in a given area. One of the problems is that at least one of the two radio stations is situated in a very low height, typically 2 or 3 meters above the ground, so the propagation of the electromagnetic field is substantially influenced by the surface of the ground like hills, buildings and vegetation. Therefore calculation of only free space propagation, which works good for e.g. TV broadcast where the antennas are placed very high, does not have enough accuracy here. A three-dimensional digital terrain model (DTM) is needed to improve the accuracy.

Short review of classical algorithms

There are several commercial computer programs on the market which solve this problem. They use digital terrain models to take the surface of the ground into account. Two main approaches to the simulation of radiowaves propagation can be distinguished:

• Most of the commercial programs use Okumura-Hata prediction model [1], [2]. This model is based on empirical formulas obtained by statistical evaluation of measurements done in 50's in Japan. The model distinguishes different kinds of the areas where MS is situated: open area, forest, suburban and urban. The basic model is sometimes improved by e.g. calculation of diffraction loss on the biggest obstacles or by line of sight checking. But typically, the model evaluates only the direct path between BS and MS. Propagation models of this kind are also called *statistical* as they are more based on statistics than on physics.

• The electromagnetic signal can, however, get to the mobile station by various ways. It can be reflected by the walls or hills, and on the corners of buildings diffractions appear. Because of this some other approach in the calculation was introduced a few years ago. Models of this class use so called ray-tracing. Its basic principle is similar to the ray-tracing used in computer graphics. They try to find the most important paths from BS to MS, calculating signal attenuation and its phase in the point of the MS. By adding all the contributions, with regards to the phase, the signal strength in the place of the MS is calculated. These models are also called *multi-ray* or *deterministic* for their clear mathematical background.

Why neuronal propagation predictor?

This article proposes a new approach how to solve this problem. We use the advantage of the robustness and the learning capability of artificial neural networks to improve the accuracy of the classical methods of prediction. While creating the classical models the inventors had to find out complicated models of dependency of different kinds of obstacles on the resulting path loss. This effort is visible mainly in the case of the deterministic models. Because of its basic features, the neural network is able to create its own internal model of behaviour of radiowaves by just observing measured values. This happens during so called training phase. This internal model can be later used for predicting these values in the places where the measurements were not performed. This is called recall phase. In other words we can say that the network is able to generalise the observed measured values.

The neural network approach has good presumptions to work better than the empirical formulas of the classical models, as the network can take into account much more details of the propagation path. In other words, the internal formula of the network can be much more complex than the one derived from the measured curves. The network is self-adapting so the dependence of the result on each particular parameter is found by the network itself. The network can even refuse some parameters if it does not see any dependency. The final accuracy of the prediction is limited mainly by the accuracy of the DTM and the measured data. There will always be a physical limit of the accuracy of the prediction. E.g. any movement of a car or a men on a street

causes a change in the propagation. This is a stochastic element in the real behaviour of the radiowaves. The most important advantages and disadvantages of the neuronal solution will be summarised later.

Neuronal networks used in our experiments

Plenty of artificial neural networks have yet been developed. All of them are, more or less, trying to imitate the functionality of the building cells of the brain, neurones. As it is still more or less unclear how these cells exactly work, the artificial neural network models differs each from other a lot. We have selected two networks to try to create a propagation predictor. Both of them are supervised networks so they need some correct pairs of inputs and outputs to be trained.

Backpropagation is one of the most famous networks. It is probably the most often used network in all practical applications. Its principle can be found in any textbook about neural networks [6]. An example of its structure is in figure 2. The structure of the network must be determined before training and it can not be changed later.
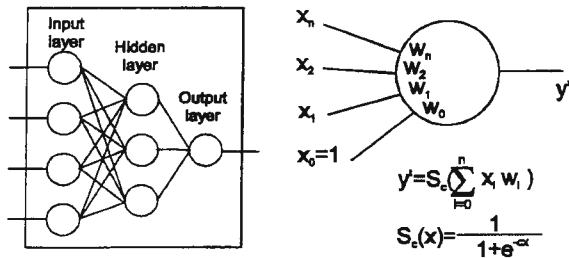
Figure 2: A typical backpropagation neural network and a single neuron of this network. The weights w_i are modifiable by training.
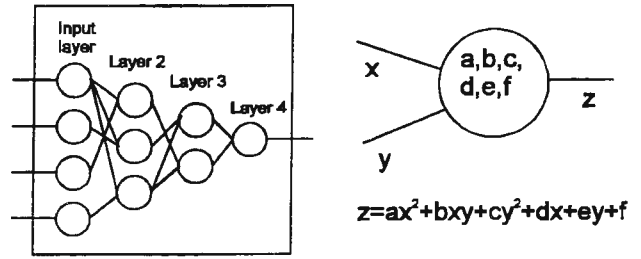
Figure 3: A typical GMDH neural network and a single neuron of this network. The weights a,b,c,d,e,f are modifiable by training.

Group method of data handling (GMDH) is a neural network which is very suitable for creation of mathematical models. An example of the network is shown in figure 3. The structure of the network is created during training in dependence on the given training examples [5], [7], [8]. We implemented the basic combinatorial GMDH algorithm.

In this article, we will think of the neural networks as being black boxes having n analogous inputs and just one output. The output is also analogous and it will represent the predicted field strength. The internal working of the networks is very important, but its description is beyond the scope of this article.

Measured training example set

Measured data is needed to evaluate and tune the classical propagation models. For the neuronal propagation model the data is even necessary for the network to create its own model of the behaviour of the radiowaves. The network has to be trained by some examples of correct pairs of inputs/outputs. These pairs are in our case based on measured samples of the field strength in various conditions. As the propagation model has to work with the DTM, each sample must contain information about the place where the sample was measured. This condition is usually satisfied by using global positioning system (GPS). So the file resulting from the measurement contains co-ordinates for each measured sample. It is important to measure the samples in many different positions against the transmitter, in towns, in countryside, in forest, hidden behind a hill, with direct visibility to the transmitter and in different distances to it. If some kinds of the samples were not represented in the measured file (e.g. no measurements in forest), the network may (but may not) respond incorrectly if it is

asked to predict the field strength in such an environment during the recall phase. The network has a great ability to generalise the given samples. So a small mistake in the measured file does not cause completely wrong function of the system, but leads only to a decrease of the accuracy of the predictions. The samples for our experiments were kindly provided by EuroTel Praha. They were obtained by a measuring receiver and a GPS installed in a special measuring car.

Preparing inputs for the networks, path signature vector

As we imagine the neural network as a black box having n inputs and one output, the inputs must contain all information about the parameters of the BS, and information about the path between the BS and the MS obtained from the DTM. If we fed the neural network just by the rough samples in the form they were measured, that is by co-ordinates of both BS and MS, the only thing the network could create a model of, would be a variation of free space propagation. It is needed to somehow give the disposal of the DTM to the network. Getting all diverse information from the DTM, we form the n inputs, which we will call *path signature vector*. To decrease the size of the path signature vector, parameters which are easy to be taken into account outside the neural network like e.g. radiation diagram of the antenna or radiated power, are calculated outside the network. This also allows to decrease the number of the measured samples.

As described above, there are two main approaches how to take the DTM into account: statistical and multi-ray. Until now we made experiments only with the class of the statistical algorithms of prediction. So we have investigated only the direct path between BS and MS. The best way to investigate this path is to draw terrain profile between BS and MS. So the path signature is derived from the terrain profile.

Two examples of terrain profiles are depicted in figure 4 and 5. The BS is on the left side drawn by a vertical line which represents the 40 meters high tower. The MS is on the right side. Its height above the ground is 1.5 metres. The information obtained from the DTM consists of two parts. The first part is digital elevation model (DEM) which contains information about ground height. The second part is called morpho classes or clutters. It contains information about the buildings or vegetation above the ground. In the profile the DEM is drawn by lines, and the morpho classes are distinguished by rectangles filled by different colours.

In figure 4 the antennas of both stations are connected by a direct line. Beside this, one more line is drawn in the figure. This is so called *Fresnel zone*. It determines an area where the most of energy between the transmitter and the receiver is transferred. The profile in this figure is *line of sight* so there are not any obstacles in the area where the energy is being transferred.
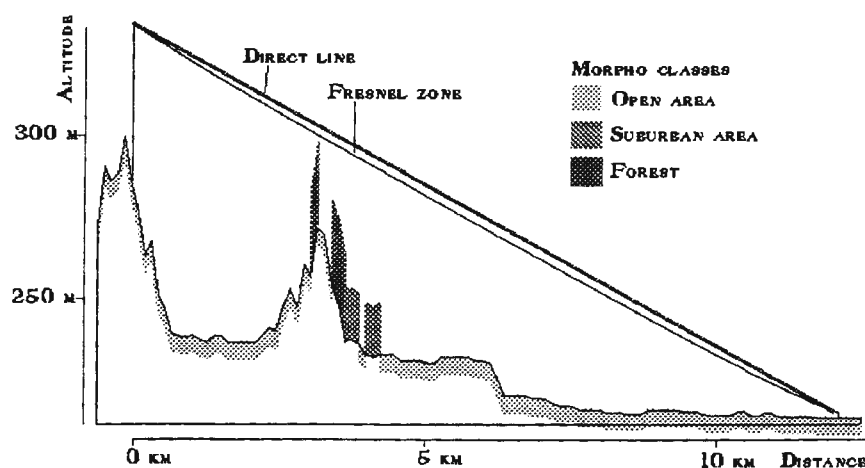
Figure 4: Line of sight terrain profile between BS and MS

In the figure 5, the hill on the right hand side acts as a big obstacle to the direct way of the signal. The path is therefore *non line of sight*. So we suppose that a the hill will work as a diffraction edge. In this sense the final way of the signal is drawn.
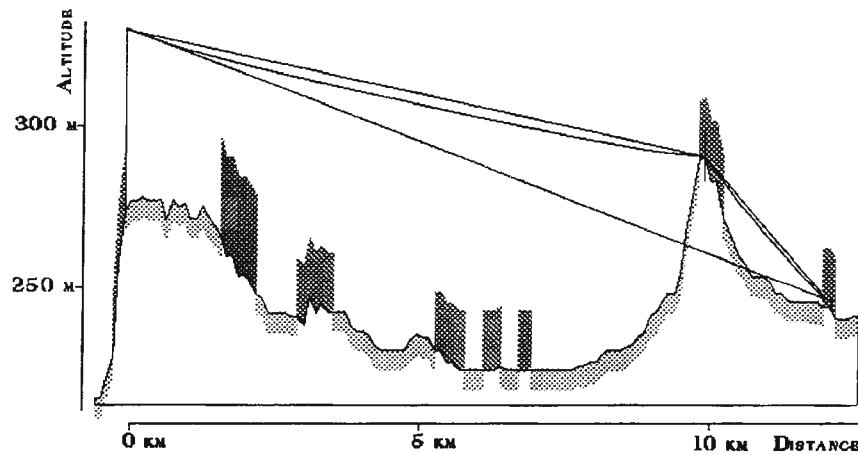
Figure 5: Non line of sight terrain profile between BS and MS

Creating path signature

The inputs of our neural network, that is the path signature, for each calculated path loss is prepared by analysis of the terrain profile between the actual position of the MS and the BS. At first approach we wanted to find out if the neural network is able to reach at least the same level of accuracy as the classical algorithms. So the path signature was constructed with regards to the profile features discussed in e.g. the article of Okumura [1]. It is better to select more features than less. As already discussed, the network is able to reject some features if they seems to be irrelevant to the trained problem. The path signatures created for our experiments contained:

- Length of the distance between the BS and the MS.

- Clutter type in the area of the MS. This clutter type was determined by evaluating of the path from the MS towards the BS in length of several hundred metres. In this way the network is told if the MS is in an area of high buildings, trees or in an open area.

- Effective antenna high. This value represents the high of the BS antenna but the terrain around is taken into account. If the antenna is on a high hill, the effective height will be large, if the antenna is in a valley, the effective height will be very low.

- Height of the biggest obstacle. If there is a line of sight profile between the BS and the MS, the value will be zero. Otherwise, the size of the obstacle is calculated as the distance between the border of the Fresnel zone and the top of the obstacle while theFresnel line is drawn as if the obstacle was ignored.

These were the basic components of the path singature vector, but to make the work of the neuronal net easier, we presented them in some variations so the final number of the components was 12. To help the neural network by well known facts about the electromagnetic energy propagation, one of the components was logarithm of the distance between BS and MS.

Results of our experiments and general comparison

We implemented both the mentioned neural networks and a system which pre-process the information from the DTM to be given to the network as an input in the form of the path signature vector. The network was trained using 1 000 precisely selected samples, and it was evaluated using 27 000 different samples. The network was trained to minimise the standard deviation:

$$\sigma = \sqrt{\sum_{i=0}^{N} (E_{mi} - E_{pi})^2} \ ,$$

where E_{mi} is a measured sample and E_{pi} is its corresponding predicted value. N is the number of all samples.

The training of the GMDH network took about 10 minutes on a Sun Ultra 1 workstation. The training of the backpropagation network took a bit longer time. The results were compared with a classical propagation model implemented in a commercial program, and with a "dummy model". The classical propagation model was based on the Okumura-Hata model with some improvements. The field strength was predicted in constant height 1.5 meters above the ground. The results are the following:

Propagation model	Standard deviation
"Dummy model"; field strength is everywhere equal to the average measured value	16.43 dB
Classical model based on Okumura-Hata, calculated using a commercial program	9.52 dB
Neuronal model using GMDH network (80 neurones, structure drawn in figure 6)	9.17 dB
Neuronal model usingBackpropagation network (12 neurones, 2 hidden layers, fig.7)	8.87 dB

All the results were calculated on the same set of measured routes. If just a smaller part of the measured routes was used, the deviation would differ a lot. The file of the measured samples consisted of 26 routes. We calculated the standard deviation also on separate routes. The achieved minimal value was even below 5 dB. Some others had, however, higher deviation than the number presented in the table. The same was with the commercial program. So it only confirms that for objective evaluation of the model, the deviation must be calculated on all roads, in all parts of the territory.
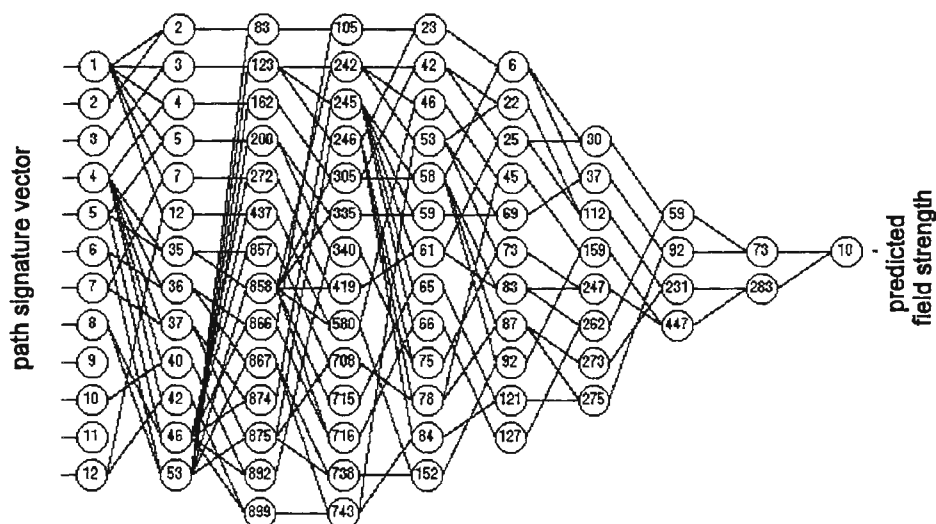
Figure 6: Final structure of the GMDH neural network for prediction of radiowaves propagation.

It is important to explain the absolute values of the deviations. It will never be possible to achieve standard deviation equal to zero. It is estimated that only the fact that cars and people on the streets are moving causes changes in the field level of about 3 dB. Next, we have to take into account that the accuracy of the DTM will always be limited. The real world is much more diverse than just terrain elevations and classifications of several morpho classes. We estimate that the propagation models can never achieve better standard deviation than about 5 or 6 dB. On the other hand, our "dummy model" shows that even very simple estimate of the level of the propagated field can achieve standard deviation of about 16 dB. This value, of course, is very dependent on the type of the terrain where the measurement was performed. In a more hilly terrain, the value would get much higher.

The GMDH network achieved better results than the classical algorithms but worse results than backpropagation. The final structure of the network is shown in figure 6. The numbers written inside the neurones determine their original position in the layer at the time the layer was in creation. That is, the skipped numbers belonged to neurones which later died. It can be seen in the figure that the network decided to ignore several values from the path signature, but some others are very heavily used. E.g. neuron 9 in the input layer represented the existence of a particular clutter type in the place of MS. The neuron 1 represented the distance between BS and MS and the neuron 4 represented its logarithm.
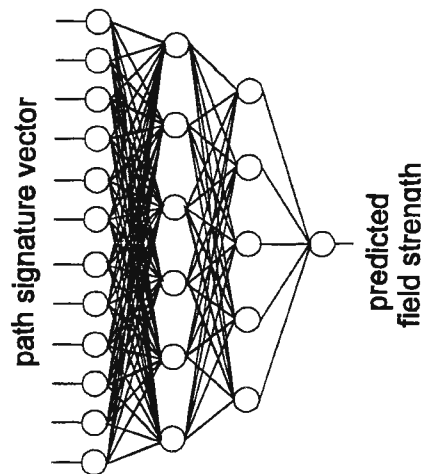
Figure 7: Final structure of the backpropagation neural network for prediction of radiowaves propagation

The backpropagation network achieved the best results in our evaluation. The final structure of the network is shown in figure 7. In comparison to GMDH the structure of this network is very uniform as it was not changed according to the training.

Advantages and disadvantages of our approach

Let us summarise where the neuronal solution is better than the classical one and vice versa.

• To create the neuronal model of propagation, one *must* have some measured data. Without this the network can not be formed. The number of the measured samples must be rather high (tens of thousands) so a special automatic measuring system with a positioning equipment is needed. This is not needed by the classical propagation models. They can be usually tuned using the measured data, but they can work with some default settings as well. The acquisition of the measured data is usually not a great problem as the companies who seriously work in this field must have measuring cars. The accuracy of any propagation model is limited and some parts of the radio network simply must be measured.

• The internal prediction model of the neural network can be much more complex in the mathematical sense. The network can take into account much more parameters than the classical models as the network adapts to

211

the parameters automatically. In the classical model, including of any additional parameter requests a complex research which role the parameter will play, and how to employ it to the existing model.

- Creation of a very accurate DTM is a very expensive task. The neuronal prediction model can adapt even to a DTM in which a systematic error occurs. So even a lower qualityDTMs can be used.

- Although the neuronal model can reach a very high accuracy, it is also very easy to be set up. The classical models have always some limitation in the frequency range and height of the antenna of BS and MS. The neuronal model just need some new measured data to can adapt to the new conditions. Of course, sometimes the preparation of the input signature needs to be changed as well. For a quick overview, one even does not have to think about units of the measured field. The network simply predicts values in the same units the samples were measured in. All these things make the model very flexible.

- The simulation of the propagation using the neural networks is, however, a bit more time complex than the usage of the empirical formulas. That is because we have to obtain more parameters from the terrain profile. The simulation of the function of the neuronal network itself takes just a fraction of this time. Although the first version of the neural network model is implemented on sequential machine, it is easy to implement it in parallel. The neural network is also easy to implement in hardware, where the level of the parallelism can be much higher than in the case of the standard parallel computers.

Conclusions

Our experiments show that artificial neural networks can successfully solve even so complex problem like simulation of radiowaves propagation. Up to now this domain belonged only to classical algorithms. The neuronal solution is very flexible and can be adapted to almost any variation of the problem. The neuronal network can be adapted even to some variations of the problem where the classical solution is not yet known or its accuracy is not guaranteed. A new adaptation, that is new training of the network, can be done just in a few tens of minutes.

First experiments shown that the neuronal predictor of propagation can achieve even better accuracy than the classical algorithms. Calculation of the standard deviation is, however, not the only way how to evaluate the abilities of a propagation model.

In our future work, we would like to concentrate on better evaluation of the results achieved by different propagation models. Next we would like to improve the implementation of the neural networks to better suit to the application. The authors of this article will appreciate any comments of this work. We would like to thank to employees of RF planning department of EuroTel Praha for their support in this work. As meaning and experience of people interested in similar subjects is very important for us, we are looking for another people and companies who would like tocooperate in this work.

References:

[1] Yoshihisa Okumura, Eiji Ohmori, Tomihiko Kawano, Kaneharu Fukuda: Field Strength and Its Variability in VHF and UHF Land-Mobile Radio Service, *Review of the Electrical Communication Laboratory*, Vol. 16, 1968, pp. 825--885,

[2] Masaharu Hata: Empirical Formula for Propagation Loss in Land Mobile Radio Services, *IEEE Transactions on Vehicular Technology*, Vol. 29, 1980, No. 3, pp. 317--325.

[3] Lee William C. Y.: *Mobile Cellular Telecommunications Systems*, McGraw-Hill Book Company, 1989.

[4] Linnartz Jean-Paul: *Narrowband Land-Mobile Radio Networks*, Artech House, 1993.

[5] A.G. Ivakhnenko, G. A. Ivakhnenko: The Review of Problems Solvable by Algorithms of the Group Method of Data Handling (GMDH), *Pattern Recognition and Image Analysis*, Vol. 5, No. 4, pp. 527-535, 1995.

[6] R. Rojas: *Neural Networks - A Systematic Introduction*, Springer 1996.

[7] Robert Hecht-Nielsen: *Neurocomputing*, Addison-Wesley Publishing Company 1991.

[8] Stanley J. Farlow: *Self-Organizing Methods in Modeling*, GMDH Type Algorithms, Marcel Dekker, Inc. 1996.

EMPIRICAL MODELING OF THE LIME KILN PROCESS WITH NEURAL NETWORKS

Mika Järvensivu

Helsinki University of Technology, Process Control Laboratory, 02150 Espoo, Finland, mika.jarvensivu@hut.fi

Abstract

The objective of this paper is to illustrate the applicability of neural networks for predicting the static behaviour of a nonlinear and multivariable lime kiln process. The lime kiln is an essential part of the kraft recovery system at the pulp mill and the kiln should be able to produce sufficient amount of the high quality lime to meet the needs of the recausticizing operation, while minimizing its energy consumption and emissions.

In this paper, the experiences and findings, as well as the practical results achieved during the development and implementation stage of the neural network predictor for burned lime residual $CaCO_3$ content is presented. The presented work was carried out as a co-operative effort between academic and industrial representatives, and the developed system was implemented at the Pietarsaari pulp mill in western Finland.

Introduction

During the last decade, Advanced Process Control (APC) strategies and optimization techniques have been extensively applied in many industrial processes. Although there are many different ways of implementing advanced process control, the common prerequisite for the success of each implementation principle is the availability of reliable, fast and accurate measurements and/or predictions of product quality properties.

Consequently, an increasing number of on-line sensors and analyzers are to be found in all fields of the process industry. The basic function of online analyzers is to monitor the state of the (sub)process, e.g. the chemical composition of the raw material, product and effluent flows. Unfortunately, it is difficult to measure the chemical composition on-line, and hence sensitive and complex analyzers are often required. As a result, an on-line analyzer may have several shortcomings, e.g. high purchase-associated costs and high operational costs, due to the frequent and expensive maintenance, repair and calibration required.

In contrast, if manual sampling and laboratory analysis, i.e. off-line measurements, are used instead of on-line measurements, the long time delays which are typically involved will significantly reduce the usefulness of the gathered information and decrease control performance in general. In addition, laboratory analyses are usually highly labour intensiv, expensive and require personnel with special skills.

Another possible approach is to obtain predictions of product quality properties without resorting to physical analyzers or sensors. Process models deduced from indirect but routine and readily available process measurements, e.g. pressure, flow, density and temperature etc., can be used instead. In fact, sufficiently accurate process models that explain the functional relationship between process measurements and for product quality properties may offer a low-cost alternative to an on-line analyzer or a laboratory analysis.

Empirical modeling approach is an alternative means of representing the functional relationship between process inputs and process outputs. Properly designed and developed empirical models can be applied to provide an adequate description of the static or dynamic relationships between inputs and outputs in processes, which otherwise may be either too difficult or expensive (or even impossible) to measure with an on-line analyzer or to predict using a model based on the fundamental principles. In fact, there has been widespread interest in various empirical modeling approaches and especially in the last few years different types of neural network models.

213

Neural networks

In general, the neural networks (NNs) can be regarded as a generic nonlinear black-box function approximator or model that maps certain process measurements to predicted network output(s). The following characteristics and properties of NNs have great importance from the system identification and process control point of view:

- the ability of NNs to learn and adapt behaviour of past operational data
- the simple structure and fast overall processing of the basic elements of NNs in a pretrained network
- the ability of NNs to approximate arbitrary nonlinear mappings and to generalize when presented with new data that do not appear in the training data (e.g. robustness with respect to noisy process data)
- the built-in applicability of NNs to process multivariable systems and their ability to operate simultaneously with both quantitative and qualitative data (i.e. data fusion)

There are a number of netwotk architectures available, e.g. single layer perceptron, multilayer feedforward, multilayer feedback and recurrent networks, with many supervised and unsupervised learning algoriths for each representation. Although a number of architectures have been proposed, the multilayer feedforward networks are by far the most widely applied.

In this work, two differerent feedforward networks, i.e the BackPropagation Network (BPN) and the Radial Basis Function Network (RBFN), are used. The BPN, presented in Figure 1, is a multilayer, feedforward network, using the supervised mode of learning. The RBFN, presented also in Figure 1, is a three layer, feedforward network, based on the concept of the locally tuned and overlapping receptive fields.

In general, given a training set of input-output pairs $\{(x^{(k)}, d^{(k)})\}$, $k= 1,2,3...,p$, feedforward networks provides a procedure for calculating network outputs, $y^{(k)}$, and for changing the weights in a network to map the given input patterns correctly to the corresponding outputs. When processing element (PE), q, in the hidden layer receives a net input of

$$NET_q = \sum_{j=1}^{m} v_{qj} x_j \tag{1.1}$$

where x_j is the input vector, v_{gj} is the weight vector, it produces an output of

$$z_q = a_h(NET_q) = a_h \left(\sum_{j=1}^{m} v_{qj} x_j \right) \tag{1.2}$$

where a_h is the activation function of hidden node and z_q is the output from qth hidden node. The net input for a PE, i, in the output layer is then

$$NET_i = \sum_{q=1}^{l} w_{iq} z_q = \sum_{q=1}^{l} w_{iq} a_h \left(\sum_{j=1}^{m} v_{qj} x_j \right) \tag{1.3}$$

and it produces an output of

$$Y_i = a(NET_i) = a_o \left(\sum_{q=1}^{l} w_{iq} z_q \right) = a_o \left(\sum_{q=1}^{l} w_{iq} a_h \left(\sum_{j=1}^{m} v_{qj} x_j \right) \right) \tag{1.4}$$

where a_o is the activation function of output node and Y_i is the actual output from the ith output node.

The most commonly used activation functions for BPN are linear ramp function and bipolar sigmoidal functions. Linear ramp function

$$
a(f) = \begin{cases} 1 & \text{if } f > 1 \\ f & \text{if } 0 \le f \le 1 \\ -1 & \text{if } f < -1 \end{cases}
$$
(1.5)

is typically used for nodes in the input layer and the output layer, and bipolar sigmoidal function

$$
a(f) = \frac{2}{1 + e^{-\lambda f}} - 1
$$
(1.6)

where λ determines the steepness of the continuos function $a(f)$ near $f = 0$, is used only for nodes in the hidden layer(s). Unlike the BNP in which the hidden nodes may have linear or sigmoidal functions, the hidden nodes in the RBFN have Gaussian activation function and produces an output of

$$
z_q = \frac{R_q(x)}{\sum_k R_k(x)} = \frac{\exp\left[-|x - m_q|^2 / 2\delta_q^2\right]}{\sum_k \exp\left[-|x - m_k|^2 / 2\delta_k^2\right]}
$$
(1.7)

where x is the input vector, m_q and σ_q are the mean and variance of the qth Gaussian function and z_q is the output from qth hidden node. Each hidden node q is said to have its own receptive field $R_q(x)$ in the input space, which in practice is a region centered on m_q with size proportional to σ_q. For an input vector x, the receptive fields with centers close to it will be appreciably activated and the output of the RBFN is then simply the weighted sum of the activations of these receptive fields

$$
Y_i = a_o\left(\sum_{q=1}^{l} w_{iq} z_q\right) = a_o\left(\sum_{q=1}^{l} w_{iq} \frac{\exp\left[-|x - m_q|^2 / 2\delta_q^2\right]}{\sum_k \exp\left[-|x - m_k|^2 / 2\delta_k^2\right]}\right).
$$
(1.8)

Other difference between the BPN and the RBFN is that, the normal functionality of the RBFN can be extended by adding additional output, i.e. maximum hidden node activation, indicating the confidence of the RBFN's prediction, which can not be easily done with basic BPN. More details about neural networks in generally and their training algoritms, as well as the BNP and the RBFN are available for instance in references 11-16.
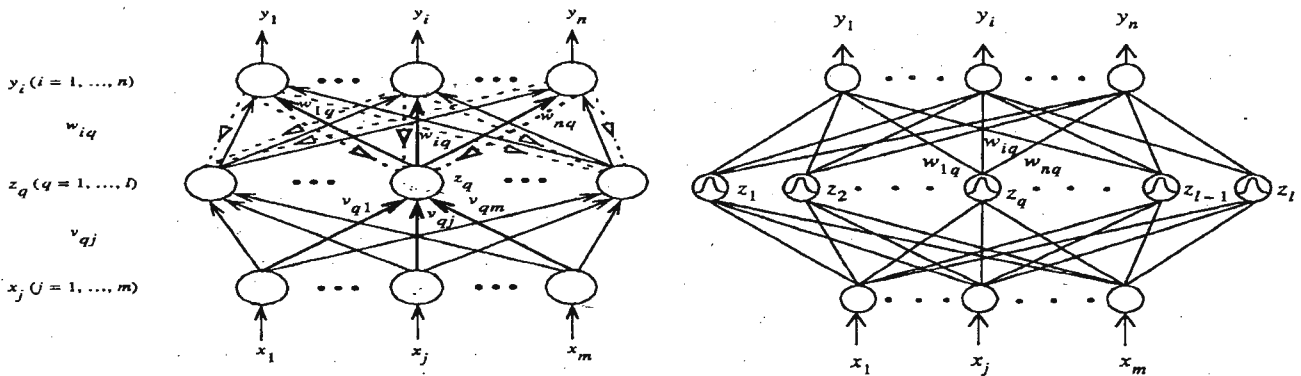

Figure 1. Structure of the BNP and the RBFN

Lime mud reburning

The causticizing process, which is part of the pulp mill chemical recovery produces white liquor from green liquor. This process consumes lime, CaO, and produces lime mud, principally $CaCO_3$, as a by-product. The purpose of the lime mud reburning plant is to convert the lime mud ($CaCO_3$) back into the lime (CaO) for reuse in the causticizing process. The quality of the reburned lime is a very important parameter in the production of low cost, high quality white liquor. For instance, the lime quality influences the rate of the slaking and the causticizing reactions, and the separation properties of the produced lime mud. /17/

Lime kiln process
The rotary lime kiln (Figure 2), which is a large cylindrical oven with a length between 50 and 120 m and a diameter between 2 and 4 m, has been the prime method of calcination in the pulp industry. Lime kilns are slightly inclined from the horizontal and are slowly rotated on a set of roding rings.
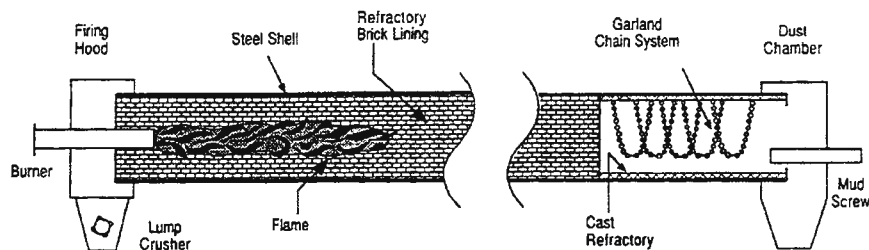
Figure 2. Rotary lime kiln /18/

Physical and chemical phenomena in the lime kiln
Conceptually, the lime kiln may be broken into four process zones. These four zones are drying of the lime mud, heating of the lime mud solids up to the reaction temperature, calcining of the calcium carbonate into the calcium oxide and completing the calcining reaction (burning).

In the drying, described by the Equation (2.1), residual unbound and bound moisture present is removed thermally to yield a solid lime mud. The unbound moisture is held in capillaries between lime mud particles, due to the incomplete vacuum filtration. The remaining moisture, i.e. bound moisture, which can not be removed by vacuum filtration, is either sorbed on the surface of the lime mud particles through hydrogen bonding or trapped in the microstructure of the lime mud particles. However, in the practice when wet lime mud is subjected to thermal drying, two processes occur simultaneously; transfer of heat from the hot fluegases to evaporate and vaporize the unbound surface moisture and transfer of internal bound moisture to the surface of the lime mud particles and its subsequent evapoartion due to the first process.

drying of the lime mud: $CaCO_3 (s) + H_2O (l) + heat \rightarrow CaCO_3 (s) + H_2O (g)$
(2.1)
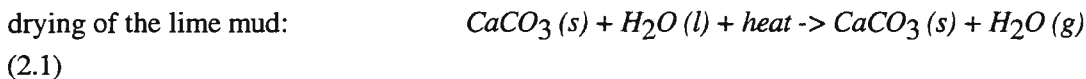
The reaction heat required for the evaporation is 2256 kJ/kg H_2O. The first process depends mainly on the external conditions, such as temperature, humidity and rate of the fluegas flow and the second process depends mainly on the physical nature, moisture and temperature of the lime mud particles. During the first stage of lime mud drying the drying rate is nearly constant, i.e unbound moisture in the surface is evaporated and the rate of drying is determined by the rate of evaporation. On the further drying moisture has to be transported from inside of the lime mud particles to the surface and when the critical moisture content has reached, i.e. when dry spots appear upon the surface, the drying rate falls. After drying zone, i.e. when all moisture is removed from the lime mud, the lime mud solids temperature starts to rise. The temperature rise settles down, when the main reaction, i.e. calcium carbonate dissociation to calsium oxide and carbon dioxide, inside a lime kiln start to consume heat energy. The calcining reaction, described by the Equation (2.2), is endothermic and reaction heat required is 1786 kJ/kg $CaCO_3$ decomposed or 2966 kJ/kg CaO formed.

calcining reaction: $CaCO_3 (s) + heat \rightarrow CaO (s) + CO_2 (g)$ (2.2)
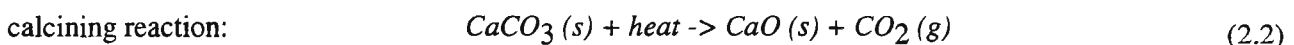
The calcination reaction starts already at about 800 $^{\circ}$C, but typically in practice the temperature up to 1000-1100 $^{\circ}$C is required in the hot-end of the kiln to increase reaction rate and complete the reaction. After calcination, the reburned lime is a very fine powder. Therefore in order to produce a usable product, the lime kiln must not only dry the lime mud and heat it in order to calcination to occur, but it must heat calcined powder further upto about 1000 $^{\circ}$C to achieve agglomeration of the powder into nodules.

Attempts have been made to mathematically model the dynamics of the solids phase, the heat transfer phenomenon and mechanisms occuring inside a rotary kiln, and the kinetics of drying, heating and calcination reaction occuring inside a rotary kiln. However, an accurate and exact mathematical model of the lime kiln process is difficult to obtain. Furthermore, if successful it will lead to highly complicated nonlinear functions in the form of partial differential or algebraic equations. As a result the evolution of the kiln process continues to depend more on rule-of-thumb empiricism and information gained from the laboratory and full scale experiments. /19-26/

Experimental results

Since 1996, the lime reburning process (i.e. lime mud filters and lime kiln operation) have been studied extensively at the Wisaforest pulp mill with the aim of developing and implementing an advanced lime kiln control system. The desirable objects of the system under development are improved lime kiln operation efficiency (i.e. increased production capacity), reduced variation in lime quality (i.e. burned lime residual $CaCO_3$ content) and decreased flue gas hydrogen sulphide emissions. During the field survey period, the burned lime residual $CaCO_3$ content was considered as critical operational/quality parameter. For instance, reburned lime residual $CaCO_3$ content has a significant impact on the specific energy consumption (GJ/t$_{CaO}$) and on the down-stream process operation. /27, 28/

On the other hand, at the Wisaforest pulp mill the burned lime residual $CaCO_3$ content is manually sampled with an 8-hour time interval, and analyzed with a delay time of about 2 hours in the laboratory. Different alternatives to reduce the time delay involved in the current practice were investigated. An empirical non-linear modeling approach using neural networks proved to be promising and was therefore selected.

Llime reburning process at Wisaforet pulp mill
At the Wisaforest pulp mill clarified lime mud from mud storage is dewatered and washed by means of two rotary vacuum drum filter devices prior to burning in the lime kiln. The kiln has a total length of 104 m, an external diameter of 3.6 m and angle of inclination of 2 $^{\circ}$and a design capacity is 400 t CaO/d.

Lime mud is fed into the cold-end of the lime kiln at about 75 % solid consistency and the drying and part of the heating of the lime mud is done in an external dryer, a lime mud drier (LMD), instead of the conventional chain section. The dried lime mud is fed into the kiln as a dry powder. The main fuel of the kiln is saw mill dust which is first dried with the flue gases from the recovery boiler and then gasified. The heat from the hot lime leaving the kiln is used to preheat the combustion air in so called product coolers. The flue gases are emitted into the atmosphere through an electrical precipator and a water scrubber.

Process data collection and preprocessing

The lime mud filters and the lime kiln operation were studied by means of a 12-months field survey, which comprised both normal operations and process experiments. During the field survey a large amount of process data was collected and archived into the plant information system database. The collected data includes routine laboratory analyses, and all corresponding process measurements. Figures 3 shows the reburned lime residual $CaCO_3$ content and production rate as well as cold end temperature during the field survey period. The large amount of process data were preprocessed by means of missing value substitution, and removing the outliers and other data anomalies. After data preprocessing, the data were examined by using multiple linear regression analysis.
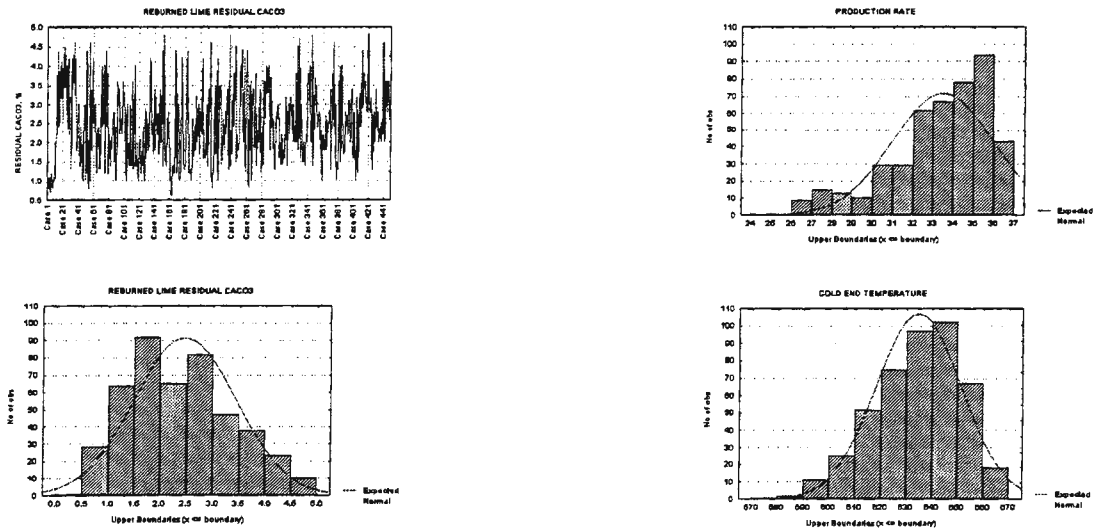
Figure 3. Reburned lime residual CaCO₃ content, production rate and cold end temperature

The search for the most relevant input variables to be used in the linear regression and in the neural network models was based on the results from statistical analysis and prior process knowledge. Tables 1 shows the input variables used in the linear regression and the neural network models.

Table 1. Input variables used in the models for predicting the residual CaCO₃ content of the reburned lime

	Residual CaCO₃ content
1	Cold end temperature with 2 hours time delay
2	Middle kiln temperature with 1 hour time delay
3	Hot end temperature with 1 hour time delay
4	Heavy fuel feed rate with 1 hour time delay
5	Production rate with 3 hours time delay

After raw data preprocessing and selection of the input variable, the data were combined into a dataset. The dataset includes values of the selected input variables and corresponding laboratory analysis results. Next, a data pair outlier filter was used to check each element of the data pairs against fixed lower and upper limits, rejecting those data pairs that violated the limits. Furthermore, redundant data pairs were removed to make sure that the dataset spreads out and evenly covers the input space. In addition, all the input variables were normalised from 0 to 1. These normalized variables were then used to generate the input patterns for the BNP and the RBFN.

Network training and validation, and detection of the extrapolation
Various types of network architecture and training algorithms to select an appropriate network architecture. In each experiment 2/3 of the data, were randomly selected to train the neural network, and the remainder were used as a test set to validate the trained neural network. The sequence of training, validation and re-randomization of weights were repeated to achieve statistical representativenes of the training and testing error. Table 2 shows the square root of the mean of the squared errors for all the data, for training and validation for the different architectures. Figure 4 shows the mean of the training and validation error with different network architectures. Figure 5 shows clearly that the network with 5 hidden nodes is not sufficiently complex and the network with 15 hidden nodes is overly complex for the problem at hand. Finally, the best architecture with one hidden layer and 8 hidden nodes was trained using all the available data. A comparison of the laboratory analysis and the neural network outputs is shown in Figure 5.

218

Table 2. Fitting errors with the different acrhitectures

5 Hidden nodes, 300 Iterations			8 Hidden nodes, 300 Iterations			10 Hidden nodes, 300 Iterations			15 Hidden nodes, 300 Iterations		
Training	Validation	Difference	Training	Validation	Difference	Training	Validation	Difference	Training	Validation	Difference
0.606	0.727	0.121	0.496	0.660	0.164	0.382	0.612	0.230	0.378	0.754	0.376
0.618	0.806	0.188	0.554	0.770	0.216	0.490	0.622	0.132	0.391	0.673	0.282
0.572	0.702	0.130	0.524	0.773	0.249	0.399	0.661	0.262	0.391	0.579	0.188
0.623	0.750	0.127	0.371	0.547	0.176	0.453	0.661	0.208	0.367	0.803	0.436
0.675	0.776	0.101	0.521	0.750	0.229	0.500	0.870	0.370	0.365	0.681	0.316
0.564	0.756	0.192	0.473	0.637	0.266	0.438	0.709	0.271	0.310	0.785	0.475
0.620	0.789	0.169	0.508	0.777	0.304	0.513	0.775	0.262	0.331	0.862	0.531
0.619	0.673	0.054	0.467	0.575	0.102	0.493	0.740	0.247	0.425	0.745	0.320
0.635	0.791	0.156	0.464	0.730	0.222	0.431	0.579	0.148	0.389	0.675	0.286
0.576	0.786	0.210	0.494	0.722	0.255	0.425	0.852	0.427	0.438	0.770	0.332
0.611	**0.756**	**0.145**	**0.487**	**0.694**	**0.218**	**0.452**	**0.708**	**0.256**	**0.379**	**0.733**	**0.354**
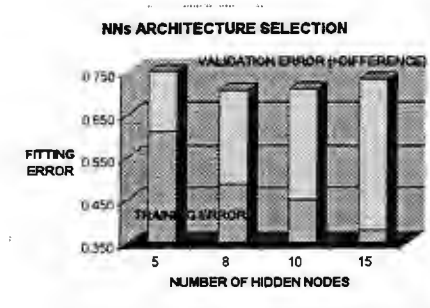
Figure 4. Mean of the fitting errors with different network architectures
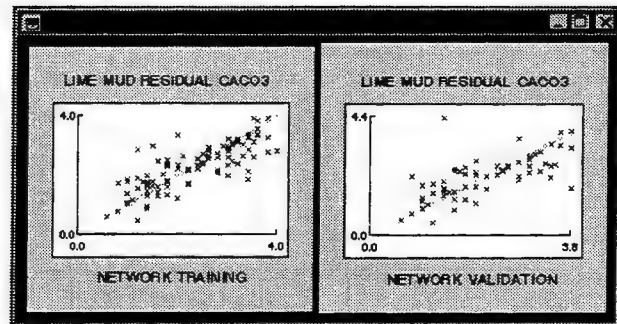
Figure 5. Fitting results when 8 hidden nodes were used

After the BPN training and validation, the RBFN were trained to determine whether the neural network starts to extrapolate. The RBFN were trained with the same data as the networks (BNP) used in actual quality property prediction. Cases which require extrapolation are automatically prevented to avoid inaccurate predictions. The presented neural network model will be used as a part of the advanced lime kiln control system. Figure 6 shows the presented neural network model connections within the overall lime kiln control system and Figure 7 illustrates the structure and the information flows of the developed and implemented system .
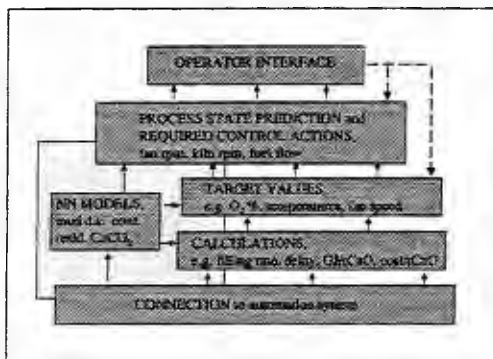
Figure 6. The neural network model connections within the overall lime kiln control system
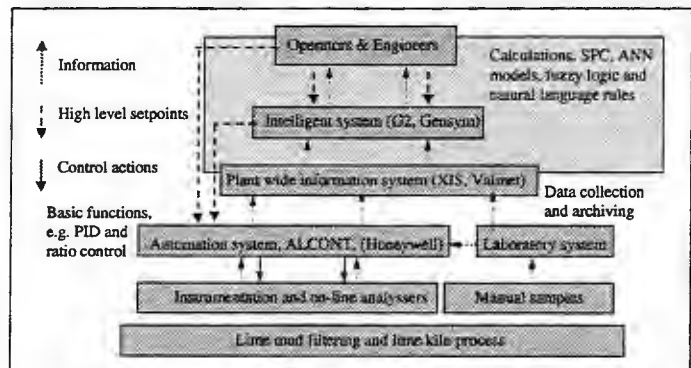
Figure 7. The structure and the information flows of the developed and implemented system

Conclusions

This paper has illustrated the applicability of neural network technologies to model an industrial multivariate nonlinear process. From a control point of view, the developed neural network model predicted the quality parameter sufficiently accurately.

Furthermore, the project has shown that comprehensive history data together with properly designed and developed neural network model give good starting point predicting optimum operating parameters for the industrial lime kiln. On the whole, primary results achieved are so promising that the project will continue and the aim of the further development is to implement neural network based control to maintain desired residual $CaCO_3$ content.

Acknowledgements

References

1. B. Pradeep, Predict difficult-to-measure properties with neural analyzer, Control Engineering, July 1997, p. 55-56
2. A. Brambilla, Estimate procduct quality with ANNs, Hydrocarbon Processing, September, 1996, p. 61-66
3. K. Hunt, Neural networks for control systems, Automatica, 28(1992):6, p.1083-1112
4. J. Song, Neural model predictive control for nonlinbear chemical processes, Journal of Chemical Engineering, 26(1993):4, p. 347-354
5. S. Ramasamy, Consider neural networks for process identification, Hydrocarbon Processing, June 1995, p.59-62
6. M. Willis, Artificial neural networks in process estimation and control, Automatica, 28(1992):6, p. 1181-1187
7. J. Sjöberg, Nonlinear black-box modelling in system identification, Automatica, 31(1995):12, p. 1691-1724
8. Boullard, L., Application of Artificial Intelligence in Process Control, Pergamon Press, 1992, 455 p.
9. Pradeep, B., Predict difficult-to-measure properties with neural analyzer, Control Engineering, July 1997, p. 55-56
10. Ramasamy, S., Consider neural networks for process identification, Hydrocarbon processing, June 1995, p.59-62
11. Hunt, K., Neural networks for control systems, Automatica, 28(1992):6, p.1083-1112
12. J. Freeman, Neural Networks, Algoritms, Applications and Programming Techniques, Addison-Wesley Publishing Company, 1991, 401 p.
13. Phillips, D., Neural Network Basics, The computer applications Journal, (1993):9, p.36-43
14. C. Lin, Neural Fuzzy systems, Prentice Hall PTR, 1996, 797 p.
15. J. Moody, Fast learning in networks of locally-tuned processing units, Neural Computing, 1989:1, p.281-294
16. J. Leonard, Using radial basis function to approximate a function and its error bounds, IEEE Trans. Neural Network, 3(1992):4, p. 624-627
17. H.Theliander, On the filtration properties of lime mud, Nordic Pulp and Paper Journal, (1990):2, p. 74-87
18. Pulp and Paper Manufacture, Volume 5. Alkaline Pulping, 1989
19. H. Guruz, Mathematical modelling of rotary cement kilns by the zone method, the Canadian Journal of Chemical Engineering, 59(1981):8, p.540-548
20. R. Backman, Modelling the chemistry in lime kilns, Åbo Akademi University, Report 91-17, 1991, 25 p.
21. S. Tscheng, Convective heat transfer in a rotary kiln, The Canadian Journal of Chemical Engineering, 57(1979):8, p.433-443
22. K. Brown, Mathematical modeling and computer control of lime kilns, TAPPI Proceedings, 1983 Pulping Conference, p. 585-592
23. T. Karhela, Dynamic simulation model of rotary lime kiln, Report 6-96, Tampere University of Technology, 1996
24. D. Smith, Dynamic mathematical model of a rotary lime kiln, TAPPI Proceedings, 1991 Engineering Conference, p. 447-455
25. L.Wang, Dynamic modeling and simulation of the recausticizing plant in a kraft pulp mill, Tappi Journal 77(1994):12, p. 95-103
26. H. Theliander, A system analysis of the chemical recovery plant of the sulfate pulping process, Nordic Pulp and Paper Research Journal, 1988:2, p.60-67
27. Järvensivu M., Evaluation of various alternatives to reduce TRS emission at the lime kiln, TAPPI International Chemical recovery Conference, 1998, Tampa, USA
28. Järvensivu M., Neural network model used for quality parameter prediction, Seventeenth IASTED International Conference, MIC' 98 Modeling, Identification and Control, February 18-20, 1998, Grinderwald, Switzerland

A Generalized Fingerprint Case Study

Miroslav Snorek

Czech Technical University in Prague, Faculty of Electrical Engineering,
Department of Computer Science and Engineering, Karlovo Namesti 13, Praha 2, The Czech Republic,
Snorek@cslab.felk.cvut.cz

Abstract: This paper describes an application of generalized fingerprint method [1]. This method gives the information whether the functionality of a system (both technical or non-technical) is standard or not. In this context as standard, the behavior of the system depends on conditions specified in advance. Standard system gives characteristic external manifestation. The classification, or, more precisely, the diagnostic according to the generalized fingerprint method is performed in our case by an artificial neural network (ANN). This approach is very suitable in cases of complex nonlinear systems with unknown dynamics or with dynamics which are impossible to describe explicitly.

Keywords: Neural Networks, Generalized Fingerprint, SOM.

Introduction

The method of generalized fingerprint is similar to the approach of a human specialist. Based on evaluation of system's manifestation under known marginal conditions the specialist is able to state that the system is "healthy", i.e. with high probability of working well. For other systems he is able to predict problems which may be encountered. We proceed from old and well known method of identification of persons according to unique and unchangeable patterns of fingers - fingerprints. Similarly, in the technology [1] the fingerprint means the pattern of standard behavior. Any change from this pattern indicates a change to non-standard state and potentially to faulty state. In [2] a fingerprint approach to standard behavior of rotating mechanical device is described.

Diagnostic described here is based on system's echo, it's manifestation, under just valid external conditions. From the echo it is possible to state whether the system is in standard or good state or to find a deviation from such a state and sort of fault, or what kind of fault one can await.

The diagnostic method used here utilizes specific features of ANNs, especially the ability to learn by examples, and thus producing better approximation of original system [3]. Neither the information on internal structure, nor the model of the system is needed for this [6]. Besides, the input data are typically multidimensional and systems under consideration are usually not well defined. It is only known that a standard or "healthy" system under known conditions is a system giving external manifestation in just a particular manner.
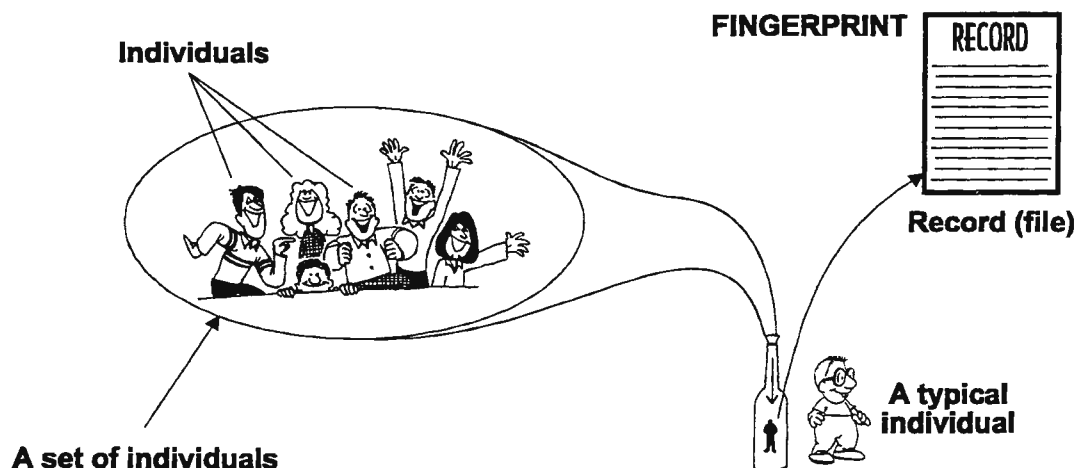
The method of generalized fingerprint represents quite a new approach in diagnostics. For it's practical application it will be necessary to elaborate a methodology for selection of the input data set, and methods of collection and preprocessing of the data. With respect to multidimensional character of these data it is necessary to design methods of

- reduction of the dimensions of data,
- data visualization,
- data classification using selected types of ANNs.

These particular items are out of the scope of this paper.

A "Bad" Student Prediction - Fingerprint Case Study

About one thousand students are admitted yearly to study at the Czech Technical University of Prague, Faculty of Electrical Engineering. The number of applicants usually greatly exceeds the capacity of the faculty. The problem of how to select the best of them is solved by so called Accepting procedure. The decision is based on secondary school results and results reached in a special entrance test (done in mathematics and physics). Our case study represents the parallel approach.
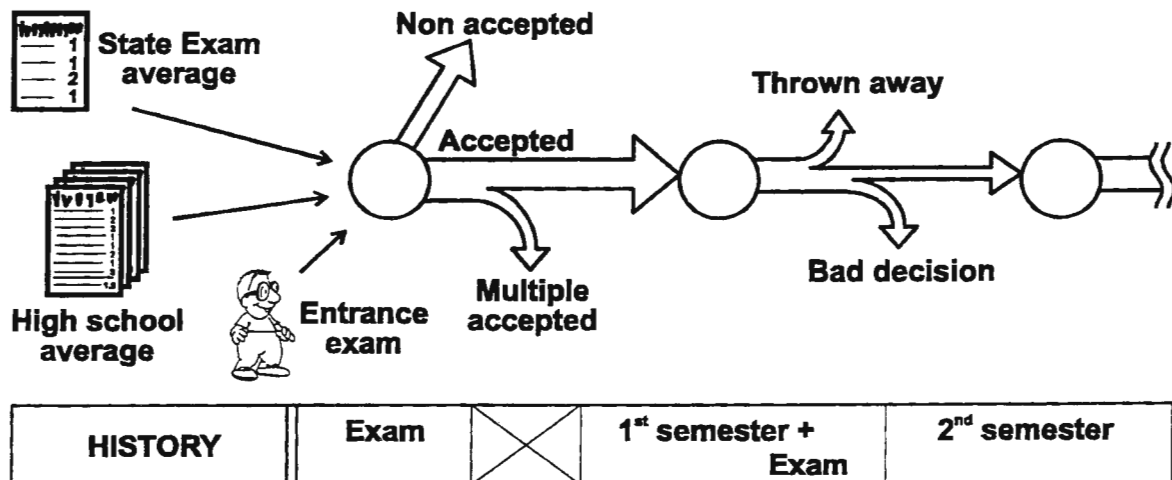
Selection of students based on the above mentioned criteria hardly covers all of their personalities. There are also other factors influencing his/her results or success which can not be predicted at the moment when they start to study (for example family problems, influence of totally new environment, out-of-school activities). Therefore one expects to lose number of successful students during their studies.

The target of the case study performed at the Department of Computer Science and Engineering, CTU Prague in 1995-1998 was to prove that the diagnostic by the method of generalized fingerprint is suitable for a non-technical application. The aim was to predict those possibly non-successful among our students by means of the generalized fingerprint method. Our first results were outlined in [4] and [5]. They are also summarized in the Table (App.1 - 3). Here we present mainly the latest results (App.5) and the roots of the improvements.

Starting point

Database information is available on several hundred students/year who started their studies in 1995 and 1997. This information contains datafields on student's secondary school careers (marks, the school type, living place) as well as their results achieved at our university (entrance exam results, number of credits).

Actual definition of the non-successful, "Bad" Student

As starting definition - the "Bad" student did not get enough credits (10) in his first term - was utilized. Several other definitions were also tested (not enough credits in mathematics, left the faculty), all in App.1-3. In the actual approach (App.5) we do not simply separate bad and good students. We predict the probability whether those, once denoted as "Bad", were really "Bad" (really did not get enough credits). As follows from the Table, this definition gives us much better results (App.5 and App.5b).

Method used

Several different Back-propagation and SOM ANNs were built and trained to find the fingerprint of a "Bad" student and then to perform the prediction. Here are the most promising structures:
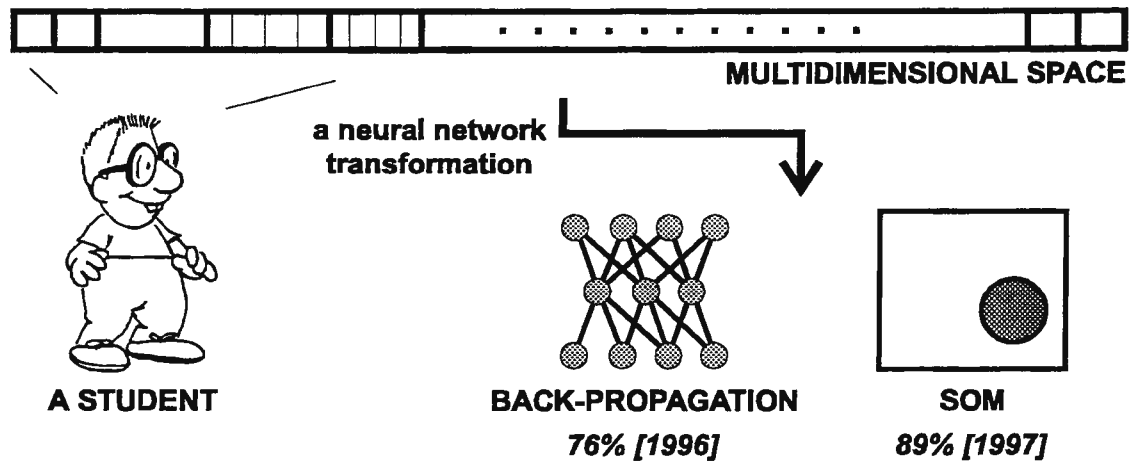
Type	Architecture	Best result	Comments
App.1 BP	10-6-1	76%	binary output
App.2 BP	3-7-2	75%	reduced input set
App.3 BP	10-117/58/29-2	76%	ternary output
App.4 BP	16-1-2	72%	cascade learning
App.5 SOM	10x10-5-1	89%	
App.5a BP	16-5-2	86%	standard input set
App.5b BP	16-5-2	88%	modified input set

Table: The most promising ANN architectures and results

Necessary background

Back-propagation ANNs are well known, so there is no need to explain any details. We only notice that both delta-rule and cascade learning were used during the learning phase.

SOM was originally developed to visualize topologies and hierarchical structures of multi-dimensional input spaces. It typically has several layers. The input layer is fully connected to a (usually) two-dimensional Kohonen layer, which is the heart of SOM. Kohonen layer transforms any n-dimensional input space into an ordered, 2-dimensional, so called feature map. Each Kohonen layer PE measures the Euclidean distance of its weights to the incoming input values. SOM learns without supervision, hence the word self-organizing. The Kohonen layer outputs are fed into a hidden layer which in turn feeds into a predictive or categorical output layer.

Results

From statistical analysis it was derived, that in the entire data set 70% of all new students are successful in their first term.

When using SOM applied to the same input data as in [4], the success in the prediction of a "Bad" student increased to 89% from 76% in [4] (BP). The net denoted 25% of students as "possibly Bad".

Although we try to improve our results, 100% success in the prediction can not be reached. A student's decision to leave the faculty can be based on thousand different reasons.

Where are the roots of improvements?

- The neural net type has been changed (SOM instead of BP),
- the training set was modified,
- the input and output variable coding was changed,
- the definition of the non-successful, "Bad" Student was changed.

Discussion

The training set modification was necessary for two independent reasons: missing items in the input data and the 50:50 score of good and bad students in the training set was necessary.

An ANN expects input vectors to be complete. Since in this case a lot of items were missing in the database, it was useful to train and process incomplete patterns as well. Here we have derived and successfully tested a new approach overcoming the missing items.

In some cases the learned net tends to the so called constant or dominant answer. To be sure that the dominance is caused by the rule and not only by the input data causality, we found, that there must be 50:50 score of good and bad student input vectors in the training set.

The input and output coding change was also necessary. Good result is now coded as 0, bad is coded as 1. This coding ensures, that the net excitation is provided by bad results only. It was also proved, that the nonlinear neuron in the output predictive layer must not separate the output interval in two identical subintervals. The center has to be set at approx. 0.7 instead of 0.5.

Conclusions

As the most valuable result of our latest investigations, the method of preventing the net's tendency towards the constant answer is supposed. The second, most valuable result, is the output interval splitting-point recommendation.

Acknowledgment. Most of the practical testing was done by Mr. Roman Mokrosz, Jurjan-Paul Medema and Martin Remmelzwaal in NeuralWorks Professional II/Plus SW package.

References

[1] Šnorek, M.: Generalized Fingerprint. in: Workshop of CTU Prague, January 20.-22. 1997, Praha, 1997, pp.305-306

[2] Mitchel J.S.: Machinery Analysis and Monitoring. PennWell Publishing Company, Tulsa, Oklahoma 1981, ISBN 0-878 14-145-6

[3] Frietman, E.E.E.: Guarding Vibration Patterns of a Mechanical Assembly with the help of a Neural Network in: Massively Parallel Processing Applications and Development, Elsevier 1994

[4] Medema, J-P., Remmelzwaal, M.: Fingerprint-based Diagnostic of Potentially "Bad" Students. Internal report of Dept of Computer Science and Engineering, CTU Prague and Department of Applied Mathematics, TU Delft, Prague/Delft 1996, 51 p.

[5] Macek,T.: Student Success Prediction by Means of Neural Nets. in: Workshop of CTU Prague, January 20.-22. 1997, Praha, 1997, pp.309-310

[6] Smit I., Rijswijk v.J., Frietman E.: A Neural Network Model of Organizational Structures. in: Eurosim '96, HPCN Challenges in Telecomp and Telecom, Elsevier 1996, pp: 267-277, ISBN 0 444 82559 2

[7] Mokrosz, R.: Fingerprint Case Study. Internal report of Dept of Computer Science and Engineering, CTU Prague, 1997, 19 p.

MAPS OF PROCESS DYNAMICS

Terhi Ylöstalo, Heikki Hyötyniemi

Helsinki University of Technology, Control Engineering Laboratory

Otakaari 5 A, FIN-02150 Espoo, Finland

Abstract

The visualization of the process data is an important factor in understanding process behavior. In this paper, both static and dynamic process information is mapped in a self-organizing map. This method is experimented in the modeling of the simulated Tennessee Eastman benchmark process.

Introduction

When the processes to be controlled are becoming more and more complex, understanding the process behavior is getting exceedingly important. When there are dozens of measurement variables, it is difficult to see their significance. Usually, many of the variables are mutually dependent and highly redundant, and the number of them can be reduced.

There are mathematical tools for achieving data compression. However, the mathematically motivated models with their computationally derived parameters do not help in understanding the process structure. An intuitively appealing approach to representing the dependencies between the process variables in a visual form is by using the self-organizing map (SOM) as presented in [Kohonen, 1995].

There exist various reports on how process state information can be mapped in SOM. Typically, only the static variables are mapped, or the measurement vector is directly used as input to the self-organization algorithm. This way, the operating regimes can be visualized, but no intuition on the process behavior within these regimes is obtained. From the control point of view, the dynamics of the process is a much more important factor than the static variable values themselves: the dynamics may remain invariant for a wide range of process variables, but for some special variable combinations, the dynamics changes abruptly. It is these regions of special dynamic phenomena that should be modeled in more detail.

In this paper, the dynamic nature of the processes is modeled in SOM. The adaptation of the map is based not only on the process measurements directly, but also on the identified process parameters. It is assumed that within each operating region the overall process model can be linearized, so that the process dynamics can be represented by a set of ARX model parameters. Using the identified parameters as input to the SOM algorithm results in a 'mode map'. The obtained model library could be applied for adaptive control of the process.

As an application, modeling of the Tennessee Eastman benchmark problem [Downs and Vogel, 1993] is discussed. This process is highly nonlinear, and its dynamic behavior in different operating points varies considerably. The experiments are based on a process simulator in the Matlab/Simulink environment.

The Tennessee Eastman process

The Tennessee Eastman (TE) process is a highly nonlinear, non-minimum phased, and open-loop unstable chemical process consisting of a reactor/separator/recycle arrangement. This process produces two products G and H from four reactants A, C, D and E. Also an inert B and byproduct F are present in the process. The simultaneous, irreversible and exothermic gas-liquid reactions are:

$$A(g) + C(g) + D(g) \rightarrow G(liq), \qquad \text{Product 1,}$$

$$A(g) + C(g) + E(g) \rightarrow H(liq), \qquad \text{Product 2,}$$

$$A(g) + E(g) \rightarrow F(liq), \qquad \text{Byproduct,}$$

$$3D(g) \rightarrow 2F(liq), \qquad \text{Byproduct.}$$
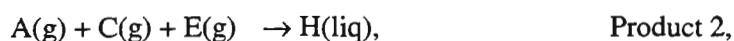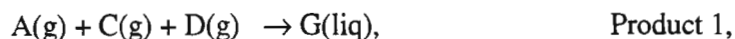
The process has 12 valves available for manipulation and 41 measurements available for monitoring or control. The detailed description of these variables, process disturbances and base case operating conditions, is given in [Downs and Vogel, 1993]. The process flowsheet is presented in Figure 1.
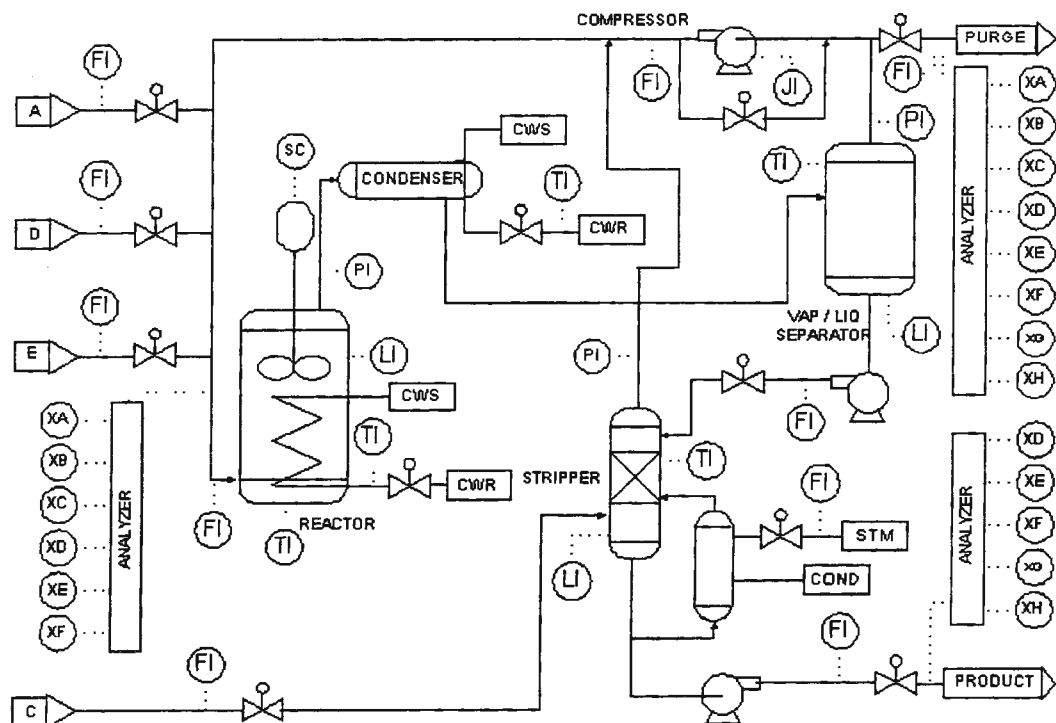
Figure 1. The Tennessee Eastman process flowsheet

The Tennessee Eastman process has been the subject of several studies [see for example Banerjee and Arkun, 1995, McAvoy and Ye, 1994, Ricker and Lee, 1995a and 1995b], but most of them have concentrated to the control of the process and the modeling of the process using input-output data has been mostly unexplored. The research work of [Sriniwas and Arkun, 1997] has made progress in this area by using input-output data of the process for the identification of an empirical model, further used to construct a Model Predictive Controller to control the process.

In the study of [Banerjee and Arkun, 1995] the process control scheme was divided into two tiers. The first tier controlled the critical variables that affected the reactor stability, and the second tier controlled the variables, which primarily affected the compositions of streams entering and leaving the reactor. In the work of [Sriniwas and Arkun, 1997] the first tier with its PID loops was used to maintain the process stability while four manipulated variables of the second tier (the set points for the controllers of reactor pressure, reactor level, D feed flow and E feed flow) were excited with input sequences and the sampled outputs of four controlled variables (reactor pressure, reactor level, product flow rate, and mass ratio of G and H in the product) were collected for the process identification. The same procedure is used in this work.

Identification of the process

In the work of [Sriniwas and Arkun, 1997] one general model was used as a reference model over the whole operating region of the TE process. However, because the process is highly nonlinear, this one single model may be too a crude approximation of the process dynamics in different operating regimes.The purpose of this current work is to identify models for the different operating points of the process to reach a set of more appropriate models for it. The same type of approach was used in the earlier work of the authors [Hyötyniemi and Ylöstalo, 1997]. In Fig. 2 typical responses for the step changes in the pressure are illustrated in different operating points. It seems that an oscillatory behavior emerges when the pressure rises.
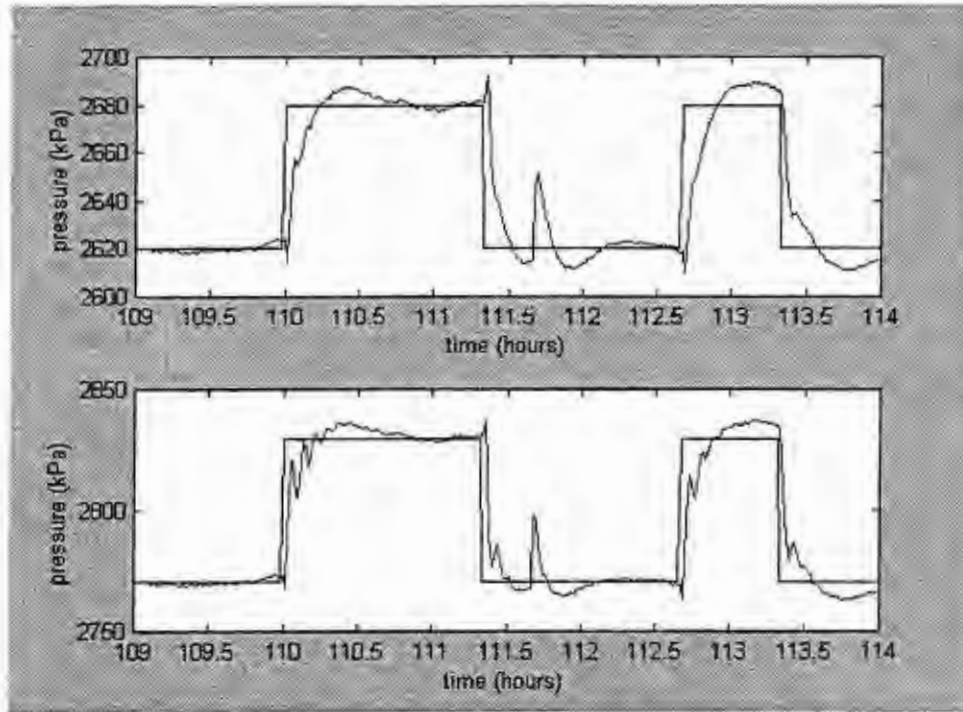
Figure 2. a) Signals u_1 and y_1 when the operating point was $p_0 = 2650$ kPa, and b) u_1 and y_1 when $p_0 = 2800$ kPa

The input-output data for the process identification was obtained by making step changes to the four different input variables in different operating points. These input variables were the reference values of the reactor pressure (u_1), the reactor level (u_2), the D-feed (u_3) and the E-feed (u_4) controllers. The simulation step size was 1 second and the data was collected once in a minute.

The structures of the four process models were chosen according to the work done by [Sriniwas and Arkun, 1997]. The models for the reactor pressure and the reactor level were SISO models and for the production rate and the product G/H ratio MISO models were used. In fact, as can be seen in Fig.2 (at time 111.5 h) there is an interaction between the pressure and some other process variable: it turns out that the setpoint of the reactor level has been changed at that point. Nevertheless, the simplified model structures of [Sriniwas and Arkun, 1997] were thought to be sufficient.

All the models were of ARX type with delay time 1 minute. The following models were used in the identification:

$$y_1(t) = -a_{11}y_1(t-1) - a_{12}y_1(t-2) + b_{11}u_1(t-1) + b_{12}u_1(t-2) + b_{13}u_1(t-3)$$

$$y_2(t) = -a_{21}y_2(t-1) - a_{22}y_2(t-2) + b_{21}u_2(t-1) + b_{22}u_2(t-2) + b_{23}u_2(t-3)$$

$$y_3(t) = -a_{31}y_3(t-1) - a_{32}y_3(t-2) + \sum_{k=1}^{3} b_{31k}u_1(t-k) + \sum_{k=1}^{3} b_{32k}u_2(t-k) + \sum_{k=1}^{3} b_{33k}u_3(t-k) + \sum_{k=1}^{3} b_{34k}u_4(t-k)$$

$$y_4(t) = -a_{41}y_4(t-1) - a_{42}y_4(t-2) + \sum_{k=1}^{5} b_{41k}u_1(t-k) + \sum_{k=1}^{5} b_{42k}u_2(t-k) + \sum_{k=1}^{5} b_{43k}u_3(t-k) + \sum_{k=1}^{5} b_{44k}u_4(t-k)$$

where $y_1(t)$ is the reactor pressure, $y_2(t)$ is the reactor level, $y_3(t)$ is the product flow rate and $y_4(t)$ is the mass ratio of G/H in the product. The sampling interval was 1 minute. The parameters a and b for the four models were identified in different operating points by using a standard least squares algorithm.

Constructing the maps

The self-organizing maps were used for the classification and organizing of the data, which consisted of the parameters of the four process models in the different operating points, and the numerical values describing the corresponding operating points, e.g. the reactor pressure, the reactor level, the D-feed into the reactor, and the E-feed into the reactor. The input vectors for the self-organizing maps included 50 parameters altogether, with four operating point values and 46 model parameters. All parameters in the input vectors were first scaled so that the standard deviations of the 50 different parameters were equal.

To visualize the difference between the 'mode maps' and the traditional maps where the process measurements are directly modeled, three different maps were constructed: in the first map, the model parameters and the operating point values were weighted equally, while in the other maps, either the model parameters or the process measurements alone were used. The parameters of the input vector for the second map were weighted so that the weight of the 46 model parameters become tenfold and the parameters of the input vector for the third map were weighted so that the weight of the four operating point parameters become tenfold. The first map is a mixture of a mode map and a traditional map, whereas the other maps are morelike examples of the two alternative basic approaches.

Two-dimensional maps with three by three neurons were used for modeling the process. The first map (Fig. 3) was initialized with random weight values.
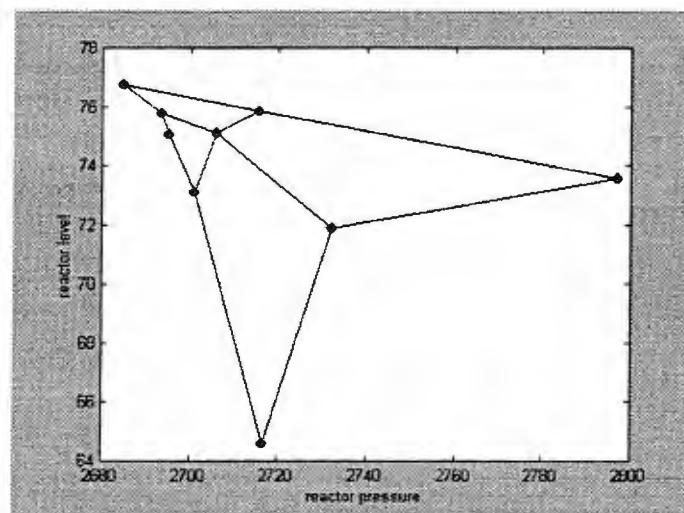
Figure 3. The first map, using the operating point values as well as model parameters as training data. The map is visualized by projecting it so that the corresponding reactor pressure and level are shown

The second and the third map (Figs. 4 and 5) were initialized with the weight matrix obtained as a result from the training of the first map. The maps were trained for 5000 cycles. The long training time was needed for the map to converge. All the calculations were done in Matlab using the Neural Network Toolbox [Matlab, 1994].

All the maps cover best the area near the base case values, since that is the optimal operating regime of the process and most of the data were concentrated there. Even if the set of operating points that were experimented and identified consisted of a rather narrow band of values, so that the dynamics did not vary remarkably, it seems that the maps clearly differ from each other. Using the model parameters in the map construction seems to be a good choice. The mode map extends wider over the variation range than the other two maps. This results from the fact that the mode map emphasizes the differences in the process dynamics. For example, when the pressure rises, the process dynamics becomes more oscillatory rather abruptly.

Abrupt changes in the process dynamics and therefore also in the model parameters are reflected in the final map that is stretched towards the dynamically 'more interesting' areas.
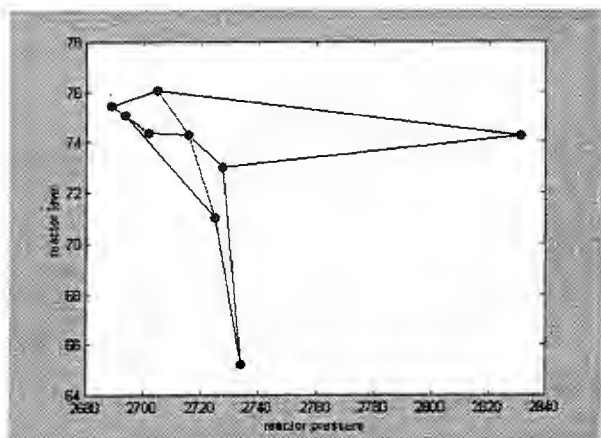
Figure 4. The second map, using the model parameters exclusively
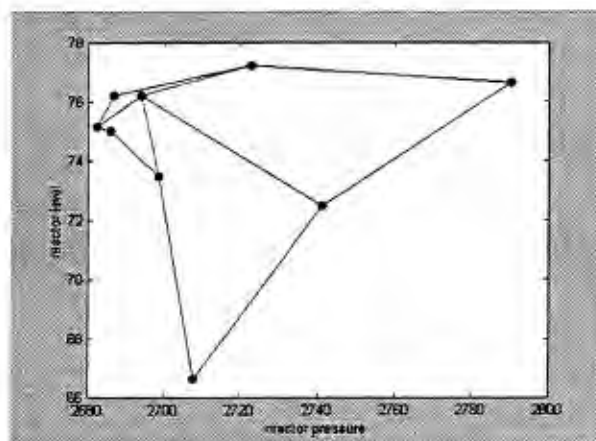
Figure 5. The third map, using only the operating point values

The map of models could be further used in the adaptive control of the TE process. In the earlier work of [Hyötyniemi and Ylöstalo, 1997] this type of control scheme was used for the control of a pH process.

Acknowledgements

References

Banerjee, A. and Arkun, Y. (1995). Control configuration design applied to the Tennessee Eastman plant-wide control problem. *Computers Chem. Engng*, **19**, 453.

Downs, J. J. and Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers chem. Engng*, **17**, pp. 245-255.

Hyötyniemi, H. and Ylöstalo, T. (1997). *Modeling and Control of a pH Process Using a Self-Organizing Map*. Preprints of the 11th IFAC Symposium on System Identification (SYSID'97), Fukuoka, Japan, July 8-11, pp. 559-564.

Kohonen, T. (1995): *Self-Organizing Maps*. Springer-Verlag, Berlin.

McAvoy, T. J. and Ye, N. (1994). Base control for Tennessee Eastman problem. *Computers chem. Engng*, **18**, pp.383-413.

Ricker, N. L. and Lee, J. H. (1995a). Nonlinear model predictive control of the Tennessee Eastman Challenge Process. *Computers chem. Engng*, **19**, pp. 961-981.

Ricker, N. L. and Lee, J. H. (1995b). Nonlinear modeling and state estimation for the Tennessee Eastman, Challenge Process. *Computers chem. Engng*, **19**, pp. 983-1005.

Sriniwas, G. R. and Arkun, Y. (1997). Control of the Tennessee Eastman process using input-output models. *J. Proc. Cont.*, **7**, pp. 387-400.

Matlab Neural Network Toolbox User's Guide. (1994), The MathWorks, Inc.

Model Semantics and Simulation of Time Scale Abstractions in Collision Models

Pieter J. Mosterman*

Inst. of Robotics and System Dynamics
German Aerospace Research Center
D-82230 Wessling
Germany
Pieter.J.Mosterman@dlr.de

Jan F. Broenink†

Department of Electrical Engineering
University of Twente
7500 AE Enschede
The Netherlands
J.F.Broenink@el.utwente.nl

Gautam Biswas‡

Department of Computer Science
Vanderbilt University
Nashville, TN 37235
USA
biswas@vuse.vanderbilt.edu

Abstract

Effective design of models requires simplification of different components of the system. For example, interactions of the system under scrutiny with its environment are modeled as idealized sources and sinks, and system behavior is derived from idealized interactions among constituent components. Some of the more detailed behavior is governed by small parameters, that produce continuous but complex nonlinear behaviors. These nonlinear continuous behaviors are often simplified to piecewise continuous forms, which then results in discontinuities at the temporal or spatial scale of interest. The resulting models are of a mixed continuous/discrete, or *hybrid*, nature. Physical systems theory based on *continuity of power* and *conservation of energy* is well understood. However, once discontinuous effects are included, these laws may be violated at the macroscopic level. The challenge is to endow hybrid models with semantics that do not violate the underlying physical nature of the system. This paper studies the effects and semantics of time scale abstraction, such as embodied by Newton's collision rule. The results show that hybrid modeling of physical systems can be systematic without the need for *ad hoc* hysteresis effects.

Keywords—**hybrid systems, bond graphs, modeling, simulation**

1 Introduction

Models are constructed to help analyze and design behavior of a system under investigation. An advantage of analyzing models is that one can focus on the system under consideration, and the environment in which the system operates can be made implicit to the analysis. However, most real-world system behaviors are governed by interactions with their environment, which makes it important to explicitly model these interactions. The part of the environment with which the system interacts is its *context*. Details of the rest of the environment are of no concern to the modeler. Because the only interest is the interaction with the system under investigation, the context can be specified by idealized sources and sinks [11].

Another important observation about the purpose of models is that they are constructed to analyze behaviors of interest. Most physical systems operate on a hierarchy of temporal and spatial scales. Often, the modeler's interest is not to analyze behavior at very fast time scales, but more to study the gross continuous system behavior at a macroscopic level. Just like specifying the details of the system context may not be of interest, the details of the fast time phenomena may be of little interest to the modeler, other than how it affects the system at the desired level of detail. Introducing abstractions that lead to simplified interactions between temporal and spatial scales often causes discontinuities in the behaviors at the macroscopic level. The resulting models cover mixed continuous/discrete behaviors, and are called *hybrid systems* [14; 17].

Physical system behavior is governed by the laws of *continuity of power* and *conservation of energy* [19] and the resulting continuous behavior exhibited by these systems is well defined. However, the more detailed continuous behavior, exhibited by complex systems is often nonlinear, and this complicates behavior analysis because [7; 13]:

- Well known linear system analysis techniques and

qualitative analysis techniques generate imprecise results.

- System order required to describe such behavior is much higher than that required to study the behavior of interest, and this can result in too much effort in designing models and in ill-constructed models. Further, knowing the correct values of system parameters is critical to analyzing the properties of the system model, yet the exact parameters of the microscopic phenomena are typically hard to obtain.

- Small parameter values producing behaviors at a very fast time scale require very small step sizes for numerical simulation algorithms. This causes the simulation algorithm to be computationally inefficient.

To overcome these difficulties, abstractions of the microscopic phenomena may be introduced into the system model. In previous work [16] we identified two types of abstractions in physical system models: (i) time scale abstraction, and (ii) parameter abstraction. Both time scale and parameter abstraction can cause discontinuities in system behavior. To illustrate, consider a bouncing ball. When the collision is considered perfectly elastic [2], the ball reverses its momentum the moment it makes contact with the floor. In reality, small elasticity effects in the ball and floor store the kinetic energy as potential energy and quickly return it during a short interval of time. The time scale of these phenomena is abstracted into a point of momentum reversal. When the collision is considered perfectly non-elastic, the ball loses all its initial momentum when it makes contact with the floor. In reality, small deformation and other dissipation effects (acoustic, thermal) are present that dissipate the ball's energy very quickly. However, because these parameters are abstracted away the ball appears to instantaneously lose its momentum. Note that this differs from the time scale abstraction where small elasticity effects were not abstracted away, but condensed into a point of time.

Good models of physical systems have a sound theoretical basis in that they are based on laws from the physical domain. The bond graph modeling language [3] provides a well defined formalism for expressing and developing physical system models where the basic physical principles, such as conservation of energy and continuity of power are directly enforced. When model abstractions require discontinuities to be introduced into physical system models, the basic laws that govern physical system behavior seem to be violated [8; 17]. Our goal in this paper is to go beyond a pure mathematical description for hybrid system models [1; 9; 10] by establishing physical semantics that govern the behavior of hybrid system models. These established

semantics are then incorporated into the bond graph framework to facilitate the building and analysis of hybrid system models.

In particular, this paper investigates the nature and effect of discontinuities created by introducing time scale abstractions to simplify models for 1-dimensional collision effects. Section 2 describes the collision chain phenomena in the device Newton's Cradle, and presents a mathematical representation of these in the form of *pinnacles* in state space [16]. It shows how these collision chains can be systematically modeled by the bond graph simulator *20sim* [5; 6]. Section 3 investigates the semantics for the collision points. Section 4 presents the conclusions and a discussion of this paper.

2 Newton's Cradle

Collision phenomena typically occur at time scales much smaller than the time scale of gross system behavior. Therefore, analysis of collisions use short duration *impulsive forces*, P, rather than the typical *forces*, F, observed in nature [2]. These impulsive forces result from abstracting the effect of collision forces that exist for very short intervals of time. In the limit, it can be assumed that the collision impulses act only for a point in time. At this point, the velocities of the colliding bodies instantaneously change from one value, the *a priori* velocity, to another, the *a posteriori* value.

2.1 Problem Analysis

Consider the collision between the two bodies in Fig. 1. In reality, when collision occurs, small elasticity effects store the kinetic energy over a short period of time and return the stored energy during a contiguous short interval of time. Because the time scale of this phenomenon is very small compared to the behavior of interest, it can be modeled as an instantaneous change at a point in time governed by Newton's collision rule [4]

$$v_2^+ - v_1^+ = -\epsilon(v_2 - v_1), \qquad (1)$$

where ϵ is the coefficient of restitution, which determines the loss of energy in the system. When $\epsilon = 1$ this represents an ideal or perfectly elastic collision where no energy is lost, whereas $\epsilon = 0$ represents a perfectly nonelastic collision where a significant amount of energy is lost. Note that this does not imply that all energy is lost for an inelastic collision.

Time scale abstraction, which results in the fast collision phenomena being reduced to a point in time, may create the impression that the conservation of energy principle is violated. But the physical principle of *conservation of state* still needs to hold despite the abstraction of a small behavior to a point. This principle is a generalized form of conservation of charge, momentum,
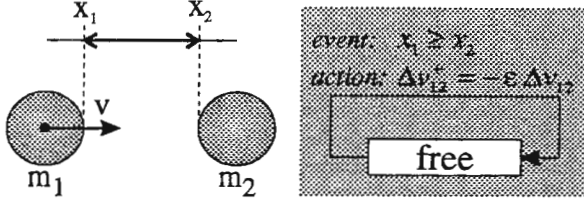
Figure 1: **A 1-dimensional collision between two bodies.**

etc. [17]. In this system, the generalized state is momentum and conservation requires $(m_1 = m_2)$

$$v_2^+ + v_1^+ = v_2 + v_1 \qquad (2)$$

Combined with the collision rule in Eq. (1) this results in the instantaneous change of velocity upon collision

$$\begin{cases} v_1^+ = \frac{1-\epsilon}{2}v_1 + \frac{1+\epsilon}{2}v_2 \\ v_2^+ = \frac{1+\epsilon}{2}v_1 + \frac{1-\epsilon}{2}v_2 \end{cases} \qquad (3)$$

This algebraic relation only holds at a point in time. As a result, the application of this relation may cause a discontinuous change in the velocities v_1 and v_2. Therefore, these changes have to be implemented as transition actions, where the value change is accompanied by a change in the mode or state. This corresponds to a Mealy type state machine [20].

Fig. 1 illustrates the state machine implementation, where the collision action that changes v_1 and v_2 is executed as a self-loop when $x_1 \geq x_2$. However, the corresponding action does not disable the discrete transition, and in a corresponding implementation model this transition would be executed infinitely many times. In fact, no new configuration is realized once the transition condition is invoked. Given that discontinuous changes are instantaneous, no real time elapses during the transitions. This violates the principle of *divergence of time* that states that any physical system behavior always progresses in time [15].

To prevent this loop of instantaneous transitions, the assumptions under which the coefficient of restitution yields a correct approximation need to be investigated. In reality, this coefficient is a function of the impact velocity [2] and tends to 0 when this impact velocity becomes small. Therefore, an additional condition to ensure the validity of the approximation is given by $v_1 - v_2 > v_{th}$, i.e., the difference in velocities has to exceed a threshold for the elastic collision phenomena to occur. If this difference is taken to be an infinitesimal value ($v_{th} \to 0$), the additional constraint becomes $v_1 > v_2$. Now, when first $x_1 \geq x_2$, collision occurs and the velocities are modified according to Eq. (3). The new values violate the $v_1 > v_2$ constraint and no further discrete events follow.

This shows that switching specifications have to embody assumptions made with regard to the abstractions that result in discontinuous behavior. A necessary condition for incorporating these assumptions is the divergence of time principle. In previous work, we have employed a multiple energy phase space analysis to verify physical correctness of the hybrid models [14; 15]. In other work, *ad hoc* hysteresis effects are introduced to prevent the loop of instantaneous changes [4; 12]. Though this eliminates the problem during simulation, it may cause modeling misconceptions.

In general, to enforce divergence of time, one can either:

- model the discontinuous phenomena in more continuous detail, or
- strengthen the discrete switching conditions to ensure there is only one possible mode for a given state vector.

Adding more continuous detail is often undesired because it increases the computational complexity of the simulation task. To strengthen switching specifications modeling assumptions that lead to discontinuity have to be made explicit. For the collision example, divergence of time can be enforced by adding the $v_1 > v_2$ condition to the transition condition.

2.2 Simulation in 20sim

Hybrid system models mix continuous behavior evolution with discontinuities that can be attributed to configuration or *mode* changes. These mode changes are invoked by discrete events that are generated when during continuous simulation pre-specified threshold values are crossed. At this point, continuous simulation is halted, and the discrete model is executed. One discrete event may cause a sequence of further events, and continuous simulation resumes after the configuration changes have converged to a new consistent mode. When a mode change occurs, the state vector values may change discontinuously, or even change in size. In this paper, discontinuous state changes are explicitly specified by transition actions in the discrete model. Situations where discrete transitions cause changes in the state vector are dealt with elsewhere [13; 14].

To develop a simulation system for 1-dimensional collisions, we used the physical modeling and simulation package *20sim* [6]. *20sim* provides an elegant modeling environment for continuous physical system models. It incorporates both the bond graph and block diagram modeling formalisms. However, *20sim* does not support discrete event modeling, e.g., in the form of finite state machines and Petri nets. The challenge is whether physical system modeling and simulation packages such as
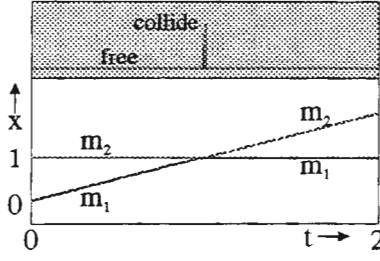
Figure 2: **Simulation of a collision between two bodies.**

20sim can be augmented to facilitate modeling and simulation of hybrid systems.

Presently, *20sim* has extensions that facilitate modeling of hybrid systems by explicitly constructing (sub)models as sorted assignment statements instead of unsorted equations. Furthermore, it offers a reset function for state variables. The collision models in this paper are formulated as assignment statement descriptions, that consist of two parts:

- the continous model differential equations of the bodies in the discrete *free* state.

- the finite state machine, separated into four parts (Algorithm 1):

  1. event detection,
  2. discrete state transition,
  3. transition actions, and
  4. restarting the continuous model part by solving the initial value problem of the newly inferred discrete state.

A *20sim* simulation result of the two colliding bodies in Fig. 1 is shown in Fig. 2. Body m_1 moves with a constant velocity towards m_2 that is at rest. Upon collision all of m_1's momentum is transferred instantaneously to m_2. Therefore, after the collision, m_1 is at rest and m_2 moves with a constant velocity.

Algorithm 1 FSM description of two bodies model in *20sim* version 2.2.

```
# FSM event detection: a collision
collide := (discrState == free) and
   collide == 0 and abs(distance) <= 0 and abs(v1 - v2) > 0
# FSM state transition: change the state of the FSM
discrState := if collide then free else discrState
# FSM action: at collide transition: momentum redistribution
newp1 := (m1 * (p1 + p2) - eps * m1 * m2 * (v1 -v2)) / (m1 + m2)
newp2 := p1 + p2 - newp1
dummy1 := resetfunc(collide, p1, newp1)
dummy2 := resetfunc(collide, p2, newp2)
# 'Restart' the IVP: recalculate the velocities,
# since the momenta changed after a state transition
v1 := p1 / m1
v2 := p2 / m2
```

From Algorithm 1 it is clear that it is straightforward to implement a graphical finite state machine editor that
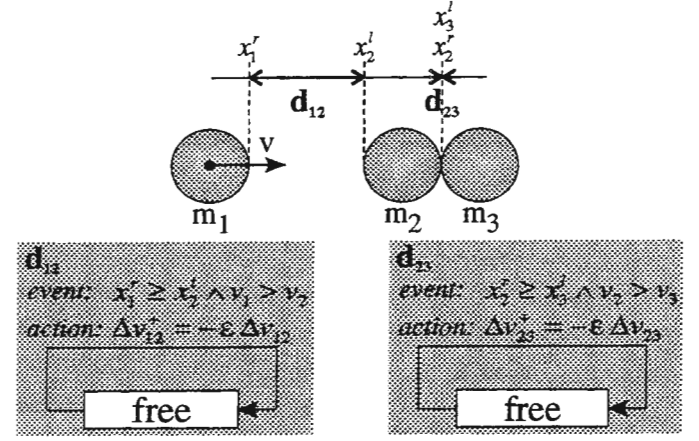
Figure 3: **A collision between three bodies.**

automatically generates discrete model code. Note that the *20sim* simulator currently only implements the state reset function without root-finding (i.e., locating the discrete event within a small temporal margin), which can lead to numerical inaccuracies, and make the result dependent on the integration method used.

2.3 Collision Chains

The described model handles one discontinuous change in state well. However, a number of collisions may occur, seemingly at the same instant in time. Consider a collision chain between three bodies (Fig. 3) where each transition action embodies Newton's collision rule. Fig. 4 shows a *20sim* simulation of the transfer of momentum between the three bodies. Note that body 2 takes on the complete initial velocity of body 1 before transferring it to body 3. This point in state space represents a so-called *pinnacle* and in simulation occurs at the same point in time where body 3 takes over the initial momentum [13; 16].

Pinnacles represent points in state space that are immediately departed when the new initial values are calculated. The reason for the immediate departure is the *dynamic coupling* between the discrete switching specifications. When the transition actions modify energy state variables, e.g., velocities, this may result in further transition conditions that are enabled. For example, in case of the three colliding bodies (Fig. 3) the $v_2 > v_3$ pre-condition for transition is satisfied immediately after the collision between m_1 and m_2 has changed v_2. All these discrete changes are executed at the same simulation time point. Once no further changes occur (given divergence of time this is guaranteed [14]), simulation of the continuous behavior resumes.

In previous work [13; 16] we have shown that such a sequence of mode changes can be attributed to time scale abstractions. The continuous state vector is mod-
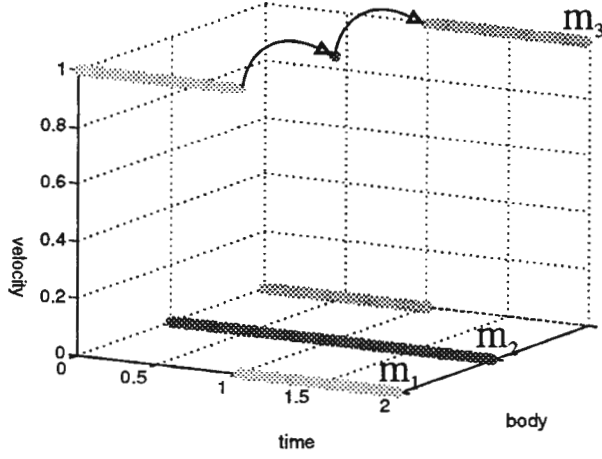
Figure 4: **Simulation of an ideal elastic collision between three bodies.**

ified as part of a transition action. Therefore, each mode in a sequence affects the system state, even though it is departed immediately after it becomes active (exemplified by the collision chains). A second class of mode changes occur because of parameter abstractions used to simplify nonlinear system behavior. The modes generated because of parameter abstraction correspond to virtual points in state space, i.e., they have no real existence on the time line, and, therefore, the mode is departed before the state vector is affected. Such mode changes are discussed elsewhere [14; 15].

3 Collision Semantics

In case of multiple collisions, several pinnacles may be traversed which in simulation all occur at the same point in time. However, semantically these points do not coincide but they follow each other immediately in time. This can be illustrated by investigating the underlying physical behavior. In reality, small elasticity coefficients are present that store the kinetic energy before passing it on, which requires a small amount of time. When this elasticity is modeled (Fig. 5), simulation shows that the point where body 2 takes on the velocity does not coincide with the point where velocity has been transferred to body 3 ($m_1 = m_2 = m_3 = m$). No matter how small the coefficient of elasticity, which governs the rate of transfer, body 2 captures the initial momentum from body 1 before it passes it on to body 3.

Fig. 6(a) shows this behavior for a perfectly elastic collision, $m_1 = m_2 = m_3 = m$. After collision, m_1 comes to rest. If the collisions were modeled to occur at the same instant in time, m_2 and m_3 would act as one body with mass $2m$, and behavior would be distinctly different. Fig. 6(b) shows a simulation of this phenomenon. In
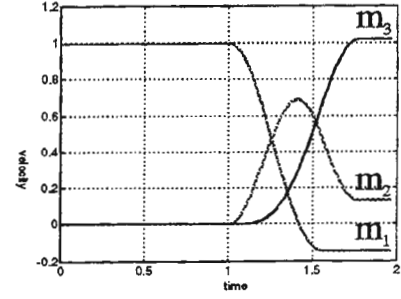
Figure 5: **An elastic collision between three bodies with small elasticity coefficients.**
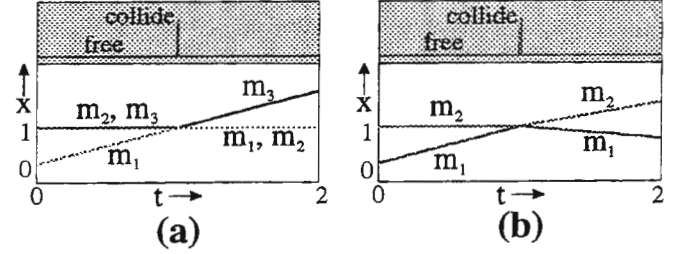
Figure 6: **Differences in behavior of one body, m_1, with mass m when (a) colliding with two bodies of equal mass, m, and (b) one of mass $2m$.**

this scenario, m_1 collides with body, m_2, with twice the mass of m_1, and m_1 obtains a negative velocity instead of coming to rest.

To investigate the consistency of the hybrid models, consider the collision in Fig. 7. If $m_1 = m_3 = m$ and $m_2 = 2m$, m_1 has an initial velocity $v_1 = v$, and velocity of m_3 is $v_3 = -v$. Depending on the initial positions, three scenarios may occur:

1. m_1 collides with m_2 first. In this case, $v_1^+ = -\frac{1}{3}v$ and $v_2^+ = \frac{2}{3}v$ after which m_2 collides with m_3 and $v_2^+ = -\frac{4}{9}v$ and $v_3^+ = \frac{11}{9}v$.

2. m_3 collides with m_2 first. In this case, $v_3^+ = \frac{1}{3}v$ and $v_2^+ = -\frac{2}{3}v$ after which m_2 collides with m_1 and $v_2^+ = \frac{4}{9}v$ and $v_3^+ = -\frac{11}{9}v$.

3. m_1 and m_3 collide with m_2 at exactly the same time. In this case the momentum transfers through m_2 between m_1 and m_3 and $v_1^+ = -v$, $v_2^+ = 0$ and $v_3^+ = v$.

These scenarios depict three very distinct behaviors, depending on the order in which the collisions occur as determined by the initial distances between the bodies. This sensitivity to a small perturbation in initial conditions (in this case the distance between the bodies) is inherent to hybrid systems in particular and nonlinear systems in general. In this system, the continuous state
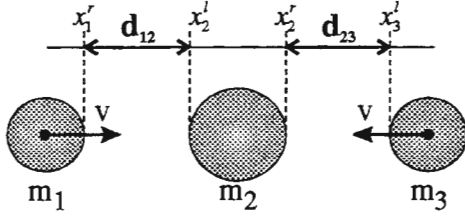
Figure 7: **Collision between three bodies with different mass.**
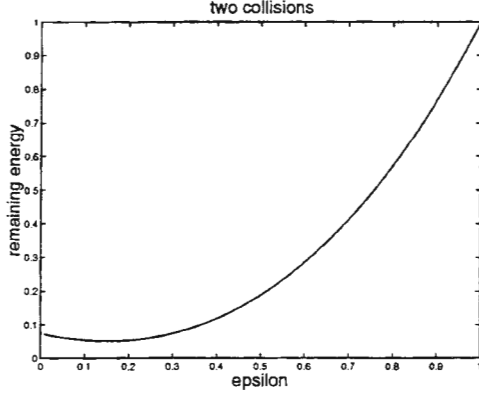
Figure 8: **Energy remaining after two collisions for different ϵ.**

Figure 9: **Collision between four bodies with different mass.**
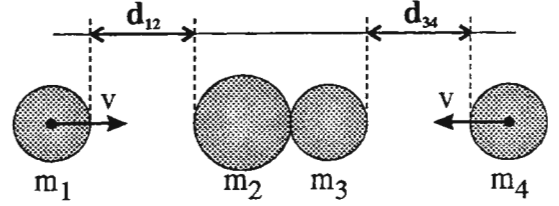
Figure 10: **Another collision between four bodies with different mass.**

space is partitioned into three segments, each with distinctly different behavior. A small change in initial condition may move the system from one segment into the other. In reality, the boundaries between these segments are not as crisp as in the idealized situation because of higher order, nonlinear, continuous effects. Therefore, when the distance of m_1 to m_2 and that of m_2 to m_3 are close but not quite equal, the actual behavior is somewhere in between both extremes. Future research includes finding correct physical semantics for hybrid systems that are less sensitive to the initial conditions. This may be possible based on energy analysis. Note that the energy in the system after two collisions is a function of the coefficient of restitution, Fig. 8. The degradation of the quality of the collision approximation is shown by the behavior of this function for small ϵ. When $\epsilon = 0$, the maximum loss of energy should be obtained. However, the function has a minimum around $\epsilon = 0.154$.

Now that we have a system sensitive to the exact time of collision, we can investigate what semantics yield a consistent formulation. Consider the system in Fig. 9 with $m_1 = m_3 = m_4 = m$, $m_2 = 2m$. If m_1 and m_4 are positioned such that the distance between m_1 and m_2 (d_{12}) equals the distance between m_2 and m_4 (d_{24}) minus the diameter of m_3 (two times its radius r_3), i.e.,

$$d_{12} = d_{24} - 2r_3, \qquad (4)$$

and when m_1 and m_4 are given an initial velocity with equal magnitude, v, but opposite directions, m_4 first collides with m_3. If the collision is assumed perfectly elastic, m_3 takes on all of the intial momentum of m_4. Next, one of the three previously discussed scenarios occurs. If the collision between m_3 and m_4 occurs in 0 time, then m_1 and m_3 collide with m_2 at the exact same time, momentum transfers through m_2 without affecting its velocity, i.e., it remains at rest.

A special case of the collision experiment between the four bodies is presented in Fig. 10. Now, m_3 is positioned such that it is in contact with m_2, $d_{23} = 0$. When m_1 and m_4 are located as before, $d_{12} = d_{34}$, and the collision between m_4 and m_3 coincides with the collision between m_1 and m_2. However, this latter collision causes m_2 to obtain a new velocity $v_2 \neq 0$. Therefore, the behavior is distinctly different, though for consistent model semantics it should be the same. Different behavior for $d_{23} = 0$ could be justified by attributing this to m_2 and m_3 being connected. However, if m_2 and m_3 are truly connected bodies, upon collision they act as one body with mass $m_2 + m_3$ and this results in yet a different behavior. This behavior conflicts with the collision model that results in correct behavior for the collision in Fig. 3 where m_2 first takes on all initial momentum before passing it to m_3, i.e., they are separated.

To obtain consistent semantics, it is reiterated that the collisions are phenomena that are abstracted to occur in an infinitely small interval. This implies that when m_1 and m_4 are positioned at equal distances from m_2, accounting for the diameter of m_3, their collisions with m_2 do not coincide. In fact, m_1 collides first, which is

235

exactly what is obtained when m_3 is positioned to be in contact with m_2.

This thought experiment shows that in order to achieve consistent hybrid systems, the semantics of configuration changes due to time scale abstraction should be implemented to occur in an infinitesimally small amount of time. This is in contrast with the mathematical hybrid systems approach [1; 9; 10] where all configuration changes are considered to occur in 0 time and, therefore, result in inconsistent models.

4   Conclusions

Physical systems operate on a hierarchy of temporal and spatial scales. Much like interactions with the environment, the detailed, continuous, behavior can be modeled by ideal effects. Often this results in discontinuities and the corresponding models are of a mixed continuous/discrete, hybrid, nature. In previous work we have identified two types of abstraction that lead to discontinuous behavior: (i) time scale abstraction and (ii) parameter abstraction [13; 16]. In this paper we discuss simulation and model semantics of time scale abstraction in detail.

Hybrid models exhibit continuous behaviors within modes of operation, and discrete transitions between modes when threshold value are crossed. During these discrete changes, conservation of energy and continuitiy of power appear to be violated. In previous work we have established the principle of *divergence of time* as a necessary condition for physical correctness of discrete mode changes [17]. This paper shows how this principle aids in revealing implicit modeling assumptions which yields insight in the modeling abstractions. Furthermore, it shows how these assumptions, when made explicit, result in models that are consistent with physical principles. In other work, *ad hoc* hysteresis effects are introduced to facilitate the simulation task [4; 12]. Though this can be physically justified, it cannot be linked to the modeling assumptions that form the basis for inconsistencies.

Due to dynamic coupling, one discrete change may invoke a number of consecutive changes. We show how the semantics of these changes caused by time scale abstraction, have to rely on infinitely small intervals of time $(\Delta t \to 0)$ to result in consistent models. This differs from the mathematical approach where such discrete changes are said to occur in 0 time, and sequences of changes occur at the same point in time, distinguished by an additional index that represents their place in the transition sequence [1; 9; 10].

The examples in this paper illustrate that hybrid systems are sensitive to small perturbations in initial conditions. In general, this is a characteristic of nonlinear systems, where entirely different behavior may ensue as a result of a small perturbation in the initial value. However, in reality this sensitivity is much less extreme. Presently, these phenomena are not facilitated by state of the art simulation tools. Future research focuses on physical semantics for hybrid behavior closer to reality and an implementation in modeling and simulation tools such as *20sim* to obtain robust hybrid system simulation results. This research is related to exloiting the effect of small parameter values in sliding mode semantics [18], which allows behavior generation for a mathematically singular plane in state space. Eventually, we aim to develop an extensive ontology for hybrid models of dynamic physical systems. Furthermore, we will enhance *20sim* with proper graphical editor facilities to elegantly specify hybrid systems with finite state machines, and add exact event localisation to the simulation engine.

References

[1] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Lecture Notes in Computer Science*, volume 736, pages 209–229. Springer-Verlag, 1993.

[2] Raymond M. Brach. *Mechanical Impact Dynamics*. John Wiley and Sons, New York, 1991.

[3] P.C. Breedveld. Multibond graph elements in physical systems theory. *Journal of the Franklin Institute*, 319(1/2):1–36, January/February 1985.

[4] Peter C. Breedveld. The context-dependent trade-off between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects. In *CESA '96 IMACS Multiconference*, Lille, France, July 1996. Ecole Centrale de Lille.

[5] Jan F. Broenink. *Computer Aided Physical Modeling and Simulation: A Bond Graph Approach*. PhD dissertation, University of Twente, Enschede, The Netherlands, 1990.

[6] Jan F. Broenink. Modelling, simulation and analysis with 20-sim. *Journal A*, 38(3):22–25, January 1998. Special CACSD issue.

[7] Jan F. Broenink and Paul B.T. Weustink. A combined system simulator for mechatronic systems. In *Proceedings of ESM 96*, Budapest, Hungary, June 1996.

[8] Jan F. Broenink and Klaas C.J. Wijbrans. Describing discontinuities in bond graphs. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 120–125, San Diego, California, 1993.

[9] Akash Deshpande and Pravin Varaiya. Viable control of hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 128–147. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[10] John Guckenheimer and Stewart Johnson. Planar hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 202–225. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[11] Derek J. Hatley and Imtiaz Pirbhai. *Strategies for Real-Time Systems Specification*. Dorset House Publishing Co., New York, New York, 1988.

[12] Yumi Iwasaki, Adam Farquhar, Vijay Saraswat, Daniel Bobrow, and Vineet Gupta. Modeling time in hybrid systems: How fast is "instantaneous"? In *1995 International Conference on Qualitative Reasoning*, pages 94–103, Amsterdam, May 1995. University of Amsterdam.

[13] Pieter J. Mosterman. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*. PhD dissertation, Vanderbilt University, 1997. http://www.op.dlr.de/FF-DR-ER/staff/pjm/papers/thesis/p.html.

[14] Pieter J. Mosterman and Gautam Biswas. A Formal Hybrid Modeling Scheme for Handling Discontinuities in Physical System Models. In *AAAI-96*, pages 985–990, Portland, Oregon, August 1996. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025.

[15] Pieter J. Mosterman and Gautam Biswas. Verification of dynamic physical system models. In *ASME-96*, pages 707–714, Atlanta, GA, November 1996.

[16] Pieter J. Mosterman and Gautam Biswas. Principles for Modeling, Verification, and Simulation of Hybrid Dynamic Systems. In *Fifth International Conference on Hybrid Systems*, Notre Dame, Indiana, September 1997.

[17] Pieter J. Mosterman and Gautam Biswas. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute*, 334B(6), 1997.

[18] Pieter J. Mosterman, Feng Zhao, and Gautam Biswas. Model semantics and simulation for hybrid systems operating in sliding regimes. In *AAAI Fall Symposium on Model Directed Autonomous Systems*, 1997.

[19] Henry M. Paynter. *Analysis and Design of Engineering Systems*. The M.I.T. Press, Cambridge, Massachusetts, 1961.

[20] K.C.J. Wijbrans. *Twente Hierarchical Embedded Systems Implementation by Simulation: a structured method for controller realization*. PhD dissertation, University of Twente, CIP-DATA Koninklijke Bibliotheek, Den Haag, The Netherlands, 1993. ISBN 90-9005933-4.

237

FEMLAB® — A MATLAB-BASED ENVIRONMENT FOR MULTIPHYSICS MODELING AND SIMULATION

Magnus Ringh[1], Svante Littmarck[2]

[1]Computer Solutions Europe AB, Björnnäsvägen 21, S-113 47, Stockholm, Sweden , magnus@comsol.se, www.comsol.se
[2]Computer Solutions Europe AB, Björnnäsvägen 21, S-113 47, Stockholm, Sweden, svante@comsol.se, www.comsol.se

1. Introduction

FEMLAB® is a MATLAB®-based environment for modeling and simulation of a wide range of problems in engineering and physics. MATLAB with toolboxes and Simulink is an integrated technical computing environment. The set of toolboxes covers application areas such as optimization, signal processing and control design and validation. FEMLAB provides powerful and flexible tools for modeling, simulation, and solution of physical systems. The FEMLAB environment includes a comprehensive graphical user interface and command-line functions for preprocessing, solving, and postprocessing models using the Finite Element Method (FEM). Support for multiple, coupled equations and integration with Simulink and the Control System Toolbox make FEMLAB an ideal tool for solving multiphysics and multidisciplinary problems. In this paper we will look at examples of FEMLAB applications in structural mechanics, control design, electromagnetics and computational fluid dynamics (CFD).

2. FEMLAB: an overview

The Model Space

The current version of FEMLAB solves partial differential equations (PDE) in two space dimensions and time. The PDEs are of the form

$$d \frac{\partial u}{\partial t} - \nabla \cdot (c \otimes \nabla u + \alpha \otimes u - \gamma) + \beta \otimes \nabla u + au = f$$

with the boundary conditions

$$n \cdot (c \otimes \nabla u + \alpha \otimes u - \gamma) + qu = g + h' \lambda$$
$$hu = r$$

FEMLAB also supports a more general form of the PDE that is well suited for strongly nonlinear problems

$$d \frac{\partial u}{\partial t} + \nabla \cdot \Gamma = F$$

with the boundary conditions

$$-n \cdot \Gamma = G$$
$$0 = R$$

Here, u is a complex-valued function defined on a bounded region Ω. In general, all coefficients can depend on the space coordinates x and y, the time, t, the solution u, and its gradient.

Many interesting physical applications live in this PDE space. See below.

The User Interface

The software is designed to work from a mathematical description. But we do not require that the end user is a skilled mathematician; we provide application oriented user interfaces and a ModelNavigator™.

The Model Navigator offers physics modes with familiar problem parameters. Application areas include:

- Structural mechanics

- Computational fluid dynamics

- Electromagnetics

- Heat transfer

- Chemical reactions

- Physics (quantum mechanics, ...)

- Wave propagation

- Multiphysics and multidisciplinary problems

- A set of mathematics modes provides general PDE modeling from the GUI

Specialized Modules

Specialized modules extend the capabilities of FEMLAB. The Structural Mechanics Module includes specialized finite elements such as beams, bars, and plates; special nonlinear solvers for problems with plastic deformation or large displacements; material libraries; additional postprocessing facilities such as principal stress plots and beam plots.

Models

From the Model Navigator, the user can design a new problem by selecting the specific application area mode, or load an existing model from a library. A comprehensive model library comes with FEMLAB. It contains a number of examples from a wide range of applications. Each model is presented in the Model Navigator with a descriptive picture and text. Models can be saved at any stage during the modeling phase or after simulation, with or without simulation results. The FEMLAB model library is intended to serve as a starting point for exploring the tool itself or a certain application area. It's easy to modify a model, changing the geometry for instance, and then simulate the new problem. A model can be saved as a datafile, a model Mat-file, and/or represented in a script language, a model M-file. The latter makes it easy to put the model in a program loop to run sequencies of models varying parameters of interest. See example 3 below.

Solvers

Powerful nonlinear state-of-the-art solvers, with automatic and adaptive meshing capabilities and error control, constitute the computational engine in FEMLAB. Special solvers include an eigenvalue solver, a DAE solver for nonlinear time dependent problems, and a multigrid solver for large scale problems.

CAD tools in the GUI

The geometry modeling supports boundary modeling and constructive solid geometry modeling. The basic geometical objects include rectangles, ellipses, points, lines, arcs, and 3rd order Bézier curves. More complex geometrical objects can be created by combining objects using set algebra. Also, users can import existing CAD geometries on DXF-file format into FEMLAB.

3. Examples

Example 1: A plasticity problem

In this example we study the stresses and deformations in a mechanical component that is subjected to such a high load that plastic deformation occurs. The detail is constrained in the left part, and we apply a load F on the vertical edge to the right (see figure 1). The load is applied in steps up to the maximum load of 25. It is then removed again in steps down to zero. The effective stresses and strains will vary in a nonlinear fashion. First the stress varies linearly with the strain up to the yield stress $s0$. Above the yield stress plasticity occurs, and when the load is removed in this region, there will be remaining deformations and strains.

The drawing was originally done inAutoCAD, and is imported into FEMLAB using the DXF-import facility.
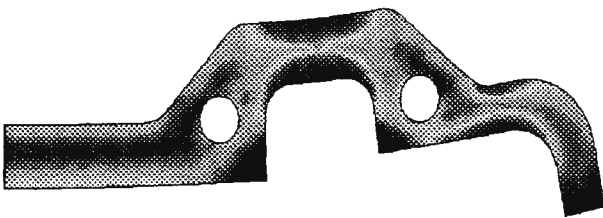
Figure 1. Effective stress and deformation of component at maximum load.

In plot mode we can see the stress level after each of these load steps. In the third load step some plastic deformation has occurred, and in the fourth load step the structure approaches breakdown. As the load is decreased again, the stress decreases, but it does not go back to zero in the parts that had reached the plastic region, even when the load is completely removed.

Example 2: Control design using FEMLAB andSimulink

In dynamical models it is very common with physical subsystems. For example, a model of a temperature control system with a thermal controller that controls the temperature in a room. The heat flow in the room is then governed by physical equations. Traditionally, simplified empirical models were used for the physical systems, e.g., in the design of controllers. But important properties can be lost after such simplifications, and better results can be achieved by using FEM models, and solving the real physical equations.

Integrating FEMLAB with Simulink makes it possible to use the block libraries is Simulink for applications like advanced control design and digital signal processing, together with finite element models.

In this example, a shelf is used to store boxes on. Boxes are continuously added and removed, so the load on the shelf will change over time. But the shelf will deform differently depending on the total load, and therefore the height will also vary. To compensate for this, a supporting force is applied, and the goal is to control this support force so that the height is constant.
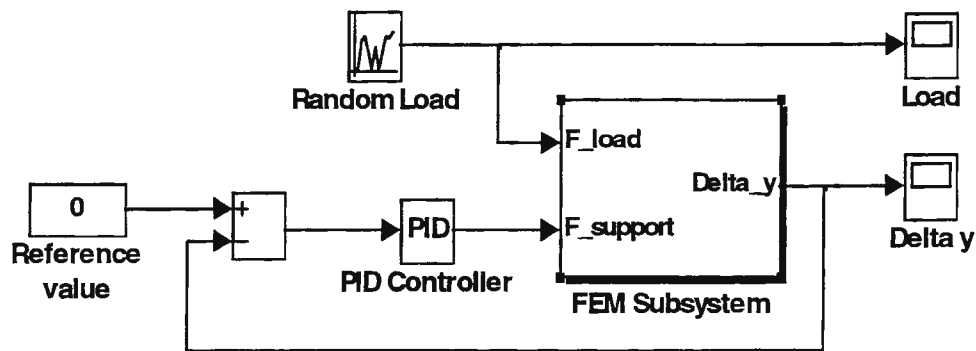
Figure 2. Simulink block diagram of control simulation with FEM subsystem modeled in FEMLAB.

Example 3: A wave guide

This example illustrates electromagnetic wave propagation in a rectangular wave guide. The waves are in the microwave range, and are used for transmitting microwaves with small losses (low attenuation).

To transmit microwaves from a transmitter unit to an antenna, wave guides are used. Parts of it could have elliptical cross-section of flexible material, and parts of it could have rectangular cross-section of a stiff material. Here we model a rectangular wave guide section. The air filled region in the wave guide is modelled by the wave equation. Solving for harmonic waves eliminates time, and the final model equation is a Helmholtz's equation. At the entrance, energy is pumped into the wave guide. That is, the entrance boundary works as a source. The operational range, frequencies that are sufficiently transmitted, of the wave guide is of great importance. We simulate a sequence of models varying the frequency and then compute transmitted energy.
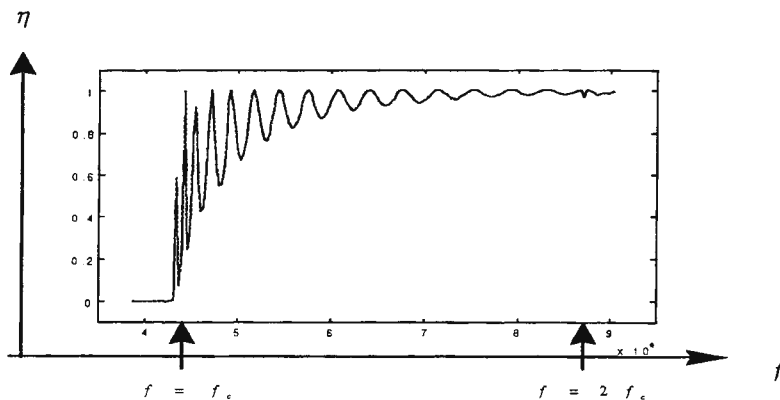
Figure 3. Simulation results in a parameter space: Transmitted energy as function of frequency.
A model sequence with 200 equally spaced frequencies reveals the transmission phenomena.
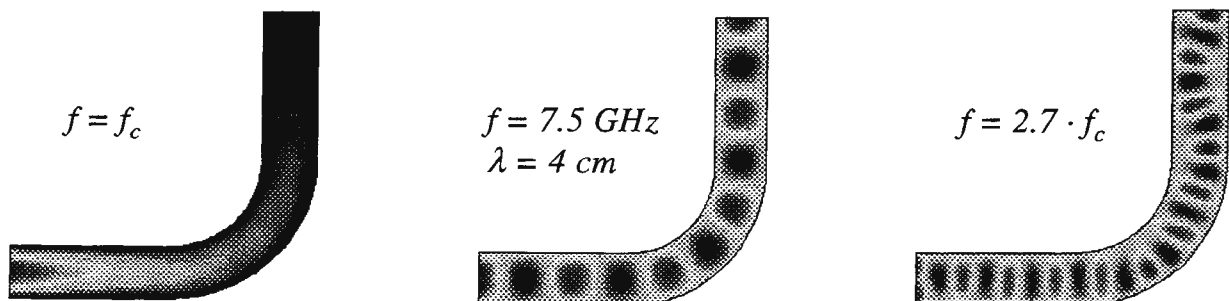
Figure 4. E-Wave propagation at cut-off frequency, within operational range, and at 2.7 x cut-off frequency

Example 4: Flow around a truck

The governing equations in computational fluid dynamics are the Navier-Stokes equations. An important special case of these are the incompressible Navier-Stokes equations. Incompressible means that we assume the density to be constant, and this is a reasonable assumption in many cases.

We solve for the velocities and the pressure. All the quantities are normalized using the scalar Reynolds number

The flow is visualized by an animated sequence. To visualize the flow we solve an "extra" problem: By setting a constant particle concentration in the inlet and solving a convection problem, the particles will be transported along the velocity field.

Figure 5. Air flow around a driving Volvo FH truck. Geometry: Courtesy of Volvo Truck Corporation.
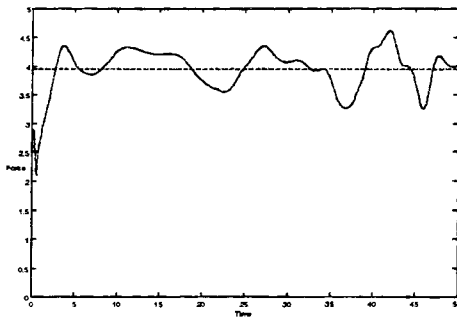
Figure 6. The form drag (wind force) as a function of time

4. Conclusions

FEMLAB is a powerful and easy-to-use tool for modeling and simulation of physical systems. Its combination of a state-of-the-art computational engine and the carefully designed user interface addresses the needs from beginner up to expert level. FEMLAB is open and extensible. The MATLAB foundation provides virtually unlimited postprocessing capabilities. Interfaces to MATLAB's control design and simulation environment makes it an ideal tool for multidisciplinary problems, where PDE based models can be integrated with dynamic systems.

Simulation-aided design of electromechanical systems as shown by the example of rolling mill drives

H. Cerv[1], M. Monse[2], V. Müller[2], S. Schulze[1]

[1]Mannesmann Demag AG - Metallurgy / Cold Rolling and Processing, D-40880 Ratingen
[2]TU Dresden, Institute of Electrical Power Engineering, D-01062 Dresden, vmueller@eeiwzb.et.tu-dresden.de

Abstract. In recent years, the use of simulation methods for the dimensioning of components for machine construction has been of ever-increasing importance. An optimum approach for the design of drives is to give equal consideration to electrical and mechanical system components in a calculation model. Using the example of rolling mill technology, we will explain the simulation of torsional vibrations as it aids the design process geared at optimizing drive configuration.

Introduction

The use of simulation for the dimensioning of machines in the field of mechanical engineering has been gaining constant ground in recent years. Only through system solutions adapted to the individual situation can the desire to reduce costs without compromising quality be fulfilled. Since in the rolling mill field almost every plant must be designed differently, the risk involved is often minimized at the draft stage by incorporating safety factors. But this approach brings with it additional weight and costs. This is why the comprehensive modelling and simulation of the electromechanical system in the time and frequency range already in the early stage of design work is of utmost importance when it comes to designing equipment in the cold and hot rolling fields. This calls for models responding to the specific tasks at hand. The article describes simulation-aided design as it applies to the investigation of torsional vibrations, with special emphasis being placed on characteristic quantities such as natural frequencies and dominating time constants as well as dynamic and stationary (static) loads.

The electromechanical system

The overview in Figure 1 shows the general structure of electrical drives which can also be found in rolling mill systems.
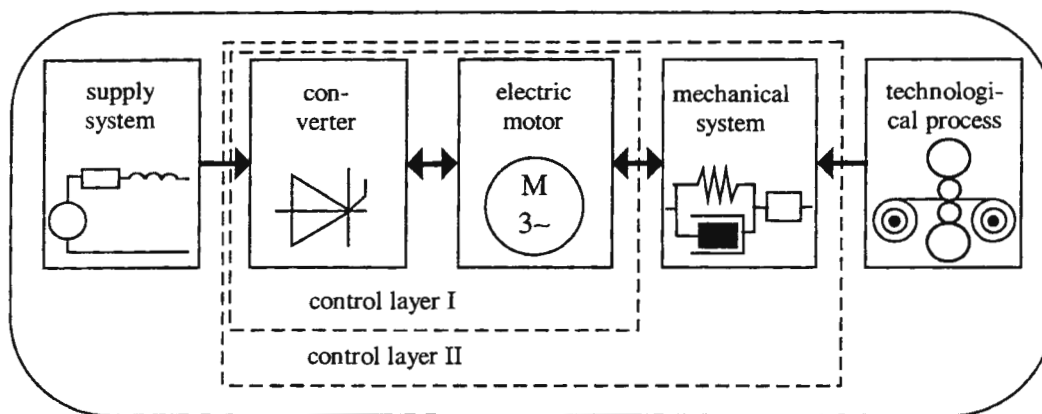
Figure 1: Principal layout of electromechanical drive systems

It illustrates the individual system components and the interfaces between them. The process to be examined consists of the following units: motor, converter and (motor-attached) control as well as the mechanical units. The control structure is usually made up of a speed control with cascade current control.

For the sake of limiting the complexity of the system, reactionless boundary lines are defined separating the energy supply network and the technological process. The energy supply network is considered to be stiff and the process behaviour of the working machine is represented by a static characteristic.

For the simulation of torsional vibrations, the structures of the mechanical simulation model are limited to representation with relative coordinates and a degree of freedom (rotational degree of freedom). The model consists of discrete energy accumulators and elastic transfer elements. The coupling of the structure elements is a result of the mechanical structure of the drive, which may vary greatly.
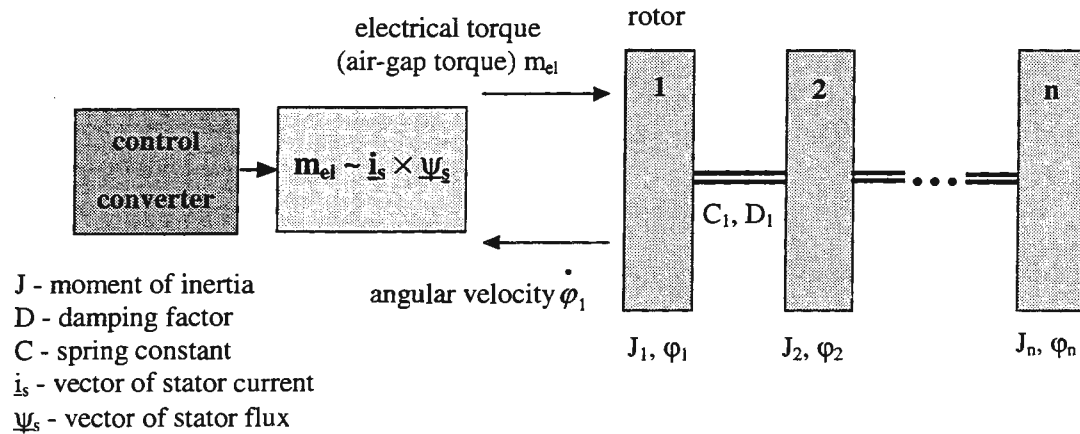
Figure 2: Coupling between the electrical and the mechanical subsystem

Of key importance is the representation of the coupling between the electrical and mechanical components (Figure 2). The air-gap torque produced by the electrical machine acts on the mechanical drive strand as an external excitation on the rotor mass. With the inclusion of the mechanical rotor mass in the mechanical model, all natural frequencies are duly considered. The feedback of the rotation of the mechanical drive strand is given by the motor speed. The current and speed control of the electrical machine is a further area to be considered.

The common simulation of mechanical and electrical components can basically be realized in two ways:

- Simulator coupling: Two partial programs are linked to each other via an interface. This method was rejected due to numeric problems and the complex mechanisms needed for synchronizing separate simulators.

- One program: This however calls for the model base to be extended and adapted to describe both parts of the subsystem. To this end an already available block-oriented simulation program (DS88) [1] was further developed by adapting the scope of its function.

With a uniform simulation, the mechanical and electrical (control) applications are united under one program. Block diagrams and circuit diagrams are used as common description tools. Thus a uniform system of differential equations of the electromechanical system is created which is solved by one simulator.

Modelling of the electrical subsystem

The focus of the model development was to derive generally applicable substitute systems which represent the electrical processes only in as much detail as is necessary. A higher resolution inevitably leads to very long calculating times and is attended with problems with regard to the determination of parameters. From the perspective of the simulation, the electrical components have, compared with the mechanical equipment in rolling mills, relatively small time constants (in the millisecond range) and are characterized by frequent

switching operations (pulse modulation of the power inverters). They also exhibit a pronounced mix of continuous and discontinuous system components (e.g. digital control).

For the three major functional groups of the electrical part system (electrical motor, power converter and control structure) special modelling properties were defined to examine the effects of the electrical components on the mechanical equipment during normal operation (e.g. torque harmonics) or as a result of worst-case disturbances.

Electric motor
- Models for DC motor, induction-type and synchronous motor [2]
- Fundamental-wave model of the symmetrically arranged machine with constant parameters
- Two-component representation
- Non-linearities in the magnetic circuit, eddy current losses, and current displacement are not considered

Inverter
- Models for rectifiers, direct converters and pulse-controlled inverters
- Sampling work method
- Representation of the power semiconductors as ideal switches
- A non-ideal commutation can be simulated by adjusting the commutation time

Automatic control
- Basic models for the control of DC, induction-type and synchronous motors
- Application of modern control procedures (field-oriented control for three-phase motors)
- Digital control
- With automatic parameterization of the control models by the system data

As an example for an inverter model, Figure 3 shows the block diagram of a pulse-controlled inverter.
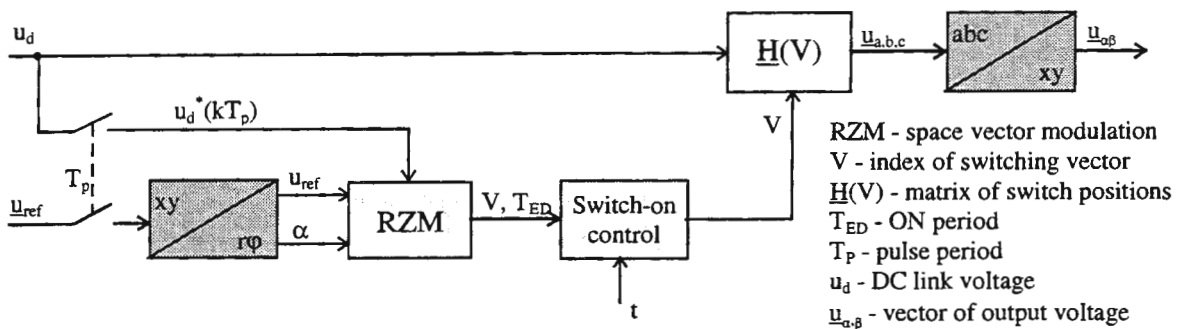
Figure 3: Block diagram of the model of a pulse converter with DC link

From the standpoint of system theory, the current inverter acts as an actuator providing the output voltage (with regard to amplitude and phase position) required by the reference signal of control system $\underline{u}_{ref}$ by modulating the DC link voltage u_d. The actuator works as a sampling element with a fixed pulse period T_p. The power semiconductors of the pulse-controlled inverter receive the switch-on and switch-off pulses from the control device (RZM) in coordination with the switch-on control. The pulses are generated in the control device in accordance with the space vector modulation method [3]. The subsequent function block couples switching vector with the index V and the DC link voltage u_d with the matrix of the possible switch positions $\underline{H}(V)$. Matrix $\underline{H}(V)$ is the result of the mesh equations of the voltage-inverter circuit [4] when considering the semiconductor valves as ideal switches. The block diagram in Figure 3 shows that the input variables from the Cartesian coordinate system (x, y) are transformed to the polar coordinate system with components (r, φ). After the calculation of the output voltage (phase voltage in the 120°-coordinate system), the variables are transformed back to the external (x, y) coordinate system.

In a extremely simplified model or draft model of the design of the controller structure, the pulse-controlled inverter can be described with a first-order time-delay element or lag element.

The oscillograms in Figure 4 and Figure 5 show phase voltages and phase currents at a pulse frequency of $f_p=1/T_p=1$ kHz.
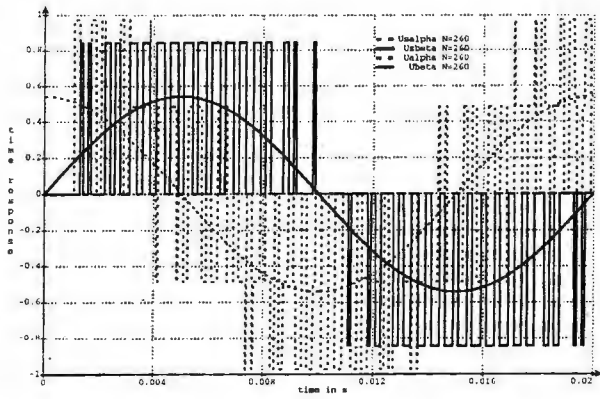
Figure 4: Reference and true value of the phase voltages
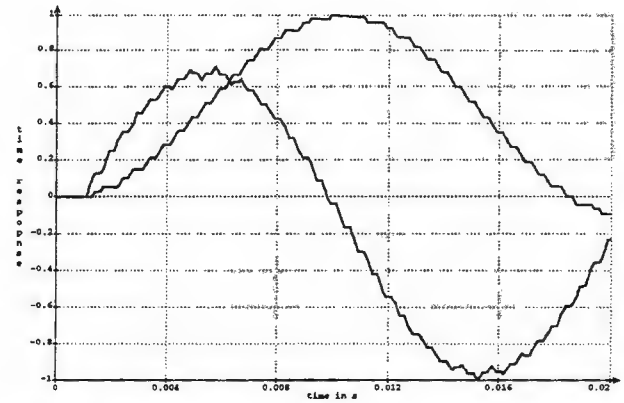
Figure 5: The phase currents contain harmonics (smoothing: $T_1=0.1$ s)

Modelling of the mechanical subsystem

The mathematical substitute system of the mechanical real system is based on the breakdown of the drive strand into twist-free concentrated masses as well as non-mass elastic and damping connections. From the generally applicable equation of motion according to Newton it is possible to derive the mathematical description of the motion also for the rotation. Hence:
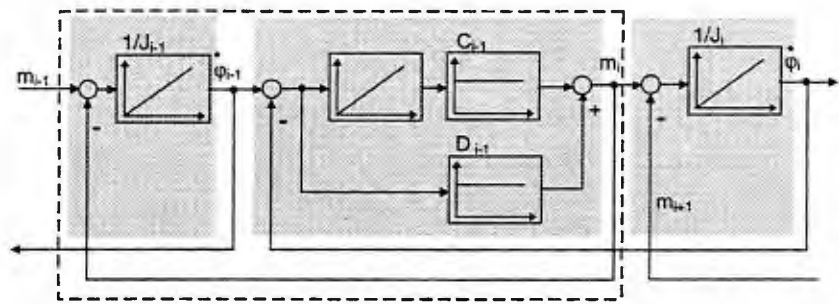
Figure 6: Signal flow chart of the mechanical subsystem

$$J_i \cdot \ddot{\varphi}_i + D_i \cdot \Delta \dot{\varphi}_i + D_{i-1} \cdot \Delta \dot{\varphi}_{i-1} + C_i \Delta \varphi_i + C_{i-1} \Delta \varphi_{i-1} = \sum m_i$$

with J - moment of inertia; D - damping factor; C - spring constant
φ - angle of rotation

This equation system can be transformed to a block diagram as in Figure 6.

The structure of the drive strand defines the structure of the coupling plan. For modeling, straight, branched, and meshed structures can be selected as desired. Assuming constant parameters, a linear system is created. When taking into account transmission elements such as gears, coupling and cardan shaft, even non-linear systems can be realized. In these models, the parameters of the transmission function are in some cases constantly calculated anew during the simulation (parameter excitation) [6].

Simulation of the overall system

The structure of the mechanical substitute system can be derived from the design drawings. With the graphical editor in *DS88* it is input as a block diagram. By means of a preprocessor the system data for the

simulation (e.g. moment of inertia, damping factor, spring constant) can be calculated from the basic data (geometrical data, material data). If the data of the electrical substitute system are not available, the electrical preprocessor calculates from rated data and rating plate data approximated values for the first simulation runs. The survey in Figure 7 shows the data flow in the design cycle.
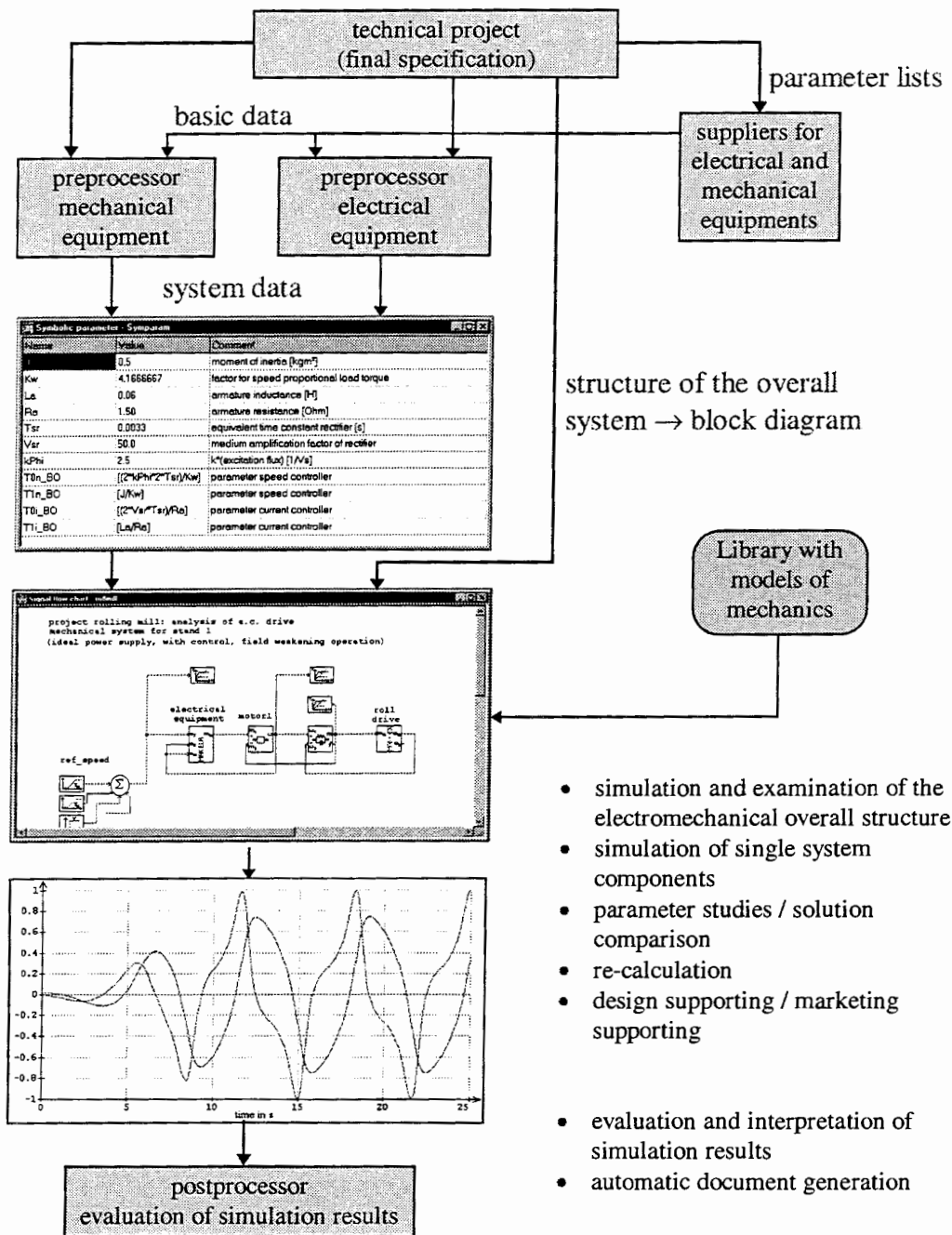
Figure 7: : Survey about the data flow in the simulation-aided design
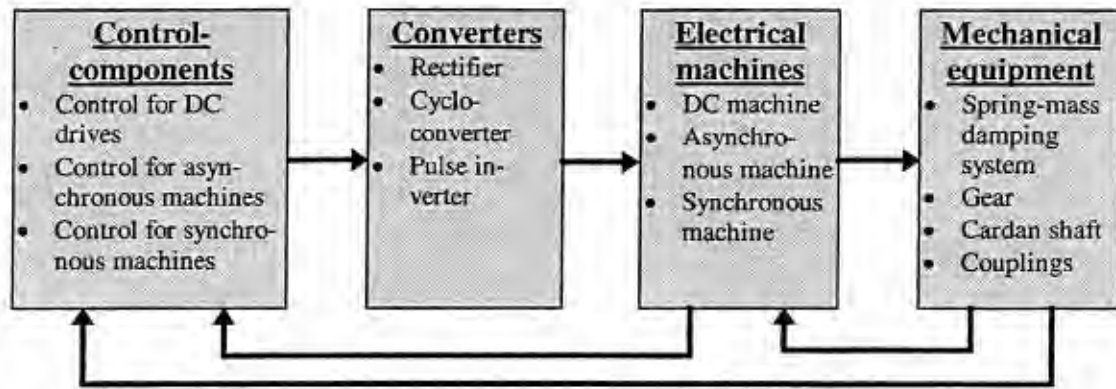
Figure 8: Block library for the simulation of electromechanical systems

For the simulation of electromechanical systems, a library of complex elementary blocks (e.g. electrical machines) and macroblocks (e.g. control structures) were created.

This in turn resulted in a modular system of user-specified functions of mutually exchangeable and combinable substitute systems enabling the analysis of a many different tasks for simulation of torsional vibrations. Figure 8 gives an overview. By a simple exchange of substitute systems it is possible, for example, to investigate how the system behaviour is affected when the direct current drives are replaced by three-phase-current drives for the same mechanical equipment. The user does not have to come to grips with the details of the substitute system creation, but rather can take advantage of existing and proven structures. He should however be familiar with the possibilities and limits of the substitute systems.

Application example

We shall take the first stand of a physical rolling mill drive of a cold rolling mill as an example. The mechanical structure is represented by a substitute system simplified to 11 masses with a motor and a pinion gearing for power distribution (Figure 9).
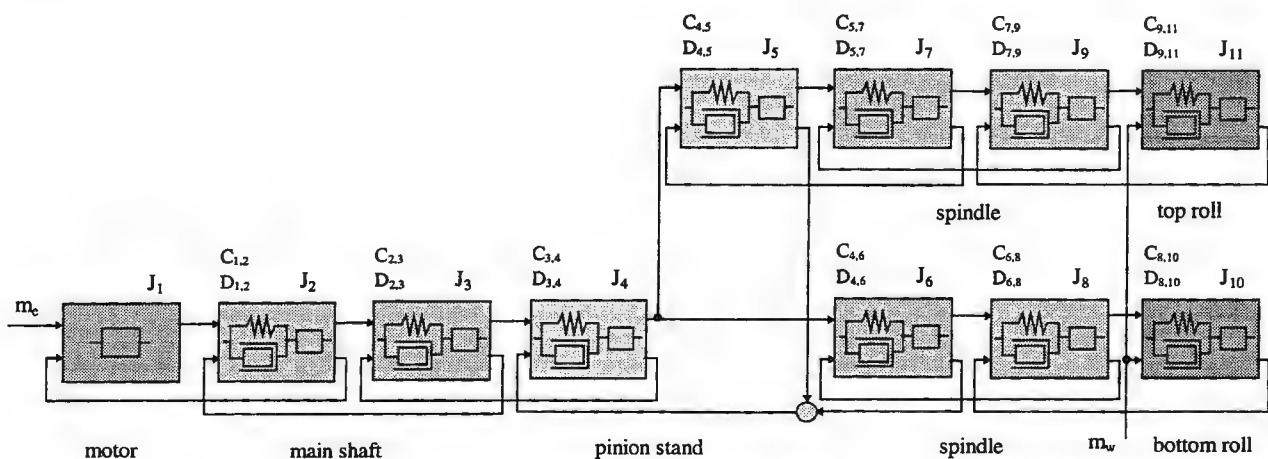
Figure 9: Block diagram of a linear 11-mass system

248
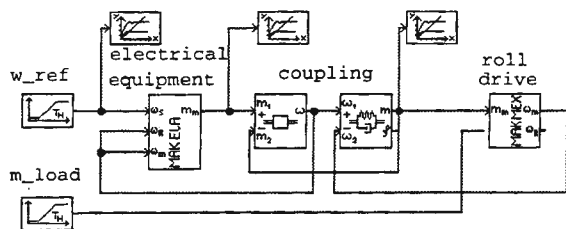
Figure 10: Overall model of the drive system with representation by macroblocks

The overall model of the electromechanical system in Figure 10 consists of the following components (viewed from left to right):

- The reference function of the drive and a load function of the mechanical equipment
- The macroblock of the electrical system portion (consisting of control, power inverter and electrical machine).
- The macroblock of the mechanical equipment of the drive (from the motor rotor via the pinion gearing up to the rolls).

The oscillograms in Figure 11 and Figure 12 show some of the simulation results.
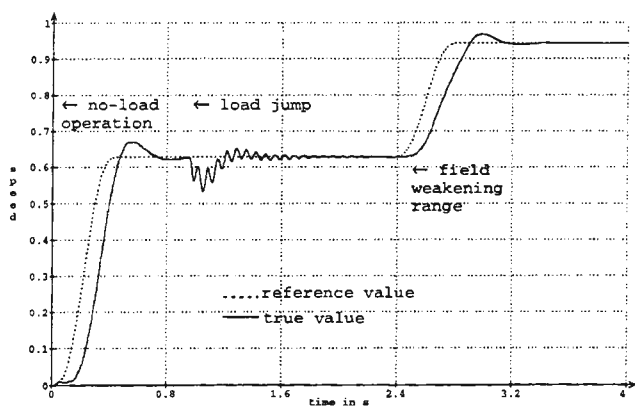
Figure 11: Reference and true value of speed
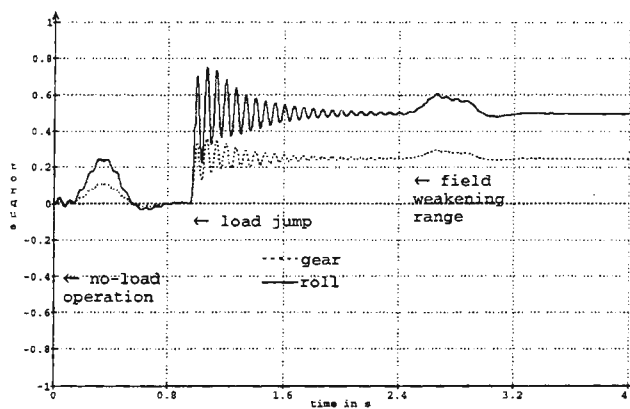
Figure 12: Torsional torque at gear and at roll

Summary

A solution for the optimization of drive systems requires a comprehensive approach and hence equal representation of the electrical and mechanical system parts. As a common means of description, block diagrams and circuit diagrams as per DIN 19226 are applied. Only in this way can the interacting energy flow be modeled to meet the requirements of simulation. Thanks to simulation program *DS88* it was possible to develop a model library which is adapted to the specific task and which can be extended.

References

[1]   Müller, V.: Handbuch zum Simulationssystem *DS88*
        TU Dresden, Elektrotechnisches Institut, 1997
        Demoversion: ftp://eeiwzb.et.tu-dresden.de/pub/aa/ds88
[2]   Bühler, H.: Einführung in die Theorie geregelter Drehstromantriebe,
        Band 1, Basel und Stuttgart: Birkhäuser Verlag, 1977
[3]   Schönfeld, R.: Elektrische Antriebe Bewegungsanalyse, Drehmomentsteuerung,
        Bewegungssteuerung. Springer Verlag, Berlin, Heidelberg, New York, 1995
[4]   Braun, M.: Ein dreiphasiger Direktumrichter mit Pulsbreitenmodulation zur getrennten Steuerung der Ausgangsspannung
        und der Eingangsblindleistung. Dissertation TH Darmstadt, 1983
[5]   Laschet, A.: Simulation von Antriebssystemen: Modellbildung der Schwingungs-
        systeme und Beispiele aus der Antriebstechnik. Berlin, Heidelberg, New York: Springer-Verlag, 1988
[6]   Cerv, H.: Softwarekonzept zur Analyse und echtzeitfähigen Simulation des dynamischen Verhaltens elektromechanisch
        gekoppelter Antriebe. Verlag der Augustinus-Buchhandlung, Aachen, 1992

COMPARISON OF SOME POPULAR CONTINUOUS SYSTEMS MODELLING AND SIMULATION TOOLS FOR HYBRID SYSTEMS

G. Kržič, B. Zupančič, G. Mušič

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, KrzicG@n-lb.si

Abstract

The paper deals with original capabilities of tools for hybrid modelling and simulation. A description of hybrid phenomena according to recognised classification by Branicky is given along with the expected properties of the tools for hybrid modelling and simulation. Some characteristic models of hybrid phenomena in two popular tools DYMOLA and MATLAB-SIMULINK, using built-in capabilities, show the advantages and disadvantages as well as current trends and needs.

Introduction

Hybrid systems or combined continuous/discrete systems have been the subject of research in automatic control and computer science for over a decade. The first useful models appeared in the computer science because of the event-driven nature of the state changes represented as different discrete states of the system. Computer scientists are very familiar with discrete event systems, so they gathered their efforts to incorporate continuous variables into a discrete model normally based on the model class of finite automaton. Complete model was suitable for simulation and verification, but not for control. However models for automatic control and other technical sciences were used with aim to ease the control of such systems. Solutions were sought in separate modelling of continuous process and discrete event system coupled with specially defined interface. Petri net was additional model class for discrete part found in those models. Immaturity and lack of common agreement in hybrid systems theory have prevented the theoretical solutions to reflect themselves in the accompanying development of suitable modelling and simulation tools. The majority of tools, still small in number, was developed due to the need of a suitable tool supporting the concrete applications of hybrid systems and has not overgrown into a market product. DYMOLA and MATLAB-SIMULINK are two very attractive tools available on the market, which in the past few years have added a number of discrete elements to the conventional continuous modelling and simulation environment to support characteristic hybrid phenomena.

Characteristic hybrid phenomena and required capabilities of tools

According to Branicky [1], discrete phenomena found in hybrid systems are classified into *switchings* and *jumps*. If the systems dynamics is described in state-space form $\dot{x}(t) = f(x(t), u(t))$, than the former represent discontinuous changes of the derivative function f while the latter discontinuous changes of the state x when the state x reaches certain boundaries of the state space. The first case expresses different *physical modes* of a system or *configurations* while the second represents *state resetting*. An example of the switching can be found in [2]. As seen in Figure 1 a tank has a form which introduces a change in dynamics at a certain level h_0 so that the complete dynamics depends on the level h as described by
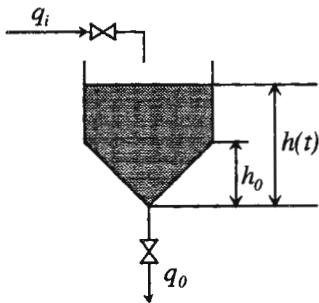
Figure 1. An example of switching as a result of a change in geometry

$$A\dot{h}(t) = q_i - q_o, \quad h > h_0 \qquad A\left(\frac{h(t)}{h_0}\right)^2 \dot{h}(t) = q_i - q_o, \quad h \le h_0 \tag{1}$$

Another example of switching would be a *hysteresis effect* (e.g. ON-OFF control). A classical example of the *jump* is the modelling and simulation of elastic impact. Consider the simplified dynamics of a free-falling ball (h ... height, v ... velocity, g ... gravity acceleration)

$$\dot{h} = v \qquad\qquad \dot{v} = -g \tag{2}$$

There is a jump at the moment of impact, i.e. on the boundary $h = 0$, when the velocity is instantly set from v to $-cv$, where $0<c<1$ is a coefficient of restitution.

Implementations of such discontinuities rely heavily on logical decision-making, memory store and some definition of event. Thus, a suitable tool should allow relations among quantities to be expressed in terms of logical expressions and equations. An event is normally a result of evaluations of those relations against logical constants true and false. In other words, the implicit definition of event is hidden within dynamical relations among variables. An event occurs, when the sign of a zero-crossing function changes in a corresponding direction.

If we consider events to be the driving force of hybrid systems, the following is still needed:
• ability to express discontinuities,
• some kind of storing and retrieving of the discrete-valued variables,
• instant equations to reset the continuous and discrete variables,
• information about possible events must be transferred to the integration algorithm in order to handle discontinuities properly and efficiently. By the aid of zero-crossing function the occurrence of state events and the instants of the events is determined by root-finding. After that the integration step is adjusted to a value which ensures, that the current step interval is terminated exactly (within the computer precision) at the instant of the event (discontinuity). Than the integration is restarted.

Hybrid modelling with DYMOLA and MATLAB-SIMULINK

Let us take a look at the example of switching in the before-mentioned tank. If statement is the easiest way to describe different configurations in DYMOLA:

```
der(h) = if h > h0 then 1/A*(qi - qo) else 1/A*sqrt(h0/h)*(qi - qo)
```

The derivative function, i.e. the right side of the equation, dynamically changes according to the level h.

To implement a switching within the block diagram of MATLAB-SIMULINK we simply use **switches**. Figure 2 shows a two-position switch with the threshold h_0 whose output is fed to the integrator.
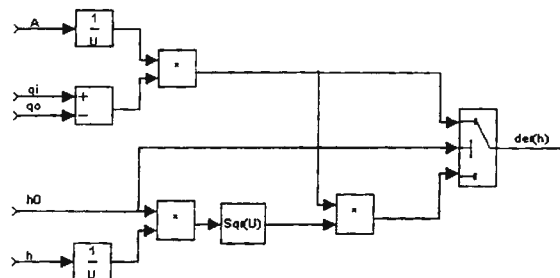
Figure 2. Implementation of switching using two-position switch

It must be admitted that the textual description in DYMOLA is superior in this case.

An example of hysteresis effect which acts as a controller in a thermostat-furnace system is taken from [3]. The thermostat control signal r switches the furnace on when the temperature T exceeds a maximum temperature T_{max} and switches it off when the temperature falls under a temperature T_{min}.

A discrete variable in DYMOLA representing control signal r, e.g. 0 or 1, can easily be changed using **new()** operator. It is used within the body of **when** statement defining the event:

```
when (T > Tmax) or (T < Tmin) then
    new(r) = if (T > Tmax) then 0 else 1
endwhen
```

MATLAB-SIMULINK offers at least two implementations. The simplest one is to use relay (hysteresis) block. Threshold values are set to T_{max} and T_{min}, and the output to 0 and 1 respectively (Figure 3a). Another option introduces a kind of discrete variable concept by means of **memory blocks**: Data Store Memory, Data Store Read, and Data Store Write. In addition, a triggering mechanism is needed to change the discrete variable. A **triggered subsystem** suits the needs. The implementation is shown in Figure 3b (top-level model left, first subsystem right).
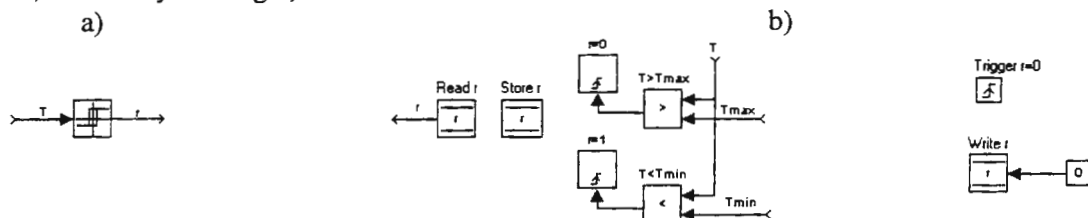
Figure 3 a) Using relay to implement hysteresis effect
b) Use of a discrete variable and triggered subsystem to implement hysteresis effect

Again the DYMOLA solution seems very convenient at least for people used to textual modelling. However, as hysteresis function is included in the SIMULINK block library it can be treated as a superior solution for this problem. The discrete variable concept is too complicated in this case but it can effectively be used in more complicated models.

The elastic impact of a ball with the surface serves as an example of a jump. Instant change of a continuous variable or state resetting in DYMOLA is again described within the when statement using **init()** operator:

```
when h < 0 then
    init(v) = -c*v
endwhen
```

State resetting is one of the properties of **Integrator** block in MATLAB-SIMULINK. Implementation in Figure 4 shows how the resetting of the state v is triggered when the state h changes from positive to negative. To avoid algebraic loop additional integrator state output must be used in place of normal output.
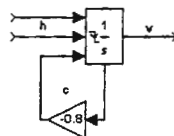
Figure 4. An example of a jump as implemented by triggered state resetting of an integrator

Hybrid automaton - a modelling example

Let us take a theoretical model of hybrid automaton [4] as shown in Figure 5. The system controls the reactor temperature by using two independent cooling rods. The aim is to maintain the temperature TH, which increases with the velocity of v_r, within the bounds TH_m and TH_M. When the temperature reaches the

upper bound of TH_M, the reactor must be cooled with one of the rods whose cooling velocity is v_1 and v_2 respectively. A rod can only be used if T time units have elapsed since its latest use. The two timers x_1 and x_2 serve that purpose. A complete shut-down of the system is required if there is no rod available.
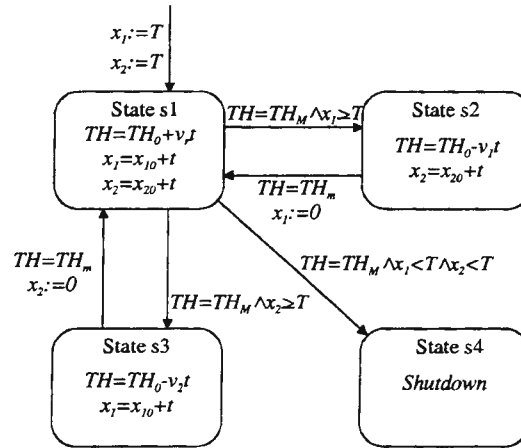
Figure 5. Hybrid automaton as a model of reactor

It is a typical example of a variable structure system. The dynamics depends on the discrete state or configuration of the system. The configuration change is a switching which can be accompanied by a jump. The transitions from state s1 to states s2, s3 and s4 correspond to switchings and transitions from states s2 and s3 to state s1 correspond to jumps. When both transitions from state s1 are possible, the transition to state s2 is preferred. The state s4 shutdown is considered to be of no dynamics.

In DYMOLA discrete variables s and p respectively are used to represent the current and the previous state. The dynamics is thus completely state-dependent. The discrete state check is a bit confusing due to the nature of relation expression in DYMOLA. The model is the following:

```
model Reactor
parameter THm=3 THM=15 T=6 vr=6 v1=4 v2=3
local TH=9 x1=T x2=T s=0 p=0

der(TH) = if (0 < s) and (s < 2) then vr ->
          else if (1 < s) and (s < 3) then -v1 ->
               else if (2 < s) and (s < 4) then -v2 else 0
der(x1) = if ((0 < s) and (s < 2)) or ((2 < s) and (s < 4)) then 1 else 0
der(x2) = if ((0 < s) and (s < 2)) or ((1 < s) and (s < 3)) then 1 else 0

when (TH < THm) or (TH > THM) then
    p=s
```

```
new(s) = if (0 < s) and (s < 2) and (TH > THM) and (x1 > T) then 2 ->
         else if (0 < s) and (s < 2) and (TH > THm) and (x2 > T) then 3 ->
         else if (1 < s) and (s < 3) and (TH < THm) then 1 ->
              else if (2 < s) and (s < 4) and (TH < THm) then 1 ->
              else 4

init(x1) = if (1 < p) and (p < 3) and (TH < THm) then 0 else x1
init(x2) = if (2 < p) and (p < 4) and (TH < THm) then 0 else x2
endwhen
end.
```

In MATLAB-SIMULINK a memory block is used to implement a discrete variable s, which indicates the corresponding state. Its value is changed within two triggered subsystems "From s1" and "To s1". The two timers are state variables implemented with two integrators. The top-level model, the two subsystems, "From s1" and "To s1", and one of the structure-like submodels of "From s1" are shown in Figure 6.

Conclusions

The built-in capabilities for hybrid modelling in DYMOLA seem to be a bit ahead of MATLAB-SIMULINK. On the current stage the direct textual model description is a step further from block diagram and gives the modeler a higher level of freedom. Manipulation with discrete variables is easier than storing signal values in memory blocks. The expected constructs like if and when statements and state-resetting operators in DYMOLA give an impression of hybrid oriented tool. However, some program structures are inflexible (e.g. in relation operators, no special structure for discrete (event) variables showing the continuous systems. Although the concept of MATLAB-SIMULINK is much closer to the engineering

practice following the classical simulation (block) scheme, the models of hybrid systems are hard to follow and sometimes reminds a user more on electronic circuits schemes, which is of course not acceptable or at least comfortable for the most engineers. With the improved blocks which are capable of zero-crossing detection and the very new memory blocks, the applicability for hybrid modelling increases.
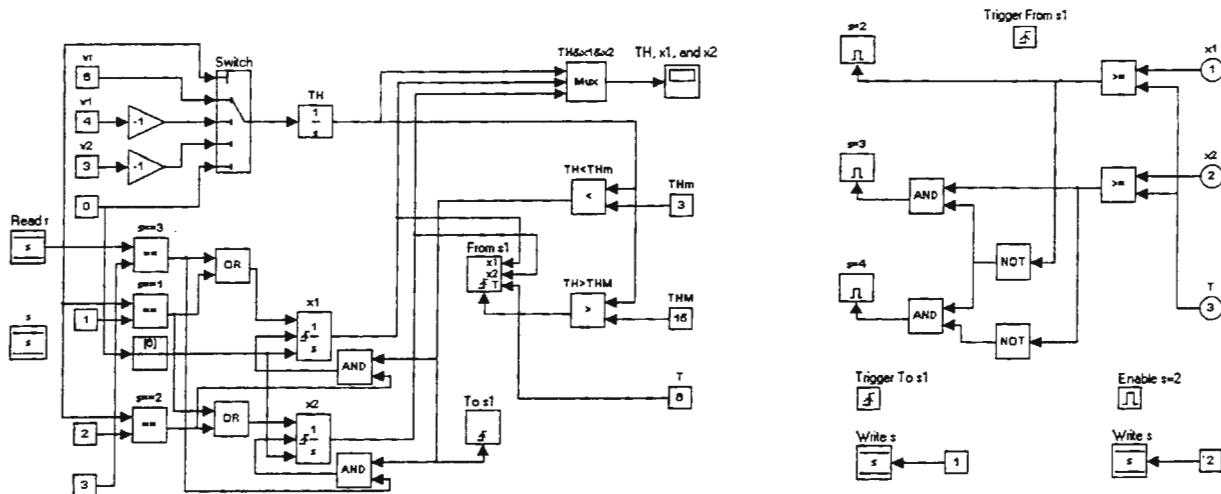
Figure 6. Hybrid control system of the reactor in MATLAB-SIMULINK

The combination of both tools in modelling and simulation process seems to be an adequate solution in particular applications. DYMOLA should serve as the starting tool for gathering the basic information about the behaviour of a hybrid system. The final application model and simulation should probably be done by using MATLAB-SIMULINK while possibly incorporating some of the already built DYMOLA submodels.

In our examples we did not face with any numerical problems. The discontinuities were handled properly and efficiently. By the aid of zero-crossing functions the occurrence of state events and the instants of the events (discontinuities) were determined accurately. The simulation results produced by both tools were the same. The accuracy was much better than in older MATLAB-SIMULINK versions which did not support state events.

The attention in this paper has been focused on the original, built-in capabilities of the mentioned tools. In spite of the fact that both tools are applicable for the modelling of hybrid systems, DYMOLA and MATLAB-SIMULINK models of shown examples (e.g. the hybrid control of chemical reactor) show that such modelling approach should not be the right direction of development for such tools, as models become confusing even for very simple problems. However, DYMOLA and MATLAB-SIMULINK are constantly developing tools and the very important add-ons, namely the Petri net library in DYMOLA and Stateflow in MATLAB-SIMULINK, could as well be mentioned. That additional functionality reveals the complete new perspective on the hybrid modelling and simulation and will probably contribute significantly to the popularity of both tools.

REFERENCES

[1] Branicky M., V. Borkar, S. Mitter: *A unified framework for Hybrid Control: background, model, and theory*, Technical Report LIDS-P-2218, LIDS MIT, 1994
[2] Lennartson B., M. Tittus, B. Egardt, S. Pettersson: Hybrid Systems in Process Control, *IEEE Control Systems Magazine*, 1996
[3] Stiver J. A., P. J. Antsaklis, M. D. Lemmon: *A Logical DES Approach to the Design of Hybrid Control Systems*, ISIS Technical Report 94-011, Notre Dame Univ., 1994
[4] Nicollin X., A. Olivero, J. Sifakis, S. Yovine: *An approach to the description and analysis of hybrid systems*, Hybrid Systems (Lecture Notes in Computer Science 736), Springer Verlag, Berlin, 1993
[5] Elmqvist H., F. E. Cellier, M. Otter: Object-oriented modelling of hybrid systems, *Proc. ESS'93, SCS European Simulation Symposium*, Delft, The Netherlands, pp.xxxi-xli.

Recipe-driven Batch Processes:
An Application of Hybrid Systems Modelling and Simulation

Martin Fritz and Sebastian Engell

Process Control Group, Dept. of Chemical Engineering, University of Dortmund, D-44221 Dortmund, Germany,
{martin|engell}@ast.chemietechnik.uni-dortmund.de

Introduction

In the process industries, batch processes are a practically relevant example of hybrid systems. Here, the continuous dynamics of the operations and the substances involved are of crucial importance for the overall behaviour of the system, as opposed to production processes in the manufacturing industries where discrete models are usually a sufficiently adequate representation of the processes. On the other hand, discrete sequential control is used to govern the stepwise execution of the process.

Modern batch process control systems follow the concept of recipe-driven production where desired production steps are expressed in a recipe [ISA95, NAM92]. A recipe is usually depicted in a sequential function chart that consists of steps and transitions, quite similar in syntax and semantics to condition/event Petri nets. Since the transition conditions in recipes depend on the lowest level on the values of continuous variables of the process, e. g. level or temperature of a vessel, continuous and discrete dynamics of the process are closely coupled to each other.

Apart from the discrete control, the processes may exhibit hybrid behaviour due to discontinuities in the plant. Examples are phase transitions (e. g. a substance that reaches its boiling point and starts to evaporate, thus decreasing the degrees of freedom of the system), a vessel running empty (thus changing the number of state variables, i. e. the substances to be balanced), or the creation of new substances by chemical reactions. These phenomena also lead to a high degree of structural variability.

In the planning and operation of batch plants, simulation is a frequently used tool. It can serve many purposes ranging from operator training to logistics analysis. Hence, the requirements on batch process modelling and simulation tools are quite diverse. The required accuracy of simulation models differs considerably which in turn affects the simulation methodology. In practice, both discrete-event as well as continuous simulation systems have been applied to batch process simulation. Most programs were originally developed for other application domains (e. g. process simulators for continuous chemical processes or discrete-event simulators for manufacturing applications) and have been adapted to batch processes.

Most of these simulation systems are realized stand-alone and do not offer interfaces for the exchange of models and results. Applying different simulators to the same process requires redundant modelling efforts and sophisticated output postprocessing to get comparable results. Under these circumstances, comparisons between different simulation approaches are hard to realize. A systematic approach to evaluate the suitability of the different approaches for the different purposes has not yet been undertaken.

BASIP

The BASiP (Batch Simulation Package) software package, developed at the Process Control Group at the University of Dortmund [FE97] tries to overcome this shortcoming by providing an open software architecture that enables the integration of different simulation approaches with a single user interface and a single model editor.

BASiP provides graphical editors to model the process using the common representation forms of the application domain (i. e. flow sheets of the plant and sequential function charts for recipes). Models created with the editors contain only generic, i. e. simulator-independent information, which consists of the structure

of the model and common parameters. Simulator-specific information as e. g. equations are added at the next level where the components of the generic model are mapped to model blocks from predefined, but extensible libraries. This structure enables the coupling to other simulators with the least possible effort of the user.

Two simulators, one based on the continuous and one on the discrete-event domain, have been implemented in BASIP. Furthermore, an interface to gPROMS exists that combines the ease of BASIP's graphical modelling with gPROMS' numerical capabilities.

For the visualization of the results, BASIP's concept of generic interfaces is used as well. Simulation output is produced in a neutral form which allows to build adapters to various tools for data visualization and analysis.

Hybrid system simulation

Most existing approaches to hybrid system simulation originate from the continuous domain. A solver for systems of ODEs or DAEs is enhanced by mechanisms for the handling of discontinuities in the system's state, expressed by time or state events. The correct detection of state events is a nontrivial task that can be of crucial importance for the accuracy and the performance of the simulator.

On the other hand, the notion of events is the core of any discrete-event simulator. If a discrete-event simulator is capable of determining the points of time when events occur and the system state at these instances, hybrid systems can be simulated accurately in the discrete-event domain.

The structural similarities of both simulation methodologies which become obvious in the context of hybrid systems are in sharp contrast to the traditional dichotomy between continuous and discrete-event simulation. The remainder of this contribution deals with the specific properties, common features, and limitations of both approaches as they are implemented in BASIP, and compares them by means of two examples.

The continuous approach

Starting in the continuous simulation domain, a natural approach to handle the discontinuous properties of a system is to partition the simulation time into intervals for which the system structure remains constant and the system input is continuous. The resulting system of differential and algebraic equations can then be solved using a standard numerical solver during that interval. The key problem of this approach is, of course, to find a suitable partition which corresponds to the problem of event detection. Event detection and handling is therefore central to the simulation algorithm that is shown in Fig. 1 [Wöl95].

In the first step after the initialisation, the state vector, consisting of the state variables and their derivatives, is reconfigured, i. e. the currently active state variables are determined and inserted in the vector. This step is repeated after each successful event detection. Note that in this reconfiguration step, the number of state variables in the state vector can change and even discontinuities in the values of state variables are allowed. By this dynamic
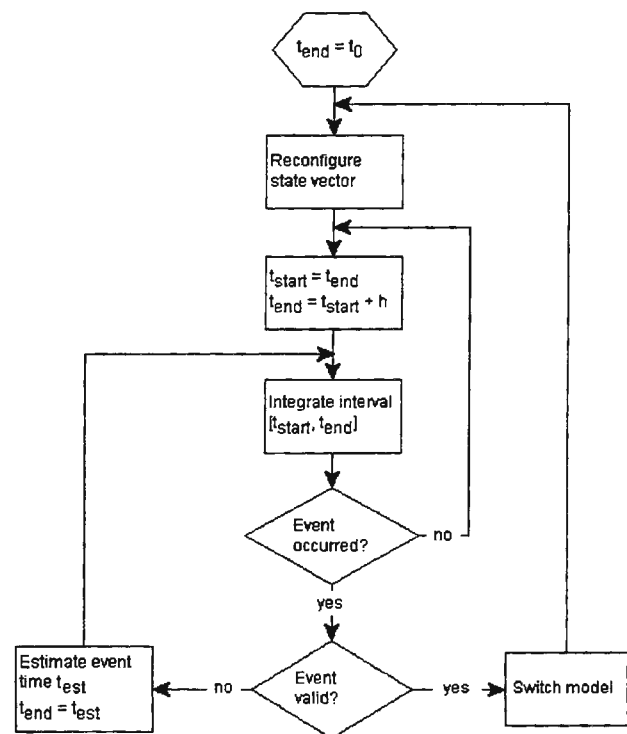
Fig. 1: Algorithm for hybrid simulation

256

reconfiguration, the size of the state vector is kept minimal, thus keeping the numerical integration efficient [WFS+96]. A state variable is considered to be active if – in the current configuration – there is at least one activity that contributes to a change in its derivative. There are, however, cases where a state variable is active although no operation of the recipe operates on it. An example are changes of the temperature of a (hot) vessel due to heat losses to the environment.

The integration routine in the next step evaluates the evolution of the system in an interval using a predefined maximum step size h. It may internally use a smaller, or variable step size, depending on the integration routine used. After each integration step, it has to be checked whether an event has occurred. In BASIP, state and time events are not distinguished, since a time event can be viewed as a special case of a state event. This leads to a more compact simulation algorithm without detectable performance degradation.

For event detection, the discrete part of the model maintains a set of state graphs, consisting of states which represent the set of equations valid in this state and transitions between these states. For each recipe and for each non-trivial piece of equipment, a separate state graph exists. Each transition is related to a threshold function that describes the condition which makes the transition valid. The thresholds of all transitions starting from active states are checked in the event detection step.

If an event has occurred, the validity of the event is checked, i. e. it is tested, whether the time of the event coincides with the end of the interval. Though it is not equivalent in all cases, it usually suffices to test if the value of the threshold that was crossed is still sufficiently close to zero. If this is the case, the event is valid and the model can be switched, i. e. the next state in the affected state graph is activated. Otherwise, an estimation of the time of the first event (more than one event might have occurred in an interval) that occurred is calculated by linear interpolation from past values (regula falsi), or, if that fails due to strong nonlinearities, by interval bisection. The end time is set to the estimated time and the system is integrated again over this interval. If the estimation was accurate enough, the event time coincides with the end of the interval and the event is declared valid. Otherwise, this procedure has to be iterated until the end point of the interval is close enough to the event time or the interval width is below a predefined limit. This algorithm is guaranteed to find each event in the correct order but may have slow convergence. See [Pla97] for an overview of other event detection algorithms.

The discrete-event approach

The main task of a discrete-event simulator is to determine the order of state changes which a system performs due to its internal dynamics and external inputs, starting from a given initial state. In timed models, the time for which the system remains in each state is also of interest. Thus, a discrete-event simulator only evaluates the system at the points of time where state changes occur.

Fig. 2 shows the simulation algorithm of a conventional discrete-event simulator. The core of the algorithm is the event list, a data structure containing all future events sorted by the time at which they are expected to occur. The simulation algorithm mainly consists of a loop that takes the first event in the event list, removes it from the list, sets the simulation time to the time of the event, and performs the action associated with this event. This action may involve the generation of new events that are inserted in the event list. Their associated time must not lie before the current simulation time. If the event list is empty or some other termination condition is fulfilled (e. g. a maximum simulation time is reached), the simulation is terminated.

It is noteworthy that the notion of event generation that usually constitutes the main part of a discrete-event simulator is semantically equivalent to event detection as explained above for continuous simulation. The difference lies rather in the point of view, whether events are something inherently contained in the system model and have to be "discovered", or whether they are created by the simulation program itself. Thus, both approaches have more in common than the traditional dichotomy between continuous and discrete-event simulation seems to imply. The term "event-based simulation" could be used as a generic term for both approaches.
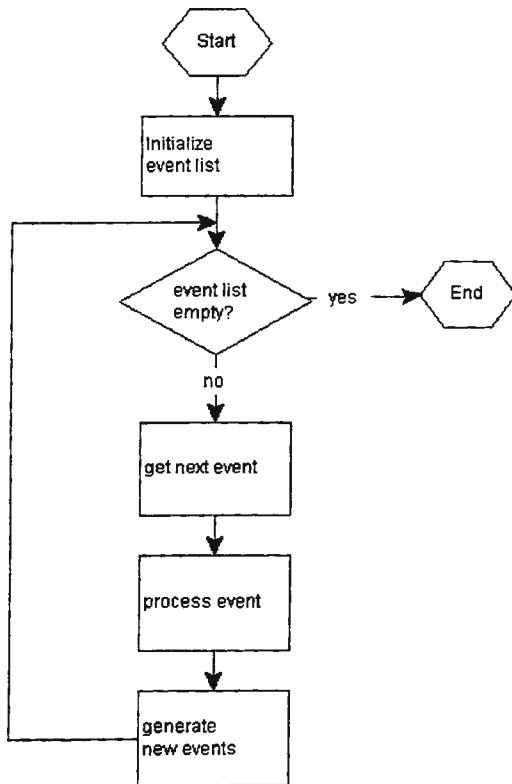
Fig. 2: Algorithm for discrete-event simulation

In the case of batch processes, the simulation is mainly driven by the execution of the recipes. Steps in the recipes correspond to states of the system. The events which the simulator has to generate in the first place are the switching times of the transitions of the recipe. There may be additional internal events to reflect structural changes in the plant, as explained below.

Thus, if a discrete-event simulator is able to compute the points of time when recipe transitions occur, combined with an adequate checking of limitations in the plant, no loss of accuracy compared to the continuous approach occurs.

However, the calculation of event times is not a trivial task. It requires that the changes of the state variables can be described by algebraic equations or at least by analytically solvable differential equations, e. g. simple integrators.

At the start of each recipe step (the activation of a *step event* in technical terms), its duration, i. e. the switching time of the following transition is calculated. A *transition event* is inserted into the event list at that time. If no time can be computed at this point, the event is temporarily deferred until the end of the simulation horizon. It may happen that the activation of another event (e. g. the start of another step of the recipe) enables that calculation. Thus, after each activation of an event, the times of all pending transition events are recalculated. This approach guarantees that no events are missed.

Another difficulty arises when equations describing a state variable change independently of (external) events arising from the execution of the recipe. An example is a substance that reaches its boiling point in a heated vessel. In the calculation of the duration of the heating step (or any operation that affects the energy balance) care has to be taken to examine this case. If it occurs, a so-called *internal event* is generated (in this example, at the time the boiling temperature is reached). After this event is activated, the evaporation of that substance has to be considered.

Example 1: Where discrete-event simulation suffices

For many applications, discrete-event and continuous simulation yield equivalent results if detailed modelling is not required. An example plant is shown in Fig. 3. The corresponding recipe is shown in Fig. 5. Here, vessel B1 is initially filled with 10 kg of substance A. In the next step, the content of the vessel is heated and substance B is filled in until a certain level is reached. Subsequently, B1 is emptied into B3 via buffer B2. It is assumed that all filling operations have constants flow rates, i. e. pumps are used. Furthermore, the heat transfer rate is assumed to be constant which is approximately true for electrical heatings. With this assumptions that lead to linear models, the results of both simulators are identical as shown in Fig. 4. The

Fig. 3: The example plant

differences arise only due to the lower number of data points generated by the discrete-event simulation since here output is only generated at event times. The calculations, however, are correct for each point of time at which the system is evaluated.

Whether the assumptions and simplifications taken are justifiable, depends on the goal of the simulation. For bird's eye views of a process, the performance advantage of discrete-event simulation may outweigh the possible loss of accuracy. A good example is the use of simulation in a production planning system for batch processes which has been described in [SFL+97].
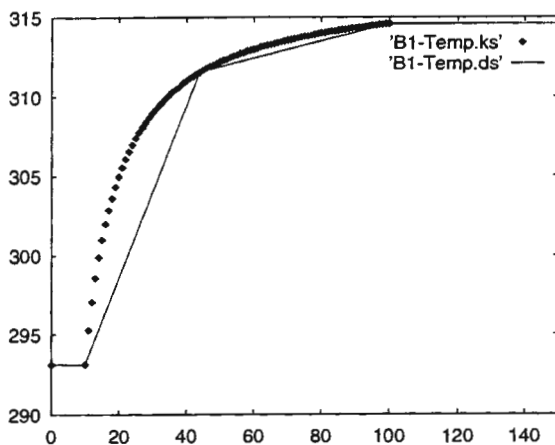
Fig. 4: Temperature in vessel B1
(dotted: continuous, solid: discrete-event)
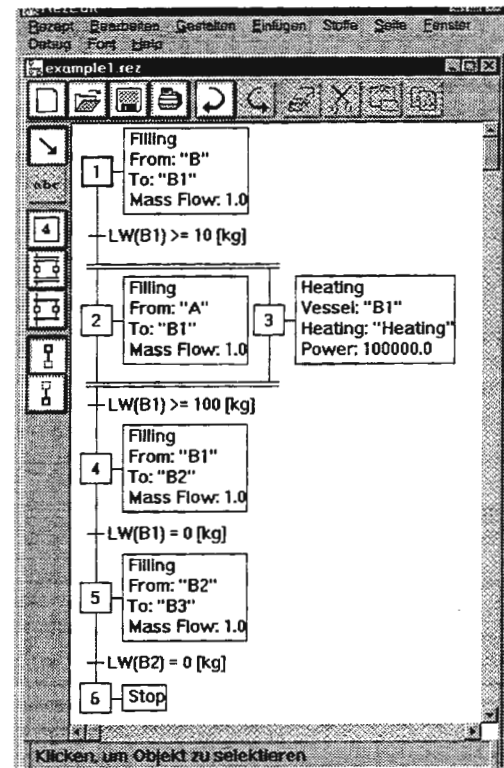
Fig. 5: Recipe for example 1

Example 2: Where discrete-event simulation fails

In some cases, however, discrete-event simulation does not only lead to less accurate results but can even show qualitative wrong behaviour. Such situations are likely to occur when multiple operations operate on the same equipment simultaneously. To demonstrate this, another recipe is regarded for the example plant (Fig. 7). For simplicity, the heating operation is omitted, instead, a recycle stream from B2 back to B1 is introduced. B1 is continuously fed with substance A, while a part of the contents of B2 are discharged into B3. The rates are chosen such that the level in B1 slightly increases and in B2 decreases. The simulation shall continue until the concentration of substance A in vessel B1 falls below a certain value.

While the hybrid simulator computes the correct solution without difficulties (cf. Fig. 6), the discrete-event simulator fails to detect the last switching condition. This is due to the fact that the cycle induces an algebraic loop in the discrete-event simulation model that the simulator is not able to solve.

Conclusions

The hybrid nature of batch processes demands a hybrid simulation approach. However, no single simulation approach is suitable for all possible applications of batch process simulation. Both the continuous and the discrete-event domain provide a sound foundation for a simulator for hybrid systems. Each approach has its specific advantages and limitations. While continuous simulation can give accurate results even for detailed models, discrete-events simulation is the method of choice when simulation speed is crucial and the process is viewed from a rather abstract level.

The architecture of the BASIP framework presented in this contribution allows to integrate different simulation approaches within a single modelling environment thus reducing the modelling efforts considerably and enabling comparisons between the different approaches.

Further work will include an enhancement of the customisation possibilities of model editors and simulators to further increase the flexibility of the software package.
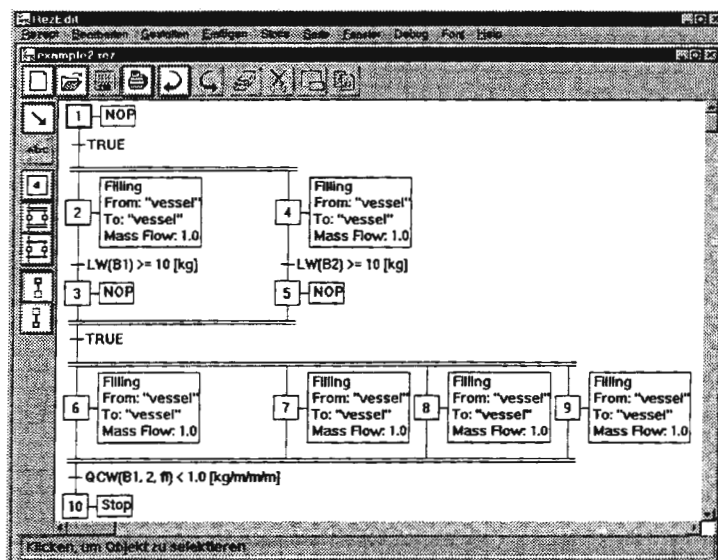
Fig. 7: Recipe for example 2
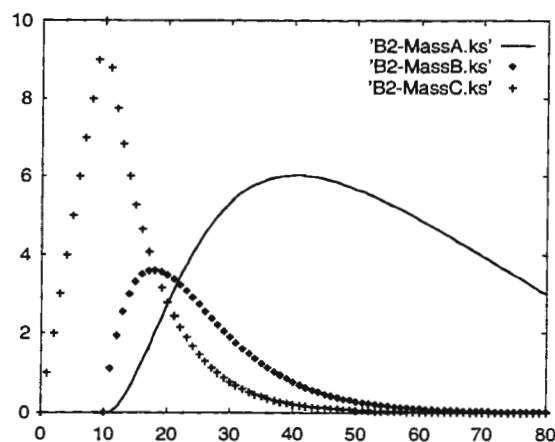
Fig. 6: Masses of substances A, B, and C for continuous simulation

Acknowledgement

References

[FE97] M. FRITZ and S. ENGELL. An Open Software Architecture for Batch Process Simulation. *Proc. PSE '97/ESCAPE-7*, Trondheim, Norway, Supplement to Comp. Chem. Eng. 21, 769-773, 1997.

[ISA95] ISA. *ISA S88.01 Batch Control, Part 1: Models and Terminology*, 1995.

[NAM92] NAMUR AK 2.3 Funktionen der Betriebs- und Produktionsleitebene. *NAMUR-Empfehlung 33: Anforderungen an Systeme zur Rezeptfahrweise (Requirements to be met by systems for recipe based operation)*, 1992.

[PB95] C. C. PANTELIDES and H. I. BRITT. Multipurpose process modelling environments. *Proc. Conf. On Foundations of Computer-Aided Process Design '94*, 128-141, CACHE Publications, 1995.

[Pla97] J. PLANK. *State Events in Continuous Modelling and Simulation Concepts, Implementations and New Methodology*, Ph. D. thesis, TU Vienna, 1997.

[SFL+97] M. STOBBE, M. FRITZ, T. LÖHL, and S. ENGELL. BASIS: Simulationsbasierte Produktionsplanung für rezepturgesteuerte Mehrproduktanlagen. *Proc. 11. Symposium Simulationstechnik ASIM '97*, Dortmund, 290-295, 1997.

[Wöl95] K. WÖLLHAF. *Objektorientierte Modellierung und Simulation verfahrenstechnischer Mehrproduktanlagen (Object-oriented modelling and simulation of chemical multipurpose batch plants)*. Dr.-Ing. Thesis (in German), University of Dortmund, Dept. of Chemical Engineering, 1995.

[WFS+96] K. WÖLLHAF, M. FRITZ, C. SCHULZ, and S. ENGELL. BASIP - Batch Process Simulation With Dynamically Reconfigured Process Dynamics. *Proc. ESCAPE-6*, Rhodes, Greece, Supplement to Comp. Chem. Eng., 20(972), 1281-1286, 1996.

HYBRID SIMULATION OF CONTROL ENGINEERING SYSTEMS AND ITS APPLICATIONS IN LARGE COMPLEX ENVIRONMENTAL SYSTEMS

F. Stănciulescu

Research Institute for Informatics
Averescu Avenue 8-10, 71316 Bucharest, Romania
e-mail: sflorin@u3.ici.ro

Introduction

The complexity of natural systems and of those developed by humans will require hybrid models, rather than unitary models, to describe, simulate and control such systems [1,2,4-6,8]. Complex system presents multiple facets: some of the processes involved are continuous, some others are discrete-time, or discrete event processes. Complex systems, or parts of them may not be well-structured [3]; a heuristic knowledge-based model better suits the simulation and/or control of complex systems.

This paper extends the concept of hybrid model including a fuzzy knowledge-based control model. The hybrid model of complex system is a model composed of interconnected (sub)models, of different kinds: a continuous simulation model and/or a discrete-time simulation model, a discrete event simulation model, and a heuristic knowledge-based and/or a fuzzy knowledge-based model [7]. A PC hybrid model can serve both simulation and control of complex systems, as for instance complex environmental systems.

Hybrid modelling of large complex systems

The continuous processes simulation model

The continuous time simulation model of a complex system is a set of nonlinear differential equations:

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + f_i(x_i, \alpha_i) + v_i(x),$$ (1)

$$x_i(0) = x_{i0},$$ (2)

$$v_i(x) = \sum_{j=1, j \neq i}^{n} g_{ij}(x_j),$$ (3)

$$x_{i\,min} \leq x_i \leq x_{i\,max}, \quad (i = 1, 2, \ldots, n)$$ (4)

where: $x_i, v_i : R^{n_i} \rightarrow R$, $u_i : R^{m_i} \rightarrow R$ and where: A_i, B_i are matrices of state and control, f_i-a vectorial function describing nonlinearities of the process. The double inequality (4) interprets the expert's desire that the state x_i should belong to the suboptimality interval $[x_{i\,min}, x_{i\,max}]$.

The discrete time simulation model of a complex system is a set of nonlinear discrete-time equations, of the standard form, resulted from the discretization of a continuous simulation model:

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + f_i(x_i(k), \alpha_i) + v_i(x(k))$$ (5)

also including the relations (2)-(4); k takes values from the discrete-time values set, $\{k_0, k_1, \ldots, k_i, \ldots, k_f\}$.

The discrete event processes simulation model

A discrete event simulation model will be included, of which standard form is a Boolean differential equations set. The model of the discrete event system evolution is:

$$X_i(k+1) = F_i(X_1(k), X_2(k)\ldots, X_n(k), U_1(k), U_2(k), \ldots U_m(k))$$ (6)

$$Y_j(k+1) = G_j(X_1(k), X_2(k)\ldots, X_n(k), U_1(k), U_2(k), \ldots U_m(k)),$$ (7)

where X_i and Y_j take values from the Boolean set $\{0,1\}$, F_i is the discrete event state function, and G_j, is the discrete event output function. Both F_i and G_i are Boolean functions. Let $x_i(t)$, $(i = 1,2,\ldots,n)$ be the simulated system states. So, we can define the discrete event variables:

$$X_j(k) = \begin{cases} 1, \text{ if } x_j(t) \in \left[x_{i\,min}, x_{i\,max}\right] \\ 0, \text{ other} \end{cases} \tag{8}$$
$(i=1,2,...,m)$

$$U_j(k) = \begin{cases} 1, \text{ if } u_j(t) \rightarrow x_j(t+1) \in \left[x_{i\,min}, x_{i\,max}\right] \\ 0, \text{ other} \end{cases} \tag{9}$$
$(j=1,2,...,n)$

where the interval $[x_{imin}, x_{imax}]$ is the so-called suboptimality interval of $x_i(t)$ established by the expert.

The fuzzy knowledge-based control model

Control problem is to maintain the state variables $x_i(k)$, $i = 1, 2, ..., n$; $k = 0, 1, ..., k_f$ into the (sub)optimality intervals $[x_{imin}, x_{imax}]$. Control model is a fuzzy one. The state variables $x_i(k)$ belong to one or another of the intervals given in table 1, as well as $dx_i(k) = x_i(k) - x_i(k-1)$:

Table 1

Interval x_i	$< x_{imin} - \varepsilon$	$[x_{imin} - \varepsilon, x_{imin})$	$[x_{imin,}, x_{imax}]$	$(x_{imax}, x_{imax} + \delta]$	$> x_{imix} + \delta$
x_i	VS	S	O	L	VL
Interval dx_i	$< -\alpha$	$[-\alpha, 0)$	0	$(0, \beta]$	$> \beta$
dx_i	-L	-S	Z	S	L

where: VS = Very Small, S = Small, O = (sub) Optimal, L = Large, VL = Very Large, Z = Zero, are qualitative values (in the sense of fuzzy set theory). The state variables x_i belong to qualitative set: $\chi = \{ VS, S, O, L, VL \}$ while the derivatives dx_i belong to qualitative set: $d\chi = \{-L, -S, Z, S, L\}$. The computation of membership functions $\mu_i(x_i)$ is made by a trapezoidal representation (Figure 1).
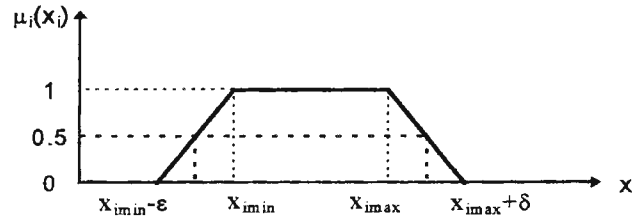
Figure 1 The trapezoidal representation of the membership function

Figure 1 let us infer that:

$$\mu_i(x_i) = \begin{cases} 1, \text{ if } x_i \in \left[x_{i\,min}, x_{i\,max}\right], \\ 0, \text{ if } x_i < x_{i\,min} - \varepsilon, \text{ or } x_i > x_{i\,max} + \delta, \\ \dfrac{1}{\varepsilon}x_i - \dfrac{x_{i\,min} - \varepsilon}{\varepsilon}, \text{ if } x_i \in \left[x_{i\,min} - \varepsilon, x_{i\,min}\right) \\ \dfrac{-1}{\delta}x_i + \dfrac{x_{i\,max} + \delta}{\delta}, \text{ if } x_i \in \left(x_{i\,max}, x_{i\,max} + \delta\right] \end{cases} \tag{10}$$

The computation formula of the new control variables, $u_i(k+1)$, able to reset the state variables $x_i(k)$ into the (sub)optimality intervals is:

$$u_i(k+1) = u_i(k) + K_{ui}\Delta u_i(k) \tag{11}$$

where $\Delta u_i(k)$ is an increment, computed by the formula

$$\Delta u_i(k) = \dfrac{\sum\limits_{i=1}^{n} \mu_i(x_i) \cdot u_i(k)}{\sum\limits_{i=1}^{n} \mu_i(x_i)}, (i=1, 2, ...n) \tag{12}$$

The set of fuzzy knowledge-based rules, useful in control of the complex controlled system result:

$< If (x_i(k)$ is χ_i) $\wedge$ $(dx_i(k)$ is $d\chi_i$)

Then the new control law is: $<u_i(k+1) = u_i(k) + K_{ui} \Delta u_i(k) >$, $(i = 1,2,...,n)$ (13)

We emphasise that the number of fuzzy control rules is: 25 n.
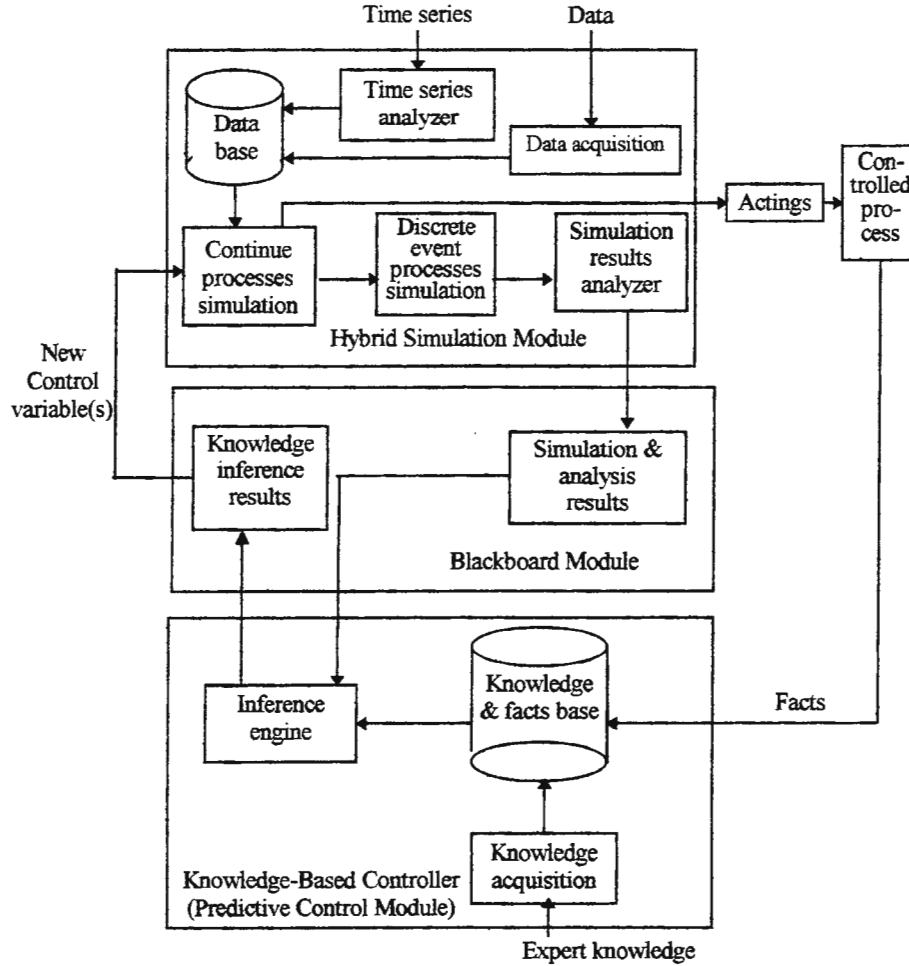
Figure 2 Block -diagram of hybrid simulation&control system

Hybrid simulation&control of large complex systems

The hybrid simulation&control of large complex systems, depicted in Figure 2, is based on the following algorithm:

Step 1: *Solve* the numerical simulation problem, described by the nonlinear discrete-time model (5), (2)-(4) and let $x_i(k)$ be the values obtained by simulation $(i=1,2,..,n; j=1,2,..,m; k \in [k_0,k_f])$;

Step 2: At each time moment k *verify*, using the discrete event model (6)-(9), whether the state variable x_i $(i=1,2, ...,n)$, obtained during Step 1, belongs to the interval $[x_{imin}, x_{imax}]$ or not: if No, go to Step 3! if Yes, go to Step 5!

Step 3: *Search* for a new value of the control variables able to reset the state variables into the suboptimality intervals, by using the fuzzy control model (10)-(13);

Step 4: *Return* to *Step 1* and restart the system simulation;

Step 5: *Examine* the evolution of the simulated state variables x_i obtained at *Step 1* which belong to the (sub)optimality intervals (4), but are close to the limits:

if $(x_i(k+1) < x_i(k)) \wedge (|x_i(k+1) - x_{imin}| < \varepsilon_1) \vee (x_i(k+1) > x_i(k)) \wedge (|x_{imax} - x_i(k+1)| < \varepsilon_2)$, go to *Step 3* !

if No, go to *Step 6*!

Step 6: *Verify If* $k < k_f$:
- if Yes, return to *Step 1*!
- if No, go to *Step 7*!

Step 7: *Print, or Plot* the states $x_i(k)$ $(i=1,2,...,n)$, $k \in [k_0,k_f]$.

Step 8: *Stop*!

The library of hybrid models for simulation and control in ecology and environment protection

Hybrid simulation and control models have applications in simulation and control of complex systems, for example: environmental systems (natural, ecological systems), technical, industrial and socio-economic systems. To accomplish this task a Hybrid Models' Library (HML) was created. The HML has a hierarchical structure (see Figure 3).
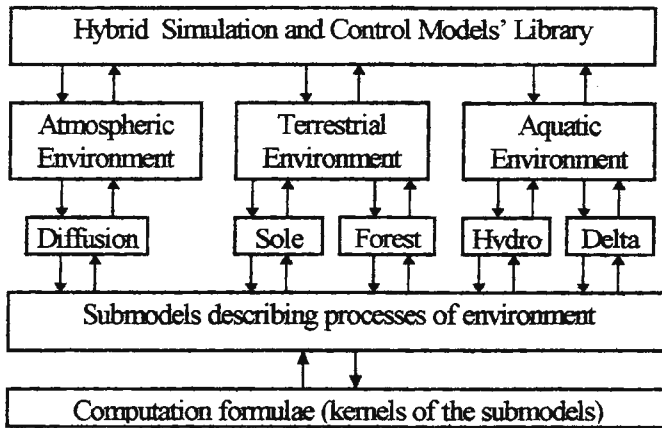
Figure 3 The structure of the hybrid models' library
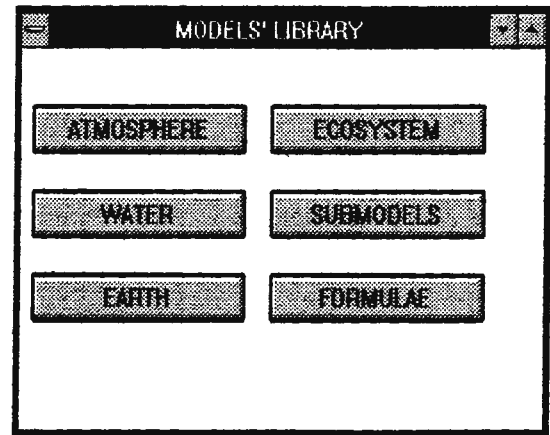
Figure 4 Library's push buttons

HML is structured on four levels: *Level 1*: a hybrid model of the environment as a whole; *Level 2*: specialised hybrid models; *Level 3*: hybrid submodels, able to describe biological, physical, chemical subprocesses; *Level 4*: computation formulae, which are the kernels of the submodels. We emphasise that one model selection of the HML is done automatic, thanks to a program written in Visual C++ language (clicking on the corresponding button of Figure 4). The HML can also be used to build new models, by using levels 3 and 4, submodels and computation formulae; the user himself can add new elements (submodels & kernels). All the models included in the HML have been tested and validated in real applications, as for instance applications concerning simulation and control of bio-hydro-chemical processes of the Danube Delta, agro-and forest-ecosystems, and diffusion of chemical pollutants in the urban area.

Control engineering applications of environmental systems

Both the theoretical approach and the models' library previously discussed have applications in simulation and control of complex environmental systems. We will describe two experiments: simulation and control of an aquatic ecosystem (the case of the Danube Delta ecosystem) and simulation and control of urban atmosphere pollution due to industrial chemical pollutants (the case of sulphur dioxide-SO_2).

Hybrid simulation of the control engineering system of the Danube Delta aquatic ecosystem

The application refers to a complex ecological system of the large scale hydrological system of the Danube Delta, composed of a network of large lakes, interconnected by canals and channels. The problem is to simulate the system behaviour, and control the evolution of some state variables. The variables of the simulation model are the input set: precipitations (p), the water inflow through canals and channels (Q_{ij}); the endogenous parameters: the surface of the lakes (A_j); the states space: the water level in the lakes (H_j), the depth of water in lakes (HH_j), the volume of water (V_j), the water refreshment factor (W_j); the concentration of nutrients from the water (c_s), and the biomass of species which live in water (B_h, B_i, and B_j); the output set: the evaporation of water (e), the water outflow through canals and channels (Q_{je}) The objectives of the system: to maintain W_j, c_s, B_h, B_i,and B_j in the suboptimality intervals, to improve W_j, using control variables, i.e. geometrical dimensions of canals/channels (a, l, L), roughness (ρ), the degree of covering with vegetation (f), construction of a new canal/channel, fishing, hunting, populating with new species etc.

The mathematical-heuristic model for simulation and control of the hydrological processes describes the circulation of water through the network of canals and channels, its inflow, its outflow, the level, depth, volume of water, and the water refreshment factor.

The numerical simulation model of hydrological processes composed of state equations (4 state equations for each lake for H_j, V_j, and W_j);

$H_j(k+1)=H_j(k)+p(k)-e(k)+V_jQ_{ij}(k)-B_jQ_e(k)+B_ju_j(k)$,

$HH_j(k+1)=H_j(k)+H_{jmed}$

$V_j(k)=A_jH_j(k)$,

$W_j(k)=V_{ij}(k)/(V_j(k)+V_{ij}(k))$,　　$(j=1, 2, ..., n$; in our case $n=10)$, where:

$Q_{ij}(k)=\rho_{ij}A_{ij}(2g)^{1/2}\,sign\,x_{ij}\,(|x_{ij}|)^{1/2}$,　$x_{ij}=z_i+V_i/A_i-(z_j+V_j/A_j)$,

$Q_{je}(k)=\rho_{je}A_{je}(2g)^{1/2}\,sign\,x_{je}\,(|x_{je}|)^{1/2}$,　$x_{je}=z_j+V_j/A_j-(z_e+V_e/A_e)$,

$A_{ij}=L_{ij}H_{ij}$,　$H_{ij}=1/2(V_i/A_i+V_j/A_j)$,

$A_{je}=L_{je}H_{je}$,　$H_{je}=1/2(V_j/A_j+V_e/A_e)$,　$V_{ij}=Q_{ij}\Delta t$,　$B_j=1/A_j$.

Initial conditions are set. We specify that L_{ij}, H_{ij} represent the width and respectively the depth of water in the canal which links the lakes i and j, V_{ij} is the volume of water entering the lake j, by canal ij. We also put some constraints on the model above, i.e. the suboptimality intervals (given by the expert):

$W_j(k) \in [0.1, 0.5], j=1, 2, ..., 10$

The discrete event simulation model of the discrete event hydrological processes is based on the matrix W. The discrete event states are defined as:

$$W(j,k)=\begin{cases}1, W(i,k) \in [0.1, 0.5] \\ 0, \text{ other}\end{cases}$$

The matrix of discrete event W results to be a Boolean matrix. The discrete event output variables being $Y_j(k)$, the discrete event model can be written as follows:

$Y_j(k+1)=W_1(k) \wedge W_2(k) \wedge ... \wedge W_j(k) ... \wedge W_n(k)$, where: $k=0, 1, 2, ..., k_f$;　$n=10$.

$$\text{If }\quad Yj(k+1)=\begin{cases}1, \text{ then the process continues} \\ 0, \text{ then alert the fuzzy controller}\end{cases}$$

The fuzzy knowledge-based control model of hydrological processes consists of heuristic rules from the knowledge base. The controlled state variables being $W_i(k)$ $i=1, 2, ., n$, the parameters in Table 1are: $x_{imin}=0.2$, $x_{imax}=0.5$, $\varepsilon=0.1$, $\delta=0.3$. The mathematical expression of the membership function $\mu_i(W_i(k))$ is:

$$\mu_i(W_i(k)) = \begin{cases}1, if\ W_i \in [0.2, 0.5], \\ 0, if\ W_i \in [0, 0.1)\ or\ W_i \in (0.8, 1], \\ 10W_i-1, if\ W_i \in [0.1, 0.2) \\ -\dfrac{10}{3}W_i+\dfrac{8}{3}, if\ W_i \in (0.5, 0.8]\end{cases}$$

The new control variables $f_i(k)$ are computed as

$f_i(k+1) = f_i(k) + Kf_i\ \Delta f_i(k)$,

where:

$$\Delta f_i(k)=\sum_{i=1}^{6}\mu_i(W_i)\cdot f_i(k)\Big/\sum_{i=1}^{6}\mu_i(W_i)$$

The heuristic coefficients Kf_i are given by the expert. In order to illustrate we give only a rule:

$<$ *If* $(W_i(k) = S) \wedge (dW_i(k) = -L)$, *Then the law of fuzzy control is:* $f_i(k+1)=f_i(k)-0.35*\Delta f_i(k) >$

(and other 24 fuzzy control rules).

Figure 5 presents some simulation and control results of the water refreshment factor W of a big lake of the complex hydrological system of the Danube Delta. The lower curve represents the natural behaviour (W is

under the lower limit); the upper curve represents the controlled behaviour and shows that the water refreshment factor resets into the suboptimality interval, as a suite of new control/acting.
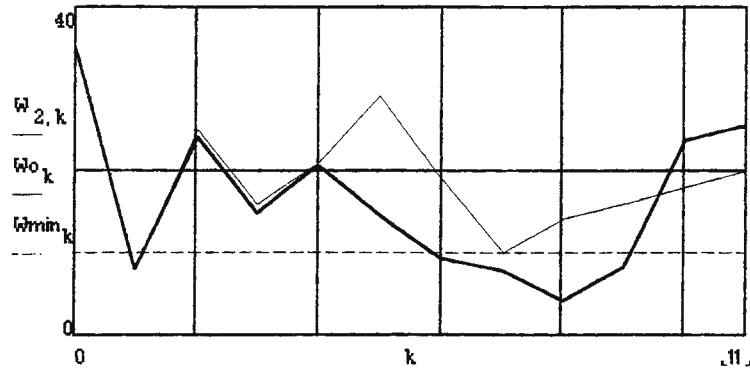
Figure 5 Simulation and control results of the water refreshment factor $W_{0,k}$,

Hybrid simulation of the control engineering system of the diffusion process of SO_2 in urban atmosphere

The problem is to simulate and control the diffusion process of the chemical industrial pollutant SO_2 in urban atmosphere, due to three emission sources. The simulation model comprises the Gaussian model of diffusion, and some heuristic computation formulae (e.g. for the mixture height, the roughness factor of the territory, the effect of some meteo-climate factors).

The numerical simulation model. The following symbols are used: H-emission source height[m], ΔH-mixture height[m], m-mass flow [g/s], D-bore of the emission source[m], w_0-gas emission speed [m/s], u-wind speed [m/s], T_1-temperature of the pollutant at the emission source[oC], T_2- air temperature[oC], I-solar light radiation [lux], P-air pressure[mm.col.Hg], U-air relative humidity [%], t-length of time of the emission process[s]. The numerical simulation model is:

$$c_{(i,j)} = \frac{S \cdot C(x(i),y(j),z,t)}{(2 \cdot \pi)^{1.5} \cdot \sigma 0(x(i)) \cdot \sigma 1(x(i)) \cdot \sigma 2(x(i))}$$ (gas concentration in $[mg/m^3]$), where:

$S = m.t.10^3$ - quantity of SO_2 during the emission process [mg]

$C(x,y,z,t) = g0(x,t) \cdot g1(x,y) \cdot g2(x,z) \cdot f1(I) . f2(P). f3(U)$

$$g0(x,t) = \exp\left[-0.5 \cdot \frac{(x - ut)^2}{(\sigma 0(x))^2}\right] \cdot \frac{1}{\lambda}$$

$$g1(x,y) = \exp\left[-0.5 \cdot \frac{y_2}{(\sigma 1(x))^2}\right] \cdot \frac{1}{\lambda}$$

$$g2(x,z) = \left[\exp\left[-0.5 \cdot \frac{(z - (H + \Delta H))^2}{(\sigma 2(x))^2}\right] + \exp\left[-0.5 \cdot \frac{(z + (H + \Delta H))^2}{(\sigma 2(x))^2}\right]\right] \cdot \frac{1}{\lambda}$$

$\sigma 0(x) = K \cdot (1 + \sigma \cdot u \cdot t)^\gamma$ - dispersion of the Gaussian cloud on an x-axis

$\sigma 1(x) = 0.32 \cdot x \cdot (1 + 0.0004 \cdot x)^{-0.5}$ - dispersion of the Gaussian cloud on an y-axis

$\sigma 2(x) = 0.24 \cdot x \cdot (1 + 0.001 \cdot x)^{-0.5} \Omega$ - dispersion of the Gaussian cloud on an z-axis

$$\Delta H = D \cdot \left(\frac{w_0}{u}\right)^{1.4} \cdot \left[1 + \frac{\Delta T}{(273 + T_2)}\right],$$ mixture height $(\Delta T = T_1 - T_2)$

The weighting formulae $f(I)$, $f(P)$ and $f(U)$, representing the quantitative influence of I, P and U on the diffusion process, are the following:

$f1(I) = 0.5(1 + exp(-\delta I))$, the weighting function due to light radiation intensity;

$$f2(P) = \exp\left(-K1 \cdot \frac{100}{P + \beta}\right),$$ the weighting function due to air pressure;

266

$$f3(U) = \exp\left(-K2 \cdot \frac{100}{U+\gamma}\right),$$ the weighting function due to air humidity.

Based on a numerical simulation model of the diffusion process of industrial chemical pollutants, simulation experiments have been made by using the Mathcad Plus 5.0 simulation system and a PC 486, for three emission sources of SO_2 (Figure 6).

Fig 6 Simulation of SO_2 diffusion   Fig 7 Simulation of SO_2 controlled diffusion

The discrete event model. We define the discrete event variable as:

$$CSO_2(k) = \begin{cases} 1, & if \quad CSO_2(k) \in [0, \ CSO_{2\,max}], \\ 0, & if \quad CSO_2 \rangle CSO_{2\,max} \end{cases}$$

A discrete event model will result:

$$YCSO_2(k) = CSO_{21}(k) \wedge CSO_{22} \wedge CSO_{23}$$

$$if \ \ YCSO_2(k) = \begin{cases} 1, \text{ then the simulation process continues} \\ 0, \text{ then alert the fuzzy controller} \end{cases}$$

The fuzzy knowledge-based control model. The control variables are: the height of the emission source stacks, the emission time, a.s.o. The mathematical expression of the relationships $\mu_i(CSO_2)$, for the four emission sources is:

$$\mu_i(CSO_2) = \begin{cases} 1, if \ CSO_2 \in [0, \ 0.005] \\ 0, if \ CSO_2 > 0.01 \\ -200 \ CSO_2 + 2, if \ CSO_2 \in (0.005, \ 0.01] \end{cases}$$

New control variables (i.e. the emission sources height H_i) will be computed as follows:

$$H_i(k+1) = H_i(k) + KH_i \cdot \Delta H_i(k),$$ where

$$\Delta H_i(k) = \frac{\sum\limits_{i=1}^{4} \mu_i(CSO_{2i})H_i(k)}{\sum\limits_{i=1}^{4} \mu_i(CSO_{2i})}$$

Figure 6 represents the map of the isopollution (isoconcentration) curves for SO_2; in some regions the concentration of SO_2 surpasses 0.006 mg/m³. By using the fuzzy control model we can evaluate new control variable/acting, able to reduce the SO_2 concentration by 50% (Figure 7).

Concluding remarks

Hybrid simulation is a complex simulation technique based on a hybrid (multimodel) approach. It consists of concurrent use of several kinds of models. The main (sub)models, of the hybrid model are: a set of nonlinear differential equations (including constraints on some state and control variables), which the simulation experiment is based on; a set of Boolean differential equations, describing the discrete events occurring in the (simulated) process; a fuzzy knowledge-based control model, able to synthesise a new control law, and finally new variables/acting. A hybrid algorithm for simulation and control of complex systems, using a hybrid mathematical-heuristic model is presented, able to synthesise the new control variables/acting, and used to reset the state variables into the suboptimality intervals when they exit from these intervals. Hybrid simulation and control models can be applied for simulation and control of complex systems, e.g. environmental systems (natural, ecological systems), technical, industrial and socio-economic systems. A *Hybrid Models' Library (HML)* for environmental problems was created, by using Mathcad Plus 5.0, including: the hybrid model of environment as a whole, specialised hybrid models for simulation and control of aquatic, terrestrial and atmosphere ecosystems; hybrid submodels, able to describe biological, physical, chemical subprocesses; computation formulae, constituting the kernels of the submodels. So, the HML can be used in simulation and control of some large complex aquatic environmental systems (e.g. a big lake, a hydrological network, a river delta a.s.o.), a terrestrial environmental system (a forest ecosystem, an agro-ecosystem), and an atmospheric system (the diffusion process of industrial chemical pollutants in the urban atmosphere), but also the dynamics of some protected populations of mammals and birds. We emphasise that selection of the HML models is entirely automatic, thanks to a program written in Visual C++ language (by a simple click on the corresponding button). The HML can also be used to build new models, by using the levels submodels and computation formulae; the user himself can add new elements (submodels & kernels). All the models included in the HML have been tested and validated in real applications, as for instance simulation and control of bio-hydro-chemical processes of an aquatic ecosystem (the Danube Delta), forest ecosystems and agro-ecosystems, and diffusion process of industrial chemical pollutants in the urban area. The inference of new knowledge from the knowledge existing in a knowledge base, is due to an expert system shell written in Clips 6.0. Simulation & control experiments have been made by using a PC 486, Some simulation results illustrate all the above and prove that they are useful in real applications, and that ecologists, hydrologists, sylviculturists, soil science experts and others can make their own decision on the simulation results, by using HML.

References

1. R. Grützner, Environmental Modelling and Simulation - Features of Experiments. In: *Proceedings of the EUROSIM'95 Conference,* Elsevier, Amsterdam, 891-896, 1995.

2. Y. Nakamori, Y. Sawaragi, Methodology and systems for environmental decision support. In: *Preprints of IFAC Symposium Large Scale System:Theory and Applications,* London, Vol. 1, 1-11, 1995.

3. D.D. Siljak, Complex dynamic systems: dimensionality, structure and uncertainty. In: *Large Scale Systems,* vol. 4, 278-294, 1983.

4. F. Stănciulescu, Simulation of complex ecological systems using a mathematical-heuristic model with application to Danube Delta system. In: *Proceeding of Schloss Dagstuhl Seminar and SAMS, Special Issue " Environmental Modelling and Simulation"* (G.R.Carmichael, G.A. Korn, A. Sydow, eds.), 281-302, 1996.

5. F. Stănciulescu, Hybrid Modelling of Complex Systems and its Applications in Knowledge-Based Simulation&Control. In: *Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics* (A. Sydow, ed.), W&T Verlag, Berlin, Vol. IV, 1- 6, 1997.

6. F. Stănciulescu, Simulation and control of environmental system using a mathematical-heuristic model and algorithm. An application to the Danube Delta system. In: *Environmental Software (USA),* Vol. 12, November, 1997.

7. L.A. Zadeh, Fuzzy Control: Issues, Contention and Perspectives. In: *13th IFAC Triennial World Congress,* San Francisco, USA, 35-38, 1996.

8. P. Zannetti, Environmental modeling - the next generation. In: *Development and Application of Computer Techniques to Environmental Studies VI. Proceedings of ENVIROSOFT 96* (P. Zannetti&C.A. Brebbia eds), Como-Italy, Computational Mechanics Publications, Southampton/Boston, 3-13, 1996.

SEMANTICS OF MODEL COMPOSITION IN HYBRID LANGUAGES

G. Fábián[1], D.A. van Beek[2], J.E. Rooda[3]

Eindhoven University of Technology, P.O.Box, 513, 5600 MB Eindhoven, {g.fabian[1], vanbeek[2], rooda[3]}@wtb.tue.nl

Abstract

The increasing complexity of hybrid models requires a mathematically sound language semantics that allows reasoning about model behaviour and supports development of robust models. Model composition in hybrid languages is realised by connecting continuous variables of sub-models. It is important to define the precise semantics of connections. Improperly defined connection semantics can *a)* introduce undesirable non-determinism and *b)* lead to undesirable semantics of conditional language constructs. In this article four different connection semantics are examined. Three of them impose strong restrictions on the modelling style or lead to less robust models. In the hybrid χ language the fourth possibility is chosen: a connection represents an algebraic equation. In combination with the initial state calculation mechanism, and the use of local variables, this results in robust and elegant composition semantics.

1. Introduction

Most hybrid languages support hierarchical models. Since many physical systems are composed of smaller components, models become more clear if they reflect the structure of the modelled system. Hierarchical modeling facilitates the process of model development in many ways. It supports both top-down and bottom-up development. Components can be tested individually and their behaviour can be studied in different environments, which promotes the development of libraries and standardisation in different application areas [1].

Model composition in hybrid languages is realised by connecting (among others) continuous variables of sub-models. For this purpose, connection elements, so-called *streams* (gPROMS [2], ASCEND [3,4]), *terminals* (Dymola [5,6], Omola [7]), or *channels* (χ [8,9]) are introduced in the interface specification of the sub-models. The variables are connected by means of these elements. In most languages, a connection imposes an equality constraint on the variables. However, this is not the only meaning that can be associated with a connection. Depending on the physical quantity that the variables represent, continuous variables can be divided into two classes: *across* variables, that become equal, and *through* variables, that sum to zero, when connected [10]. In this article we concentrate on the semantics of connections that impose equality. However, the requirements of a connection semantics equally hold for connections of through variables. The only difference is that in that case, the connection equation is a summation to zero.

Specification of discontinuities in hybrid models is an important facility, provided by hybrid languages. One way to express discontinuities is by changing the values of the continuous variables, including the connected ones. Therefore, the language semantics has to define, what the values of the connected variables are after such a change, and how equality is (re-)established. When equality is (re-)established, the value of the variables of other sub-models may change as well. Particularly when sub-models are executed concurrently, the way equality is ensured among connected variables can alter the behaviour of the entire model. Therefore, it is essential to properly define the semantics of connections when one or both of the connected variables are changed discontinuously. In Section 2, situations are examined where model behaviour depends on the semantics of connections.

In Section 3, possible semantics of connections is treated. The different choices made by developers of hybrid languages are compared, and major advantages and disadvantages are given. In Section 4, the semantics of connections chosen in the χ language is discussed in more detail.

2. Discontinuous changes of connected variables

Hybrid languages provide discontinuity specification facilities in different forms. These facilities are used for several purposes. A discontinuity in the model can be used to describe an instantaneous action, the time scale of which is much smaller than that of primary interest. Also, external actions that change the system state can be described in this way. An example could be the interaction of the plant operator, who switches a valve on or off. Discontinuities can also be used to model actions that are themselves of no interest; only the results of these actions are important. A possible example is the process of filling a tank with reactants in preparation for a batch process.

One way of introducing a discontinuity is to change the value of a continuous variable. It can be changed explicitly by an assignment. Another, more subtle way to change a continuous variable is by altering the equations or other variables, which determine its value. After such a change, the model has to be transformed into a consistent state. A model is in a consistent state if *a)* the variables satisfy the equations and *b)* the connected variables are equal. In Section 3 we show, that some connection semantics ensure that the second requirement is automatically satisfied in any state of the system (e.g. shared variables). In other connection semantics the second requirement is the special case of the first one (algebraic variables). When equality is re-established among the connected variables, the variables of other sub-models may change as well, which influences the behaviour of these sub-models. In Sections 2.1 and 2.2 we examine, how the behaviour of the connected sub-models depends on the way equality is ensured among connected variables.

In the examples, the χ syntax is used. The elementary building blocks of a χ model are *processes*. Processes are executed concurrently. A χ process consists of several optional sections, and is declared in the following way:

proc *name (parameter declarations)=*
|[*variable declarations ; initialization*
| *links*
| *DAEs*
| *discrete-event statements*
]|

The process interface declares parameters, including discrete and continuous channels. A continuous channel is denoted by the —o symbol. In the declaration section, the continuous variables are declared with the :: symbol, whereas discrete variables are declared with the : symbol. In the declaration of continuous types and continuous channels, the units of measurement are given between square brackets. The null dimension is denoted by [-]. In the link section the continuous variables are linked to channels. For this, the —o symbol is used as well. Variables of different processes are connected if they are linked to the same channel. Equations are given in a symbolic, analytical form, where x' denotes the first time derivative of variable x. For assignments to continuous and discrete variables the ::= and the := symbols are used, respectively.

2.1 Scheduling dependency
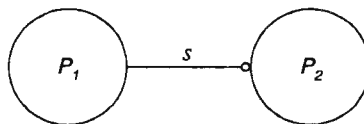
Figure 1. System of two processes.

This example illustrates how the value of a connected variable may depend on process scheduling. Consider the system consisting of two processes depicted in Figure 1. The two processes P_1 and P_2 are connected by channel s. This is specified in system S. *Systems* are used in χ to describe the fixed connection layout of the contained processes and subsystems.

$$\text{syst } S = \lVert \; s :: [-] \; | \; P_1(s) \lVert P_2(s) \; \rVert$$

The local variables s_1 in process P_1 and s_2 in process P_2 (see below) are connected to channel s therefore, they are equal. Process P_1 determines the value of s_1. The value of s_1 changes periodically with a period of 4 seconds. In the first 3 seconds of each period s_1 is 0, and in the remaining 1 second it is 1. Process P_1 is as follows:

```
proc P₁(s:: −∘[−]) =
‖ s₁ ::[−]; s₁ ::=0
| s −∘ s₁
| s₁′=0
| *[Δ3; s₁ ::=1; Δ1; s₁ ::= 0]
‖
```

In the discrete-event part a repetition construct is used: statements that are enclosed between *[and] are repeated forever. The Δ symbol denotes time passing. Processes executing a statement Δc, where c is a numerical expression, remain blocked for c seconds.

Every second process P_2 stores the value of s_2 in a list. Lists are declared in χ by appending the * symbol to a type. The empty list is denoted by [], and ++ is the append operator.

```
proc P₂(s:: −∘[−]) =
‖ s₂ ::[−], xs:real*
| s −∘ s₂
| xs:=[]
; *[Δ1 ; xs:= xs ++ [s₂]]
‖
```

The value of s_2 after 3 seconds, when P_2 executes the statement $xs := xs ++ [s_2]$, is determined by the connection semantics. If P_1 is executed before P_2, s_1 becomes 1. Depending on whether the value of s_2 is changed immediately or not, the value of s_2 in the assignment of xs is 0 or 1 respectively. One might find the desired output or even the correctness of this model questionable, since the value of s_2 is stored exactly at the time instant when s_1 changes in P_1. However, situations like this can not be prevented in complex models. The reason for this is that it is very difficult to oversee whether events in different sub-models happen at the same time. In fact, due to accumulating rounding errors even those events, which in principle should happen at distinct but very close time instants, may happen at the same time.

2.2 Conditional language constructs

In hybrid languages conditional constructs are used for several purposes. In many hybrid systems the describing set of equations depends on the state of the system. To model this, some form of a conditional equation is used. Such an equation specifies different sets of equations to be active, depending on the state of the system. In state-event specifications conditions are also used. The semantics of the connections affects the semantics of other language constructs as well. Suppose, for example, that in process P_2 the values of s_2 are sorted into two lists, according to whether they are smaller or equal, or bigger than 0.5.

```
proc P_2( s:: --o[-]) =
I[ s_2::[-], n:int, xl,xh:real *
|  s --o s_2
|  n:=0; xl:=[] ; xh:=[]
; *[∆1; [ s_2 ≤ 0.5  →  n:=n+1; xl:=xl ++ [ s_2 ]
        [] s_2 > 0.5  →  n:=n+1; xh:=xh ++ [ s_2 ]
        ]
   ]
]|
```

For selection, the guarded command of the form $[b_1 \rightarrow S_1 []\ b_2 \rightarrow S_2 [] ... []\ b_n \rightarrow S_n]$ is used. The boolean expression $b_i\ (1 \leq i \leq n)$ denotes a guard, which is open if the expression evaluates to true, and is otherwise closed. After evaluation of all the guards, one of the alternatives with an open guard is chosen for execution. In this example, a choice is made based on the value of s_2, therefore, execution of P_2 depends on the connection semantics. If after 3 seconds P_1 is scheduled first, and the assignment to s_1 results in the immediate change of s_2, then the second alternative ($s_2 > 0.5$) is chosen, otherwise the first one. Furthermore, if the concurrent assignment of s_1 ($s_1::=1$) in P_1 changes the value of s_2 between the evaluation of the guards and the append action, then the first alternative ($s_2 \leq 0.5$) is chosen in P_2, yet the value 1 is appended to list xl. Obviously, this would not be the desired behaviour of P_2. Normally, list xl should only contain values smaller or equal than 0.5, while values bigger than 0.5 should be stored in xh.

The example illustrates that when conditional language constructs use a connected continuous variable two problems may arise depending on the connection semantics. One problem is that the choice may depend on process scheduling. For the purpose of modelling industrial systems it is desirable that operations on a connected variable do not result in non-deterministic behaviour. The second problem is that after an alternative with a true condition is selected, the condition might immediately become false. This means that the condition can not be used as post-condition, when reasoning about the model behaviour.

3. Semantics of connections

In this section different connection semantics are surveyed. The purpose of this survey is to present different possibilities and to examine how different semantics determine, and in most cases restrict, the modelling style. Wherever possible, we mention some simulation languages that employ the given approach. However, it is not our aim to detail the consequences of choosing certain semantics in a given language. In order to do this, a language considered would need to be studied in full detail, and special language characteristics would need to be revealed. In many cases, however, there is not enough documentation available.
From the available documentation it appears that often the nature of the language guarantees that situations shown in the previous section do not occur. Furthermore, the way certain language constructs can be used and combined may be restricted in order to decrease the chance for erroneous or ambiguous models. In gPROMS, for example, two sub-models cannot manipulate the same connection equation simultaneously.

3.1 Shared variables

Connected variables become a single entity shared by the sub-models. It implies that changes to a variable in one sub-model are immediately visible in the other sub-model. This approach is implemented in ASCEND, if streams are connected with the aid of the ARE_THE_SAME operator. Streams can be merged in gPROMS as well, if the model's interface with its environment corresponds exactly to the interface of its sub-model. The disadvantage of this approach is that in parallel languages the simulation result may depend on the actual process scheduling. Furthermore, as the second example of the previous section illustrates, the semantics of the conditional language constructs may be affected as well. Solutions used in parallel programming languages to control access to shared variables (semaphores, access protocols) would eliminate the simplicity gained by merging the variables, and probably would be an unacceptable burden in

most models of industrial systems.

3.2 Causal connections

Connections define a causal interaction between the subsystems, where in each connection one side acts as input and the other side as output. One problem with this approach is that the direction of the information flow is fixed for each component. It restricts re-use of a component in other environments. Another restriction is that the connections should not lead to algebraic loops. In order to satisfy this requirement, the model may need to be transformed.

General purpose simulation packages like ACSL [11] and SIMULINK [12] use this approach. These languages require the Differential Algebraic Equations (DAEs) of each subsystem to be written in explicit state-space form $\dot{x} = f(x, u)$, $y = g(x, u)$, where u is used as input and y as output. Transformation into explicit state-space form (if possible) requires extra engineering skills and the resulting equations often do not reflect the natural system structure. With the advent of DAE solvers that can solve the implicit state space form $f(\dot{x}, x, t) = 0$, hybrid languages do not need to employ causal relations any more [13,14].

3.3 Master object

From the model's topology, first the sets of connected variables that are equal, are identified. For each set, a master object is introduced, which holds the value of the connection. This value is distributed among the variables at the end of each continuous phase. During the discrete-event phase, the sub-models have access to their (local) variables only. Changes of these variables may lead to a temporal inconsistency in the system. The system is brought back to consistency, by updating the master object, and distributing its new value among the variables. This idea is inspired by distributed systems, where several processes share a common memory. In order to allow concurrent accesses to the same data, duplicate copies are made of the shared block (i.e., block replication). For data coherence control, one copy is designated as a master copy, the value of which is distributed among the other copies. The difficulty in hybrid systems is to find a suitable protocol that determines the value of the master object. The mechanism has to be simple and intuitive, such that developers of complex models can oversee the system behaviour. Also, the protocol may have to comply with the way non-connected variables are assigned. See the difference between assignments to differential and algebraic variables in Section 4. A possible choice could be to assign the master object each time one of the connected variables is assigned. In this way, the master object always holds the last change that has been made to the connection. Its value could be distributed only at the end of the discrete event. However, the value of the master object would depend on the process scheduling. Another difficulty with using a master object is that it can hardly be generalised for connection of through variables.

3.4 Algebraic equations

Connections define an additional algebraic equation between the connected variables. This is the case in gPROMS, Dymola and in χ. However, the question remains what the semantics is of such an equation, i.e., when is it valid and how is equality ensured after a discontinuous change to one or both of the connected variables. Also, it is not always clear whether the semantics of such an equation is the same as that of other equations. Furthermore, if such an equation is replaced at a lower level by one shared entity due to optimization, the first alternative, (a shared variable) is implemented. Such a replacement is only allowed if the semantics of a connection equation between two variables is identical to that of referencing a single shared entity by both variables.

4. Semantics of model composition in the χ language

The χ language is a concurrent language. In order to avoid the disadvantages of using shared variables or a master object in parallel languages, algebraic equations have been chosen as the semantics of connections. In the sequel, the precise meaning of such a connection and the consequences of this semantics are explained.

The χ language is a hybrid specification language, that has been designed to be suitable for modelling discrete-event, continuous-time and hybrid systems. Simulation of a hybrid χ model can be described as a

273

sequence of continuous phases alternated with discrete phases at certain time points. The continuous-time behaviour is given in the form of DAEs, while discrete actions are specified in a CSP-like [15] concurrent programming language. Variables representing the system dynamics are continuous variables; those that occur with primes in the DAEs are differential variables, whereas other continuous variables are algebraic. The values of continuous variables are calculated from the DAEs; assignments determine their value only for the current time point. The values of discrete variables are determined by assignments. They hold their value between two subsequent assignments. In a χ model all variables are local to their processes. This complies with the language philosophy: data exchange among processes in discrete sub-phases takes place only by communication through discrete-channels. Furthermore, using only local variables turns out to be essential for a simple and robust connection semantics.

4.1 Connections

A connection between two continuous variables of different χ processes implies an additional algebraic equation. To keep the language small and robust, the semantics of a connection equation has been chosen to be identical to that of any other equations defined in the processes. The semantics is as follows. Equations are valid from the beginning until the end of a continuous-phase. The discrete-phases are divided into sub-phases. Each sub-phase ends when the discrete bodies of all the processes are blocked, waiting for events to happen. During discrete sub-phases the equation consistency may temporarily be lost due to discontinuous changes of continuous variables, or due to assignments to discrete variables appearing in the equations. At the end of each sub-phase a new consistent initial state is calculated, so that equality between connected variables is re-established.

4.2 Initial state calculation

The initial state is calculated by solving the system for the algebraic variables and the time derivatives of the differential variables. The differential variables are taken as known. Steady state calculation is requested, if time derivatives of the differential variables are assigned. In this special case, the value of the differential variable, the derivative of which is assigned, is calculated and its time derivative is taken as known. The language syntax supports new state calculation by allowing a special assignment symbol $::=$ to be used only for assignments to differential variables (e.g. $v::=0$) and, in case of steady-state calculation, for assignments to time derivatives of differential variables (e.g. $v'::=1$). Algebraic variables can only be given an "initial guess". For this purpose, the $:\sim$ symbol is used as in a $:\sim 10$. It is necessary to give algebraic variables an initial guess if more solutions are possible, or if the initial state solver requires a good starting value. The initial guess is used only to ensure that the solver converges to the required solution. Consider for example the equations describing a resistor and a capacitor connected in series with a voltage source.

$$
\begin{aligned}
u_R &= R \cdot i \\
C \cdot u_C' &= i \\
u_0 &= u_R + u_C
\end{aligned}
$$

The only differential variable is u_C, therefore its value can be freely chosen, e.g. $u_C::=10$. Algebraic variables (u_R, i) can take an initial guess, for example, $u_R :\sim 5$, $i :\sim 0.5$. As a result, the values of u_R and i change temporarily. However, when a consistent state is calculated, their values are determined by the equations.

4.3 Consequences

An advantage of treating connection equations in the same way as other equations becomes apparent in the situation where a differential variable is connected to an algebraic one. The value of the differential variable can only be changed by an assignment in the process where it appears differentiated in the equations. Consider, for example, the system of a mass m, connected by a linear spring with stiffness k as shown in

Figure 2.
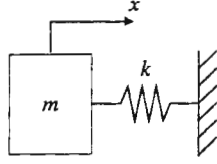
Figure 2. Mass-Spring system.   Figure 3. Mass-Spring model.
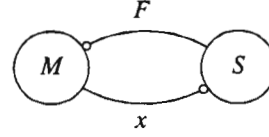
The χ model which describes this system consists of two processes: M and S, representing the mass and the spring. The processes are connected by channels F and x, which represent the force applied by the spring and the position of the mass, respectively. In process M the Newtonian equations describe the movements of the mass.

```
proc M(F::−∘[N], x:: −∘[ m ], m:real)=
I[ F₁::[N], x₁::[m], v::[m / s]; x₁::=0.3
  | F −∘ F₁, x −∘ x₁
  | F₁ = m·v′
  , v = x₁′
]|
```

```
proc S(F::−∘[N], x:: −∘[ m ], k:real)=
I[ F₂::[N], x₂::[m]
  | F −∘ F₂, x −∘ x₂
  | F₂ = k · x₂
]|
```

```
syst MS =
I[ F::[N], x::[m]
  | M(F,x,0.1) II S(F,x,0.05)
]|
```

In this model x_1 is a differential variable that is connected to an algebraic variable: x_2. This means that the connection equation $x_1 = x_2$ is added to the set of active equations. The value of x_1 can only be changed in process M, for example, by means of an assignment. Since x_2 is algebraic variable, it can only be given an initial guess. However, this does not have any effect on the value of x_1 because, when the initial state is calculated, x_1 is constant and x_2 is determined by the connection equation. Allowing only one process to change a differential variable discontinuously promotes more disciplined modelling.

Other situations, where the value of the connection should be changed in both processes are less common. When two algebraic variables are connected, like in the above example F_1 and F_2, the values of both of them are calculated from the equations. Therefore, it is unlikely that their value should be changed explicitly. The language reflects this: both can only be given an initial guess. Finally, when two differential variables are connected, one becomes dependent. This is the case, for example in multibody systems, where rigid bodies are connected to each other by joints that restrict the relative motion between pairs of bodies. The variables that represent the position of the bodies are differential, since they appear differentiated in the equations that describe the system motion. However, the position of a connected body can not be freely chosen; it is (partially) determined by the position of the body, to which it is connected. In general, connection of differential variables means, that one reduces the degree of freedom of choosing the values of differential variables. Furthermore, the index of the DAEs is increased, which leads to numerical

275

difficulties.

Another advantage of the connection semantics chosen in the χ language is that discontinuous changes to connected variables do not alter the behaviour of connected processes within a discrete sub-phase. This is because all variables are local to the processes and equality is established only at the end of the sub-phase. This means that process behaviour in a discrete sub-phase can be fully analysed based on the values of local variables, independently of whether a variable is connected or not.

5. Conclusions

The possibility of discontinuous changes of connected variables requires the semantics of connections to be properly defined. Improperly defined connection semantics can *a)* introduce undesirable non-determinism and *b)* lead to undesirable semantics of conditional language constructs.

The semantics of model composition can be defined in several ways. However, use of shared variables may lead to less robust models, whereas use of causal connections imposes restrictions on the modelling style.

By choosing the semantics of connections to be identical to that of other equations, the hybrid χ language is kept small and robust. In combination with keeping all variables local, this results in simple and elegant model composition semantics. Other beneficial consequences of this combination are that *a)* a differential variable can be changed in one process only and *b)* model behaviour in a discrete-event depends on the values of local variables only.

References

[1]   S.E. Mattson. "On model structuring concepts." In *Preprints of the 4th IFAC Symposium on Computer-Aided Design in Control Systems*, pp. 209-214, Beijing, 1988.

[2]   P.I. Barton. *The modelling and simulation of combined discrete/continuous processes*, Ph.D. Thesis, University of London, 1992.

[3]   *ASCEND IV Manual*, Carnegie Melon University, 1997.

[4]   P. Piela, T. Epperly, K. Westerberg, and A. Westerberg. "ASCEND: An object-oriented computer environment for modeling and analysis: the modeling language." In *Computers and Chemical Engineering,*15(1), pp. 53-72, 1991.

[5]   H. Elmqvist, D. Brück, and M. Otter. *Dymola – User's manual*. Dynasim AB, Lund, Sweden, 1996.

[6]   H. Elmqvist, F. Cellier, and M. Otter. "Object-oriented modeling of hybrid systems." In *Proceedings of European Simulation Symposium*, ESS'93, The Society of Computer Simulation, October, 1993.

[7]   S.E. Mattson, M. Andersson, and. K.J. Åström. "Object-Oriented modelling and simulation." In Linkens Ed., *CAD for Control Systems*, chapter 2, pp. 31-69. Marcel Dekker Inc, New York, 1993.

[8]   J.M. van de Mortel-Fronczak, J.E. Rooda, and N.J.M. van den Nieuwelaar. "Specification of a flexible manufacturing system using concurrent programming." In *Concurrent Engineering: Research and Applications*, 3(3) pp. 187-194, 1995.

[9]   D.A. van Beek, S.H.F. Gordijn, and J.E. Rooda. "Integrating continuous-time and discrete-event concepts in modelling and simulation of manufacturing machines. "In *Simulation Practice and Theory*, 5, pp. 653-669, 1997.

[10]   H.E. Koenig, Y. Tokad, and H. Kesavan. *Analysis of Discrete Systems*. McGraw-Hill, 1967.

[11]   E.E.L. Mitchell, and J.S. Gauthier. "Advanced continuous simulation Language(ACSL)." In *Simulation*, Vol. 26, No. 3, pp. 72-78, 1976.

[12]   Simulink User's Guide, The MathWorks, Inc., 1993.

[13]   H. Elmqvist, and S.E. Mattson. "MODELICA - The next generation modeling language - an international design effort." In *Proceedings of the 1st World Congress on System Simulation* , WCSS'97, Singapore, 1997.

[14]   F.E. Cellier, and H. Elmqvist. "Automated Formula Manipulation in Object-Oriented Continuous System Modeling." In *IEEE Control Systems,no. 13(2)*, pp. 28-38, 1993.

[15]   C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, Englewood-Cliffs, 1985.

ENTERING BOND-GRAPHS AS GRAPHS

Alojz Hussu

Faculty of mechanical engineering, Aškerèeva 6, 1000 Ljubljana, Slovenia, alojz.hussu@fs.uni-lj.si

Abstract

To ease the use of computers for simulation and analysis of dynamic systems described with bond-graphs, a program was written for entering bond-graphs into computer directly as graphs. The program enables definition and description of elements, interconnections, inputs and outputs as well as other necessary parameters for a successful simulation and analysis.

Why bond-graphs?

Different modelling approaches are used for different systems, depending on the field of application. If a system incorporates several subsystems, for example electrical, mechanical, thermal and/or others, it is useful to use such a modelling tool, that all subsystems are modelled in some uniform way, with similar symbols for components that have similar behaviour. Bond-graphs provide symbolism and notation for modelling of either simple or composed systems [1,2,3,4].

Simulation and analysis with BONK

There are now several simulation software packages available. Some years ago a package named BONK was written by the author [6] for simulation and analysis of linear and nonlinear systems. Through simulation analysis of time responses of linear and nonlinear [7] systems is possible. For linear systems only there are more kinds of analysis available: mathematical models, continuous and/or discrete, transfer functions, continuous and/or discrete, frequency characteristics can be analysed, either as Nyquist or Bode plots, the roots of the characteristic equations in s-plane, the roots of the transfer functions (zeros and poles), the roots of the characteristic equation of the discrete transfer functions in z-plane, roots of discrete transfer functions and root-locus plots for either continuous or discrete transfer functions. As a way to determine the sensitivity of systems to certain parameters or to test robustness of control systems, the program BONK provides means to vary selected parameters and display the influence of the variation on either the time responses, the frequency characteristics or the roots of the systems.

Descriptions of systems can be given in several ways, however for the reasons given above and the intended purpose of the BONK program (mainly educational), the description of systems with bond-graphs was considered very useful. As at the time when BONK was conceived only rudimentary computer equipment was available, the usual procedure for using the program BONK was to hand draw the bond-graph of the system to be analysed, then transform it into a table and write with a text editor a file with several named blocks in which elements of the bond-graph, their values and bonds themselves are given as well as the necessary additional data. Some of the additional data are connected with the bond-graphs, for example if the assignment of causality is required on certain bonds or if there is a preferred direction of power on some of the bonds or if some of the variables or signals are declared as active or if there are modulated elements (in that case a block with functions is given; BONK includes an interpreter to compute the functions). Otherwise the program BONK determines power directions and causality by itself. Some additional data are connected with the kind of analysis. If variation of parameters of certain elements is desired, selected elements to be varied are specified. If time responses or discrete mathematical models or discrete transfer functions are desired, time interval has to be specified. To compute the time responses a discrete model of the system is used. Of course, outputs must be defined for any kind of analysis.

To reduce this duality of descriptions, a hand made drawing of the bond-graph on the one side and a file with data entered into the computer on the other side, to reduce the problem of keeping the two descriptions synchronised, to automate the numbering of elements, to offer a possibility to add some other relevant information to the bond-graph, to easily change the structure of the graph, to easily maintain it and keep updated and to have a simple way to incorporate bond-graphs into other documents, reports, as a way of

describing, documenting, communicating bond-graph descriptions of systems, a program was written, that offers a graphical interface to the BONK program and at the same time quite independently acts as a drawing tool for bond-graphs.

Drawing bond-graphs

For technical drawing and diagramming with a computer a lot of packages exist, some more general, for example AutoCAD, DesignCAD, CorelDraw, CorelFlow, Visio, some more specialised. There are also some programming languages that supply efficient graphical support, like Delphi, versions of C++, Visual Basic and others. On the other hand there are some software packages specialised just for drawing bond-graphs [3,4,5]. In [8] different approaches were analysed. After taking all relevant circumstances into account a conclusion was made to write a program for drawing bond-graphs on the basis of Borland's Delphi.

To shorten the story, on the next figure the window that the program offers for drawing the bond-graphs is shown. As for demonstrations of the program a laptop computer is used with a limited screen surface, instead of usual drop-down menus especially for activities with files, a panel is used where all the necessary buttons are installed. They form three groups. The first from left enables opening and saving of files representing bond-graphs, saving BMP files, printing of bond-graphs and erasing graphs. The next group supplies elements of the bond-graphs (s and p is used instead of 1 and 0; no fields are supported at the moment; sys is for future extension). The last group is for editing, moving, deleting and some support tasks.
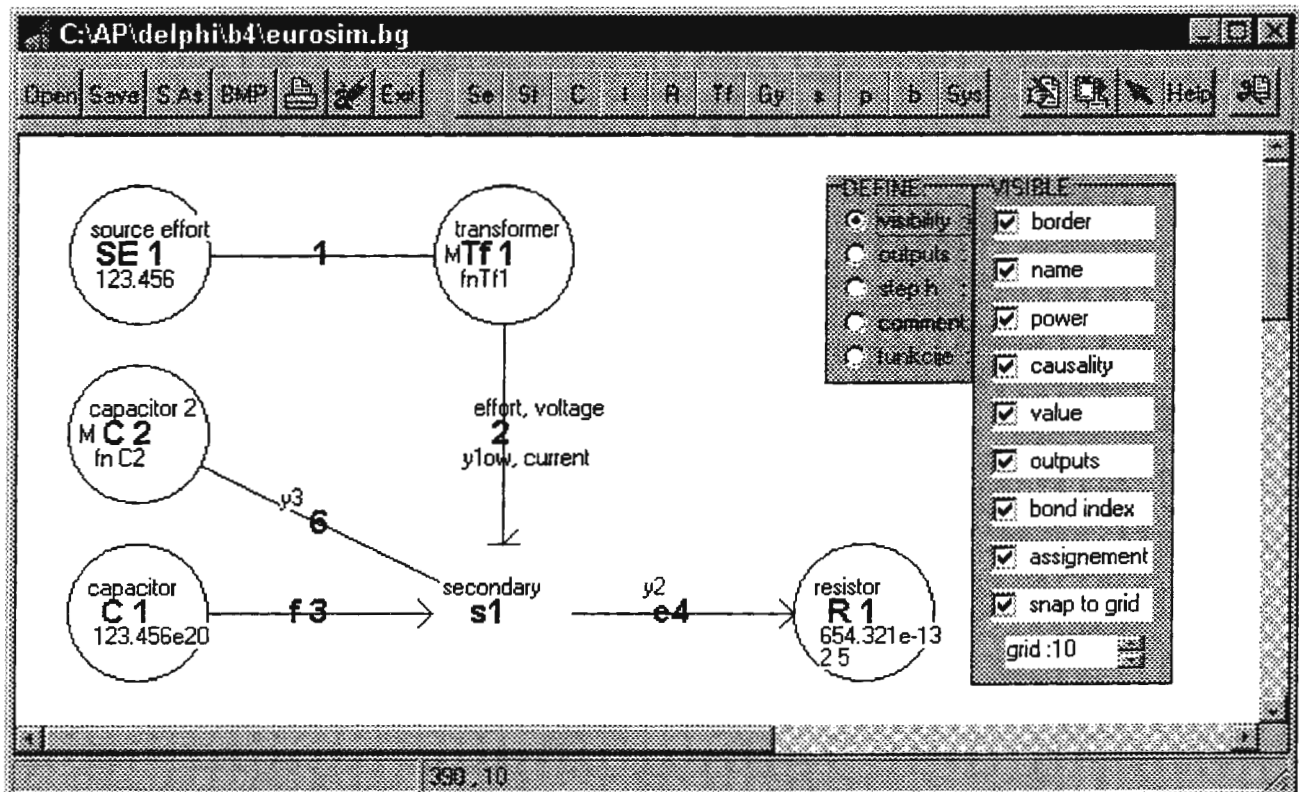
When inserting elements of the bond-graph, only type of the element is prespecified and the current index of the element is supplied. All other parameters, descriptive names of elements (above the name of the element, consisting of the type and index - bold characters), index itself, value of the element (below the name), letter M telling that the element is modulated (if an element is modulated, no value is given, instead of the value a function is initialised, name of the function is automatically defined as 'fn'+'name of element'; functions are later specified in a separate block as text). Below the value the optional variation of the element can be given as two numbers, one integer, telling how many changes in one direction is desired (twice as many changes are actually made, in positive and negative direction around the nominal value), the other real, telling the change of the nominal value in percent.

For connecting elements (s an p) the name of the common variable can be given or some other text.

On bonds either effort variable and/or flow variable can be named, power direction and causality can be prespecified, for active bonds activation must be assigned. Keys > or < are used in appropriate edit boxes when placing elements on the bond-graph or when editing elements for power direction, causality and activation, on bond-graph power direction and causality is shown in a usual way, activation is shown as full arrow, but in addition the letter of the activated variable, either effort (letter e) or flow (letter f) is shown on bond-graph in front of the index of the bond (bonds 3 and 4 on the example above).

For transformers and gyrators it was planned to draw them together with both bonds they are connected to to resolve the ambiguity which side is primary and which is secondary (there is a difference if one works with mechanical transformers, gear transmissions for example, where flow variable is emphasised, or electrical transformers, where effort variable is usually emphasised). Now bonds are always drawn separately between two already existing elements, however when adding second bond to a transformer or gyrator, direction is determined (from first bond added to the second) and on both bonds connected to the respective element direction is shown (numbers in selected color: number 1 for primary side on one bond and number 2 for secondary side on the other bond). Direction can be changed when editing transformers or gyrators.

Double clicking on empty space shows a pop-up menu enabling the user to select what is visible on the bond-graph (the bare minimum are names of elements and lines representing bonds, see the figure below), to define outputs, time step h for discrete models and time responses, to comment the graph and to define equations, when there are modulated elements in the bond-graph. Nonlinearities and input functions that are not step functions are treated as modulation of elements. Outputs can be shown on the graph, they are represented as letter y with the appropriate index above the bond name (when effort variable is selected for output, bond 4 and 6 on graph above) or below the bond name (when flow variable is selected for output, bond 2 above).

When drawing is finished and button Save is pressed (or Save As), two files are written. The first stores the graphical data of the bond graph as well as all other information of the bond-graph so as that further work on the bond-graph is possible later on. The other file stores parameters of the bond-graph in a way as requested by the BONK program for input files. Graphs can be printed or included in other documents as BMP files.
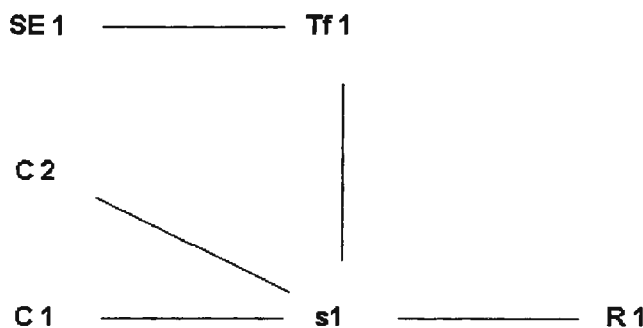
Conclusion

A program for drawing bond-graphs is presented. The goal of the program is achieved. However the next step in development is to join this drawing program with the BONK program to enable the user more effective interactivity in simulation and analysis of systems. The graphical routines in BONK (runs under DOS) must be converted to routines that run under windows. Later still the inclusion of fields is planned.

References:

[1] Karnopp,D.C.,Margolis,D.L.,Rosenberg,R.C., System Dynamics: A Unified Approach, Wiley, 1990

[2] Wellsted, P.E., Introduction to System Modelling, Academic Press, 1979

[3] Thoma,J.U., Simulation by Bondgraphs, Springer Verlag, 1990

[4] Breedveld,P.C., Dauphin-Tanguy,G.,(editors), Bond Graphs for Engineers, North-Holland, 1992

[5] Bacharach,B.I., Annotated Bond Graphs - A Communication Tool, Trans.of ASME, J.of Dynamic Systems, Measurement and Control,118,4,Dec.1996

[6] Hussu,A., Modelling with bond-graphs (in slovene), Proceedings of the First Electrotechnical and Computer Conference ERK'92, Portoro , Slovenia,1992

[7] Hussu,A., Bond-Graphs and nonlinearities (in slovene), Proceedings of the Third Electrotechnical and Computer Conference ERK'94, Portoro , Slovenia,1994

[8] Hussu,A., Graphical interface for modelling with bondgraphs (in slovene), Proceedings of the Fifth Electrotechnical and Computer Conference ERK'96, Portoro , Slovenia,1996

SIMULATION OF ROUTE TIME TABLE OPTIMAL DIVISION FOR TOWN TRANSPORT VEHICLES.

Mikuláš ALEXÍK, University of Žilina, Slovakia.

1. Hierarchy of Control Proces Tasks.

Transportation as process is characterized [1] by moving goods or passengers and this is the reason why it is connected with necessity of control. It is introduced by dispatch control or planning of trajectory with feedback control of vehicle on its trajectory [2], [3], and [4]. Interactions in the development of businesses, sciences ant technique enabled to define two kind of problems in transportation control also in theoretical level. Mathematics solutions, modeling possibilities and explicit solutions showed that the tasks of transportation control were divided into two sphere of theory. The tasks of planning of transportation route were solved by methods of operational analysis, theory of graphs [2], [3] theory of games i.e. algebraic methods, what means that problem was solved as static one. Control of vehicle on its trajectory had usually dynamic feedback and was solved dynamically and was modeled by means of theory of automatic control, what means that problem was solved dynamically [3], [6]. From one side it was possible to talk about dispatch control and on the other hand it was feedback control or control in real time. But in every case it is complex process and one part of it is shown in Figure 1.

Technical possibilities of communications enable for state variables data collection and information transfer of one vehicle or stream of vehicles direct from trajectory. There is also possibility of transfer of control variables back to the vehicle on its trajectory. For example it is possible to control crossroads lights taking into account global stage of network, informations of drivers about connected communications and their density of transport what means dynamic directioning in the network, modifications of traveling time among the individual stops for drivers of busses, trams and underground vehicles. It means the applications of methods of automatic control also in the area of dispatch control [3],[8], because there are at disposal feedback informations gained from trajectory of individual vehicle (precise values of speed and position of vehicle). There are limited possibilities of transfer actuating variables "on trajectory of vehicle" but in some cases it is necessary (ships and planes) to transfer them.

Control of transportation process then can be understood as distributed, large scale, multilevel, hierarchical system of control. Example of such access to control of town transportation by means of light rapid transit (LRT) as tube vehicle is in the [8], for control of town commuters transportation [15], or for control of road transportation [5]. The applications of methods of (optimal) automatic control for general solution of transportation tasks was introduced in [3], [4], [6]. Possibility of realization of synergetic method for control of transportation process were applied in [9]. This contribution described the overview of tasks of simulation modeling for description of dynamic behavior of transportation means and process to create possibility for applications of synergetic methods further for control of transportation processes.

Simulation methods on the level of dispatch control can serve as helpful in the decision taking process or in situation control in real time only in the case that they are enough fast. Such kind of simulation models should be capable to simulate situation including dynamic of individual vehicle. This is the reason why is necessary to calculate also with dynamic of transportation events which can occur among the individual knots of transportation network.

The individual task which can occur where divided into three level as it was described and they are shown in the Figure 1. The analysis, mathematical description and solution including developed software is described in connection with aforementioned division of problem.

Mikuláš Alexík, Prof. dr.Ing. University of Žilina, Faculty of Management, Control and Informatics, dept. of Technical Cybernetics, ilina, Slovak republic. Email:alexik@frtk.utc.sk
He interesting of the area of Selftuning Adaptive Control and Simulation of Dynamics Transport Vehicles and processes. He also working as representative member of Czech & Slovak Simulation Society in the EUROSIM.
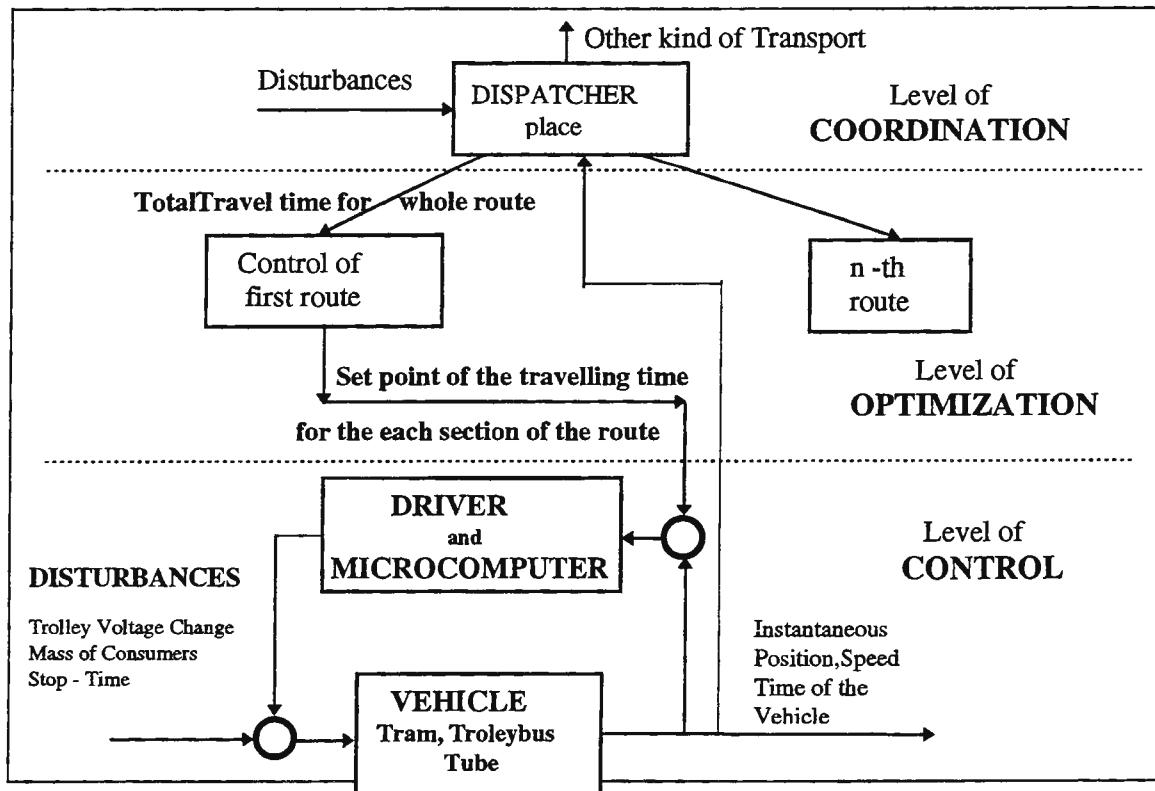
Figure. 1. Hierarchy of Control Proces Tasks

2. Optimal Trajectory of Vehicles.

Many works in our country and abroad has been done in the area of optimal trajectory computing [2], [6], [7],[8]. The problem was also solved in [4]. Software for calculation of optimal trajectory is part of the standard equipment for places which take part in transportation from the point of view safety and planning. Shape of optimal trajectory is usually two, three or four sections trajectory. As the individual phase is understood start-up (SU), constant speed run (CSR), run without force (RWF) and braking (BR). During realization of [4] the computer with connected of A/D and D/A converters was at the disposal what was reason for implementation of control algorhitm which is able to compute optimal trajectory in real time. Successful realization of previous control encouraged interest in computation of higher levels of control in real time. The tendency to dynamic control of transportation control abroad was checked in [3], [5], [6], [8]. Description of the linear and nonlinear algorhitms for computation of optimal trajectory can be followed next.

System of two differential equations (1) and (2) is solved for description of vehicle trajectory. In [4] it was solved for simplified problem when only linear part of vehicle traction resistance was calculated (3) and (4). It was enough for small slope of route. Part of solution was described in [14] and its enclosure described heuristic algorhitm for calculation of optimal trajectory in real time.

$$(dL/dt) = V \tag{1}$$

$$(dV/dt) = a - \left(k_0 + k_1 V + k_2 V^2 + S_r\right) \tag{2}$$

a – accerelation

k_0, k_1, k_2 empirical coeficients of travel resistance

S_r – average slope of route

$$(dL/dt) = V \tag{3}$$

$$(dL/dt) = a - \left(k_s V + S_r\right) \tag{4}$$

$k_s = \left(k_0/V_s\right) + k_1 + k_2 V_s$; V_s – average velocity

Algorhitm for linear model was used as first one to accelerate calculation under nonlinear model. The points of optimal trajectory were computed as it was shown in [7] but the calculation was simplified and generalized. The calculation of optimal trajectory starts from real point with negative signun of time, has complicated iteration and the constant profile of route is supposed. The algorhitm is not applicable in real time. Figure 3 shows the optimal trajectory of simple path with points which are necessary to calculate.
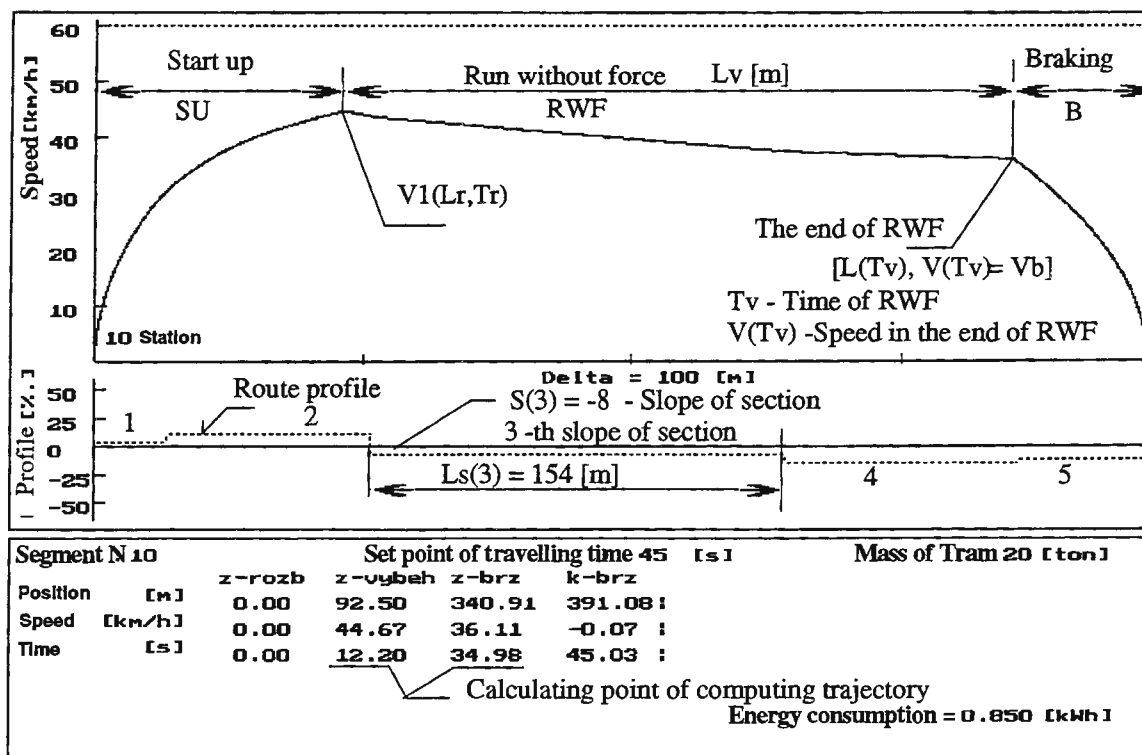
Obr. 3 Optimal trajectory of the tram.

The optimal trajectory is calculated next way. The relation (2) is replaced by (5) and after simplifying the integral (6) is necessary to solve. For this integral the substitution which depends od value on coefficients k_w, k_r and k_q is used in the relation (7). When (6) is solved the equation (7) is obtained. If integration constants from (7) are calculated from initial conditions the velocity (8) and position (9) for chosen time of the end of RWF is result. Eligible iteration method for this calculation can be used.

$$(dV / dt) = - (k_{0s} + k_1 \, V + k_2 \, V^2) \tag{5}$$

$$k_{0s} = k_0 \, S_r$$

$$\int \left[1/\left(k_{0s} + k_1 \, V + k_2 \, V^2 \right) \right] dt = - \int dt = - \left(t + C_1 \right) \tag{6}$$

$$D_1 = 4 \, k_{0s} \, k_2 - k_1^2 \; ; \text{pre} \quad S_r \geq 0 \Rightarrow D_1 > 0 \Rightarrow D = \sqrt{D_1}$$

$$-\left\{ (D/2) \; \text{arctg} \left[\left(2 \, k_2 \, V + k_1 \right)/D \right] \right\} = t + C_1 \tag{7}$$

$$C1 = - (2/D) \; \text{arctg} \left[\left(2 \, k_2 V_m + k_1 \right)/ D \right]$$

$$V(Tv) = \left[D/\left(2 \, k_2 \right) \right] \text{tg} \left\{ \text{arctg} \left[\left(2 \, k_2 \, V_m \, k_1 \right)/D - (D/2) \, T_v \right] - \left[k_1 /\left(2 \, k_2 \right) \right] \right\} \tag{8}$$

$$L(Tv) = \left(1/ k_2 \right) \ln \left\{ \left[\cos\left((D/2)(Tv + C_1) \right) \right]/ \left[\cos\left((D/2) \, C_1 \right) \right] \right\} \left[k_1 /\left(2 \, k_2 \right) \right] Tv \tag{9}$$

For the case if reduce slope is $S_r < 0$ and value $\mathcal{D}_1 < 0$ substitution (10) in the equation (6) must be used. Solution is described in equation (12) and (13).

$$D_1 \, k_1^2 - 4 \, k_{0s} \, k_2 \; ; D = \sqrt{D_1} \; ; \tag{10}$$

$$(1/D) \ln \left\{ \left[2 \, k_2 \, V + k_1 + D \right]/\left[2 \, k_2 \, V + k_1 - D \right] \right\} = t + C_1 \tag{11}$$

$$\text{for} \quad k_d = \left(D + 2 \, k_2 \, V_m + k_1 \right) / \left(D - 2 \, k_2 \, V_m - k_1 \right)$$

$$V(T_v) = \left[1/\left(2 \, k_2 \right) \right] \left\{ D - k_1 - \left[(2 \, D)/\left(1 + k_d \, \exp\left(D \, T_v \right) \right) \right] \right\} \tag{12}$$

$$L(T_v) = \left(1/k_2 \right) \left\{ \ln\left| 1 + k_d \, \exp\left(D \, T_v \right) \right| - \ln\left| 1 + k_d \right| - 0.5 \left(D + k_1 \right) T_v \right\} \tag{13}$$

In the described equations is the velocity $\mathcal{V}_m$ instantaneous velocity of the vehicle. In the each sampling interval is it necessary to solve equations (7-13), if is possible switch of run with force, because the prescribed travel time for the route will be realised with RWF. The progam for PC, which was realised , can solved looking for optimal trajectory of the vehicle which has minimal energy consumption, as can be see on the figure 3. Speed-tractive-effort curve and energy consumption of the tram were programming on the base of measurement [15]. Measurements were realised with the tram ÈKD Tatra KT8 Anakonda (VUD ilina researchers) on the route "Sokolovská - Palmovka " in the Prague. Measurements were realised with using the equipment Correvit ,which can realised optical measuring of speed, time and distances. Evaulation of simulation model of the tram are in the literature [45] where are compare trajectory of the tram from measurements and simulation experiments on the route "Sokolovská - Vitková". Evaulation were realised for every on 17th route segments in the both direction.

3. Identification of Human Dynamic during Control of Vehicle.

This task belong into area Man - Machine Systems (interface) and is coupled not only with dynamic of operator but also with dialog systems for decision making process control, system and situation identification, planning of future activities and their interactions with transportation process etc.
It was started with the most simple problem - identification of chanel "eye - hand" during control of technical process. Figure 4 shows the simplified scheme of experiment during identification and supposed model of chanel dynamic. The time delay of operator is variable and is described in the literature [11].
Further possible tasks are operator's following of harmonic and accidental signals practise of skillness till models of decision process during situational control. This problems is described in literature aimed in this area and results achieved are presented in references in the end of [11].
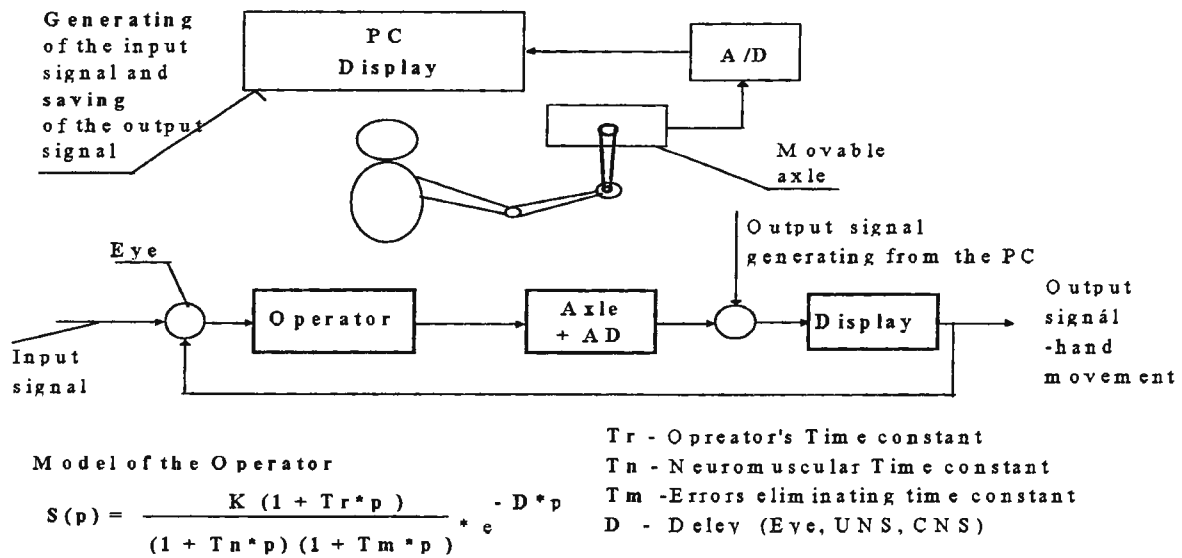
Figure. 4. Modelling of the chanel "eye-hand" dynamic during technical process control

4. Optimal Division of Whole Route Total Travel Time.

As a part of adaptive control of vehicle dynamic process is division of the total travel time (TTT) among the individual sections of the route. It was verified by software means known advance- division of energy comsumption as a function of prescribed travelling time done by hyperbolic description in [7]. For one line of solved route it is done by equations (14) - (17).

$$E_i = \frac{1}{a_i + b_i \, T_{u,i}} \quad (10) \qquad\qquad E_i = \frac{1}{a_i + (b_i / T_{u,i})} \quad (14)$$

a_i, b_i — paramters for hyperbolic model i − th partial section of line

$T_{u,i}$ — Demended travelling time i - th section

$$b_i = \frac{\left\{ n \sum_{i=1}^{n}[1/(E_i \, T_i)] - \Sigma(1/E_i)\, \Sigma(1/T_i^2) \right\}}{\left\{ n\, \Sigma(1/T_i^3) - \Sigma(1/T_i)\, \Sigma(1/T_i^2) \right\}} ; \quad a_i = (1/n)\,[\sum_{i=1}^{n}(1/E_i) - b_i\, \Sigma(1/T_i)] \quad (15)$$

$$(dE_1 / dT_{u,1}) = (dE_2 / dT_{u,2}) = ... = (dE_n / dT_{u,n}) = \text{konst.} = K \quad (16)$$

$$K = \sum_{i=1}^{n}\sqrt{(-b_i / a_i)} \left[T_{uc} + \sum_{i=1}^{n}(b_i / a_i) \right]^{-1} ; \quad T_{u,i} = \left[\sqrt{(-b_i)} / (a_i\, K) \right] - (b_i / a_i) ; \quad (17)$$

This solution was not enough precise for the sections with high slope and many speed limitations and is also slow for application in real time. New algorithm for these cases was developed . This heuristic algorhitm, which is described by equation (18) is based on the shape and time for the fastest and the slowiest way of travelling.

$$K = \left[T_{uc} - T_{min}(s,j) \right] / \left[T_{max}(s,j) - T_{min}(s,j) \right] ;$$

$$T_{u,i} = K\, T_{mn}[c(s,j),i] + T_{in}[c(s,j),i] \quad (18)$$

T_{uc} — Total travelling time

T_{max} — maximal travelling time for route in direction s, j - number of route

T_{min} — minimal travelling time for route in direction s, j

$T_{mn}[c(s,j),i]$ — maximal travelling time in the section i route s, j

$T_{in}[c(s,j),i]$ — minimal travelling time in the section i route s, j
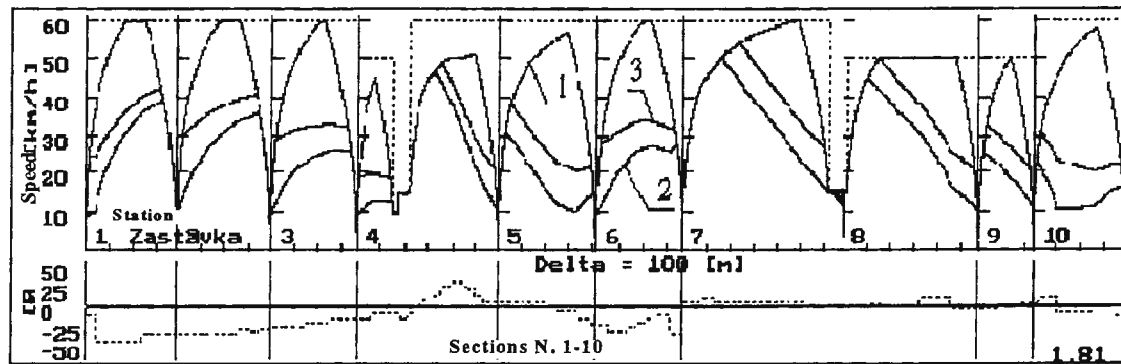
Figure 5. Trajectory of route sections and optimal trajectory.

The courses of trajectories for maximum possible travelling time and time optimal control shows better characteristics of route profile and is also sensitive to speed limits done by heuristic algorhitm. Example of division of total travelling time is shown in the Figure 5.

5. Simulation Model of Transportation Dynamic

The last task is to realize level of coordination in the hierarchical structure control from the Figure 1. To make possible dynamic control of vehicle on its route it is necessary by the mean of designed algorithm for dynamic control to correct deviations from "optimal travelling plan" or time-table. This is the reason for description of movement of all vehicles among the all stations as a dynamic system.

For this purpose the theory of state space used mainly in automatic control is very suitable. These kind of access are frequently used for control of underground and are also suitable for crossroads control. Based on the literature review the suitable access was found and it can be seen in the figure 6. Complete simulation experiments with state space model were not finished yet.
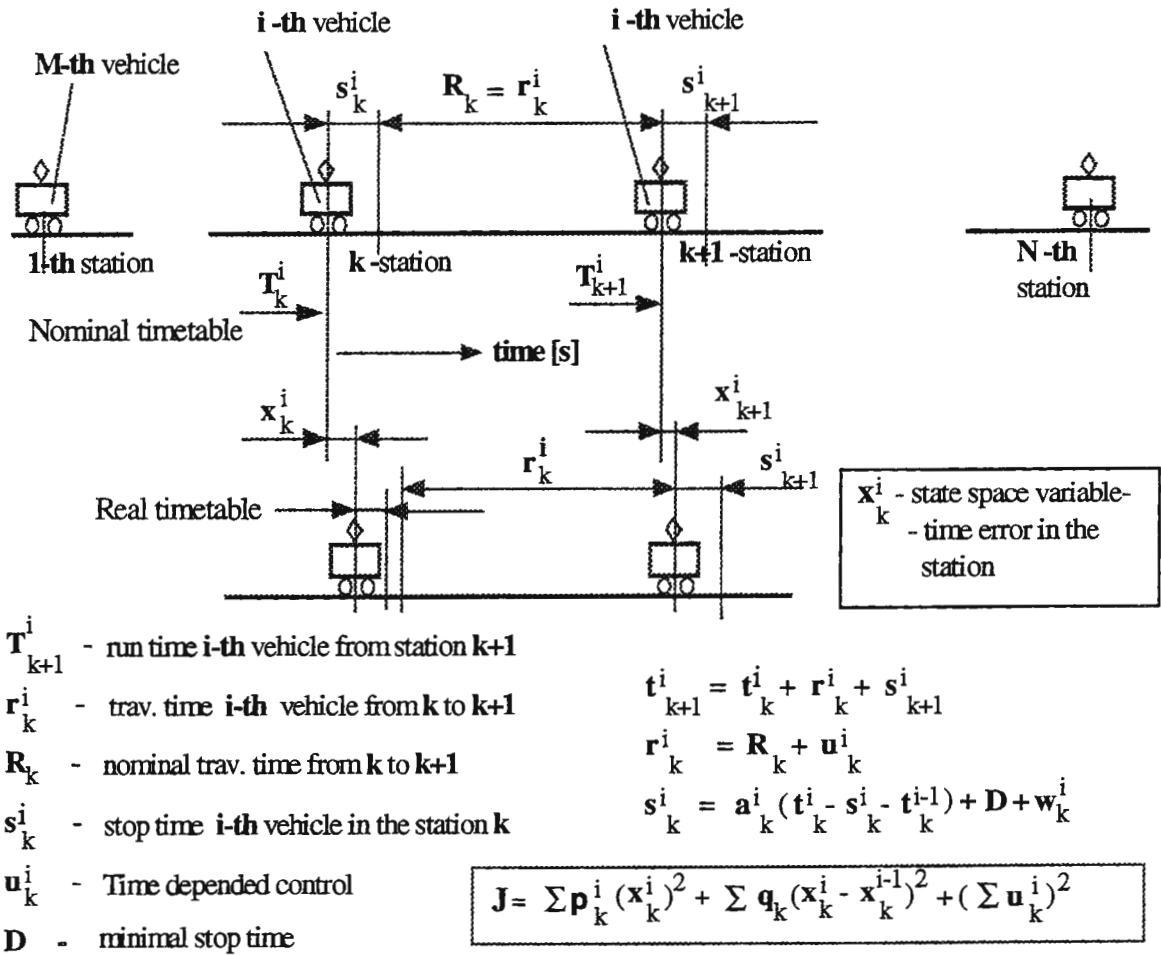
Figure 6. State space model of the line control.

T^i_{k+1} – run time i-th vehicle from station k+1

r^i_k – trav. time i-th vehicle from k to k+1

R_k – nominal trav. time from k to k+1

s^i_k – stop time i-th vehicle in the station k

u^i_k – Time depended control

D – minimal stop time

x^i_k - state space variable- - time error in the station

$$t^i_{k+1} = t^i_k + r^i_k + s^i_{k+1}$$

$$r^i_k = R_k + u^i_k$$

$$s^i_k = a^i_k (t^i_k - s^i_k - t^{i-1}_k) + D + w^i_k$$

$$J = \sum p^i_k (x^i_k)^2 + \sum q_k (x^i_k - x^{i-1}_k)^2 + (\sum u^i_k)^2$$

State space model with optimal control ensure suitable distribution of control actions over more sections, because it is clear to follow that deviation cannot be limited immediately in the first or second section. The same application of automatic control concepts to traffic flow modelling and control were first described in [3] for simulation of highway traffic flow control. State space model of one line dynamics is prepared and for more lines will be prepared and descriebed in the next contribution.

6. Conclusions.

Presented paper deals with description of optimal control of vehicle with some original contributions. The intentions are to spread possibility of this perspective application of automatic control, to simplify some theoretical but very composed access such way that it will be possible to verify them as real time algorhitm not only in laboratory conditions but also in transportation practise. This part describes some original access and also the stage of the area of vehicle and trafic process dynamic simulation and control.

Literature :

[1]   Èerný J., Kluvánek P.: TheTransport Teory (in Slovak). Veda, Bratislava 1990.
[2]   Skýva a kol. "Energy Optimal Control of Transportation Systems (in Czech). Praha, NADAS, 1987.

[3]   Papageorgiou M.:Applications of AutomaticControl Concepts to Traffic Flow Modelling and Control.   Springer Verlag, 1983.

[4]   Alexík M.: Hybrid simulation of the control of train running. (In Slovak) .Ph.D.. work. VŠDS-KTK ilina, may,1980.

[5]   Papageorgiou M.: Dynamic Modelling Assigment and Route Guidance in Traffic Networks. Transportation Research, dec 1990.

[6]   Kacprzak J.: Avtomatyka i sterovanie elektrycznych pojazdow trakcyjnych. Wydawnicztwa Komunikacji i Laczenosci. Warszava 1981.

[7]   Cenek P.: Energety optimal control of nonlinear dynamical system. (In Czech).Ph.D. work. VŠDS KTK ilina, sept. 1978.

[8]   Schnieder E., Kraft K.H.:Decentralised Hierarchical Control of Advanced Guideway Transit Systems.   IFAC/IFORS Symposium, Warsaw 1993.

[9]   Alexík M.: Using Synergetics Methods in Dynamics of Transportation Processes .(In Slovak). The works and Study of the VŠDS, seria kyb. zv. 7/89, str. 93-99.

[10]   Wang Xinqyuan: The research on the Dynamic Chaos Model for the Transportation Systems. 7th IFAC Symposium on Transport Systems. Tianjin, China, august 1994.

[11]   Bui Xuan Ngo: Simulation of the Dynamics of Transportation Processes. (In Slovak). Ph.D. work. VŠDS - KTK ilina, 1988.

[12]   Alexík, M.:Simulation of the Control of Running Train. (In Slovak). Automatizace 7/87. pp. 180-186.

[13]   Alexík M., Hynek J.: Optimalization of the control of Running Trams. Doprava 1/90, pp. 22-26.

[14]   Alexík M., Hynek J.: The Using of the On-Board Computer to an Optimal Control of Running Trams. Internatonal Conference. "Transportin the Hexagonal Countries". VŠDS ilina, may 1992.

[15]   Alexík M.: Simulation of the Running.Trams. (In Slovak) HZ 64/207/89, VUD ilina . ilina, January 1990.

[16]   Alexík M.:Simulation of the Control of Transportation Processes. (In Slovak) . Inauguration lecture. FR VŠDS, July 1995.

MICROCOMPUTER SIMULATION OF OPTIMAL FLIGHT PATHS

Tuomas Raivio, Kai Virtanen, Harri Ehtamo and Raimo P. Hämäläinen

Systems Analysis Laboratory, Helsinki University of Technology, P.O. Box 1100, 02015 HUT, Finland.
E-mail tuomas.raivio@hut.fi

Introduction

In aircraft trajectory optimization problems, flight paths that optimize a given payoff are sought. The payoff can be, for example, the elapsed time or the amount of burned fuel. Optimal control theory provides a framework for optimal trajectory design, but solving an optimal control problem is a laborious and time-consuming task. In this paper we introduce VIATO, 'Visual Interactive Aircraft Trajectory Optimization', which is a PC based optimization and simulation software for optimal trajectories. The paper is based on ref. [5].

VIATO aims at offering low-cost aircraft trajectory simulation and optimization possibilities for parties unfamiliar with mathematical modeling, optimal control and computer programming. It consists of an automated optimization server, a model server and a graphical user interface. When solving a particular optimization problem, the user of VIATO only needs to choose the aircraft type from the model library, select the payoff and specify the initial and terminal states. VIATO then generates the corresponding optimal control problem and solves it automatically. Finally, VIATO visualizes the results and allows for sensitivity analysis with respect to model parameters.

Some programs for solving optimal control problems are already available. Nevertheless, the user is expected to be able to formulate the problem and to carry out substantial source code modifications. Hence they mainly serve mathematically literate users.

Modeling considerations

A usual convention is to describe the system to be optimized by state variables and state equations. Specifying the state equations can be a complex problem even for experts. In VIATO, the mathematical modeling is avoided by specifying the aircraft model structure in advance. In addition to state equations and state and control constraints, the functional form of different payoffs is is fixed in advance, which makes the problem formulation unnecessary. For details and the structure of the aircraft model, see[5].

Aircraft of certain type obey structurally similar mathematical models that differ only in parameters. In VIATO, the parameters for different aircraft are stored in a model library. Many of the parameters, like the reference wing area and initial mass, are constants, but certain parameters, like the drag coefficients, are functions that depend on the state. The exact functional form of these parameters is not known, and for optimization purposes a smooth and continuous approximation must be generated from tabular measurement data. Besides keeping track of the model library, the model server of VIATO automatically provides these approximations. For one dimensional data, rational polynomials or alternatively splines are used, and for two dimensional data, like the thrust force of a jet engine, B-splines (see [1]) are employed. The user can affect the tradeoff between accuracy and smoothness of the approximations.

The optimization method

Once the optimal control problem is defined the optimization server of VIATO converts it into a nonlinear programming problem by discretizing it either with direct collocation or a method based on differential inclusions. For a comparison of these methods, see [4] and for an overall review of discretization methods, see [3]. The resulting relatively large optimization problem is solved using NPSOL subroutine [2] which applies

288

Sequential Quadratic Programming. The optimization server scales the decision variables and provides an appropriate initial guess for the iteration.

Another possibility for solving optimal control problems would be to solve the multipoint boundary value problem arising from the first order necessary conditions for optimality. This approach, however, is hardly suitable for automated solution, since the convergence domain of the solution methods is relatively small, and the correct sequence of free and constrained solution parts must be known in advance.

An example problem

We solve a minimum time trajectory for a generic fighter from the origin, the altitude of 5 km and initial velocity of 200 m/s along the positive x-axis to the point x=10 km, y=15 km, h=10 km with the final velocity vector again aligned with positive x-axis. The user interface is shown in figure 1. The problem type and aircraft model are selected with mouse-driven menus, and the initial and final states are specified. The drag and thrust parameters of the selected model together with the measurement data are shown in the model window. Solving this particular problem with a 200 MHz Pentium processor and Windows NT operating system takes 15 seconds. The solution is then plotted in a 3-dimensional cube that can be viewed from an arbitrary direction. The optimal flight time is 63 sec.
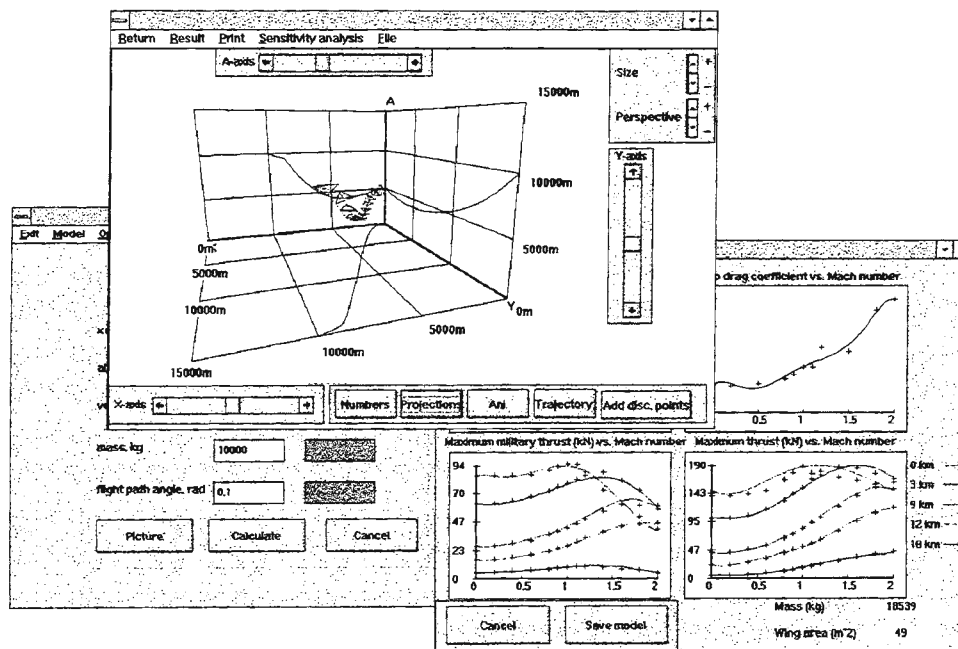
Fig. 1. The user interface of VIATO. Windows from the left: Initial and terminal state specification, result visualization, model management.

References

[1] De Boor. C.: A Practical Guide to Splines. Springer-Verlag, New York, 1978.

[2] Gill, P., Murray, W. and Wright, M.: User's guide for NPSOL 4.0. Report SOL 86-2, Dep. of Operations Research, Stanford University, USA, 1986.

[3] Hull, D.: Conversion of Optimal Control Problems into Parameter Optimization Problems. Journal of Guidance, Control and Dynamics Vol. 20, pp. 57-60, 1997.

[4] Raivio, T, Ehtamo, H. and Hämäläinen, R.P.: Aircraft Trajectory Optimization Using Nonlinear Programming. In: Dolezal, J and Fidler, J., Eds.: System Modeling and Optimization. Chapman & Hall, 1996, pp. 435-441.

[5] Virtanen, K., Ehtamo, H., Raivio, T., and Hämäläinen, R.P.: VIATO - Visual Interactive Aircraft Trajectory Optimization. Submitted to IEEE Transactions on Systems, Man and Cybernetics, Part C.

INTERACTIVE SPREADSHEET MODELLING OF REGULATION STRATEGIES FOR A LAKE-RIVER SYSTEM

Raimo P. Hämäläinen[1], J. Mäntysaari[2]

[1]Systems Analysis Laboratory, Helsinki University of Technology, P.O.BOX 1100, FIN-02015 HUT, Finland, Raimo@hut.fi
[2]Systems Analysis Laboratory, Helsinki University of Technology, P.O.BOX 1100, FIN-02015 HUT, Finland, juha.mantysaari@hut.fi

We describe a modelling tool running on the MS Excel spreadsheet software. With the introduction of the optimization solver in Excel it has become a practical platform for the development of visual interactive models even for relatively complex problems [1], [2], [3], [4]. The present model solves a dynamic multicriteria optimization problem describing the regulation of a lake-river system in Finland.

In environmental decision making the understanding and communication of the opinions of interest groups is very important [5] and modelling support is needed to separate facts from values [6]. The management of a lake-river system is a dynamic multicriteria optimization problem with strong control constraints and essential uncertainties [7]. Yet in the development of new water level management policies one would need to be able to interactively test different regulation policies. In our case stakeholders whose interests have to be taken into account in the policy include e.g. the power production companies (seasonally different value of hydro power), farmers (flooding), environmentalists (nesting of birds and hatching of fish, biodiversity, erosion), owners of lakefront recreational properties (quality of shores, flooding, stability of water level, property values), fishermen, boat owners and lake transportation (water level). The model will be used by the representatives of these stakeholder groups to learn about the different regulation alternatives.
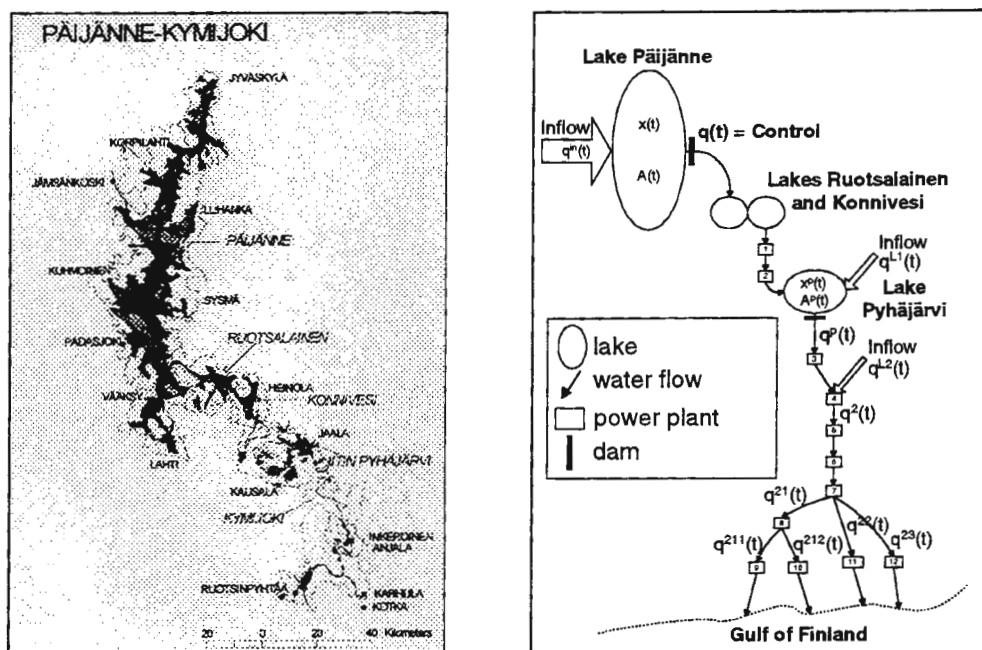
Figure 1: The Päijänne-Kymijoki lake-river system. On the left a map of the system and on the right a simplified system chart.

The system consists of four lake reservoirs only one of which is dynamically controlled. The flow rates in the connecting river are strictly constrained by min and max flows. Thus the problem is a constrained dynamic control problem. The seasonal drainage for which the regulation policy is planned is the historical data from a period of five years. The annual weather varies considerably and the solution has to be such that even under occasional rainy years the constraints are met as well as possible. The average benefits of the stakeholders are

related to different parts of the annual water level profiles in the lake and to the flow rate in the river. There are separate models used to compute the impacts from the levels and flows. The price of electricity is highest in the winter time and thus the power companies want to collect water by raising the levels towards autumn and then run the lakes into a low water level by the following spring. On the other hand such a strategy is not good for certain environmental criteria.
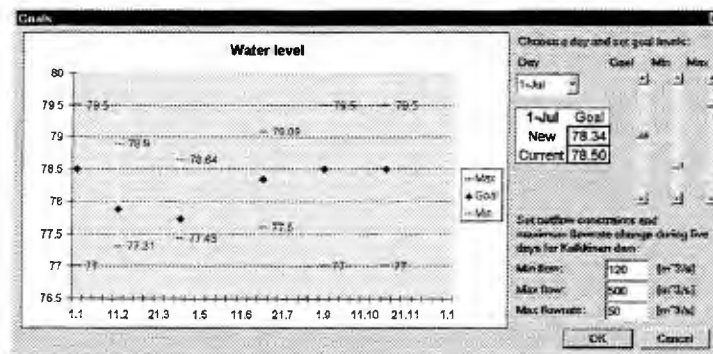
Figure 2: Setting goal levels and acceptable intervals.

The strategies are defined in terms of five or six annual goal points for the water levels with min and max acceptability limits for each point. (See Figure 2 for the window for setting these goals). The problem is to find the regulation strategy that would minimize the deviations from these points given a series of historical inflows. In the development of new regulation strategies one needs to be able to quickly test the consequences on the different criteria. Thus a tool like this is of utmost importance. For more details refer to [8].

References

[1] R. P. Hämäläinen and J. Parantainen. Load analysis by agent-based simulation of the electricity distribution system. In *Proc. of the 2nd IFAC Symposium on Control of Power Plants and Power Systems, SIPOWER'95*, volume I, pages 213–217, Cancún, Mexico, December 6-8, 1995.

[2] R. P. Hämäläinen. Agent-based modeling of an electricity distribution system. In *Proc. of the 15th IASTED International Conference on Modelling, Identification and Control*, pages 344–346, Innsbruck, Austria, February 19-21, 1996.

[3] R. P. Hämäläinen, E. Kettunen, and H. Ehtamo. Game modeling and coordination processes for two-level multi-agent systems. In *Proc. of the 16th IASTED International Conference on Modelling, Identification and Control*, pages 234-240, Innsbruck, Austria, February 17-19, 1997.

[4] J. Mäntysaari and R. P. Hämäläinen. Prototyping customized DSS with a spreadsheet program - The Power Agents system. *Journal of Decision Systems*, 6(4), 1998.

[5] M. Marttunen and R. P. Hämäläinen. Decision analysis interviews in environmental impact assessment. *European Journal of Operational Research*, 87(3):551–563, 1995.

[6] R. P. Hämäläinen. Facts or values - how do parliamentarians and experts see nuclear power? *Energy Policy*, 19(5):464–472, June 1991.

[7] Y. Y. Haimes. *Hierarchical analyses of water resources system: Modeling and optimization of large scale systems*. McGraw-Hill, New York, 1977.

[8] R. P. Hämäläinen and J. Mäntysaari. Interactive spreadsheet modelling of regulation strategies for a lake-river system. In *Proc. of the 17th IASTED International Conference on Modelling, Identification and Contro*, Grindelwald, Switzerland, February 18-20, 1998, to appear.

A DECISION ANALYTIC APPROACH TO FLIGHT SIMULATION

Kai Virtanen[1], Tuomas Raivio[2], Raimo P. Hämäläinen[3]

[1,2,3]Systems Analysis Laboratory, Helsinki University of Technology, P.O.Box 1100, FIN-02150 HUT, Finland
E-mail: Kai.Virtanen@hut.fi

Introduction

In this paper, a tool from decision analysis, the influence diagram [5], is applied to model the decision problems of a pilot during one versus one air combat. Analysis of air combat tactics and technologies as well as pilot training are expensive tasks, and not all air combat situations can be analyzed in practice. Therefore different optimization [11], game [9] and simulation models [7,8,10] have been developed. In simulation a model that imitates pilot decision making is needed. A decision model must be able to analyze an irregular flow of incoming data that describes a dynamically evolving environment. The available information is incomplete and the outcome of any particular action is at least partially uncertain. Furthermore, a decision model must involve competing objectives, such as the need to achieve a firing position and simultaneously avoid the opponent's weapons.

Decision making models of the existing air combat simulators are knowledge based expert systems [8,10] or value driven systems [7]. They utilize computational techniques of artificial intelligence (AI). However, expert systems have problems to model decision making under uncertainty since AI research has paid little attention to the importance of modelling human preferences and attitudes toward risk [4]. Decision theory, see, e.g., [1,2], extends the language of probability to express human preferences and risk attitude in uncertain situations. It provides a rational and structural framework for choosing between alternative courses of action when the consequences resulting from this choice are imperfectly known. Although decision analysis has been applied to various application areas, pilot decision making has not been modeled using a decision analytic approach.

Influence diagram

An influence diagram [5] graphically represents a decision process and describes the major factors of a decision problem. It consists of arcs and three types of nodes: decision nodes, chance nodes and deterministic nodes. A decision is represented by a decision node that includes different decision alternatives and can have numerical values associated with each alternative. A chance node represents an uncertain event or a random variable and have a numerical value and probability associated with each outcome. Deterministic quantities or variables are modeled by deterministic nodes whose value is either constant or a function of its inputs. The meaning of the arcs in the diagram depends on their destination node. Arcs leading into a chance or deterministic node represent probabilistic or functional dependence. Arcs pointing to a decision node imply time precedence and show which quantities are known to the decision maker before the decision is made.

An influence diagram associates a probability with each possible state of the world. A special deterministic node, called the utility node, includes a utility function that is an essential tool of multi attribute utility theory [6]. A utility function evaluates each state by assign real numbers to states. The objective of influence diagram analysis is to determine such alternatives that provide maximal expected utility. Furthermore, information gathering activities that will reduce the uncertainty of the model, can be modeled and evaluated using Bayesian reasoning, see, e.g.,[3].

Pilot decision model

A simple influence diagram modeling a pilot's decision process is shown in figure 1. The maneuvering decision is modelled by the *Maneuver* node and the missile launching decision by the *Launch Missile* node. The values of the predicted states related to each control alternative are given in the deterministic node labelled *Predicted State*. States are predicted by integrating the realistic equations of motion with different control alternatives. The deterministic node *Opponent's Predicted State* includes the exact values of the predicted opponent's state. The opponent's and decision maker's states define the momentary relative geometry that is included in the *True Geometry* node. The chance nodes *Sensor 1*, *Sensor 2* and *Sensor 3* model the decision

maker's observations from the state of the opponent. The chance node *Geometry Perception* represents the pilot's comprehension of the relative geometry of the air combat situation and is affected by the sensor information and by the values of the *Predicted State* node. The chance node *Situation Comprehension* includes the outcomes that describe the situation of the air combat from a decision maker's point of view. The probability that a launched missile will hit the opponent is modelled by a chance node labelled *Hit Opponent*. Prior probability distributions of uncertain outcomes are specified in advance. These distributions characterize the state of the air combat at a particular time. Posterior distributions are calculated using Bayesian reasoning and they are based on outcomes of the node *Geometry Perception*. The deterministic *Situation assessment* node includes a multi-attribute utility function that maps the values of the *Geometry Perception* node into a utility scale. Weights of the utility function and shape of single attribute utility functions depend on outcomes of the *Situation Comprehension* and the *Hit Opponent* nodes. A utility function describes the preferences of the air combatant that are results of the doctrine and training as well as the pilot's risk attitude.
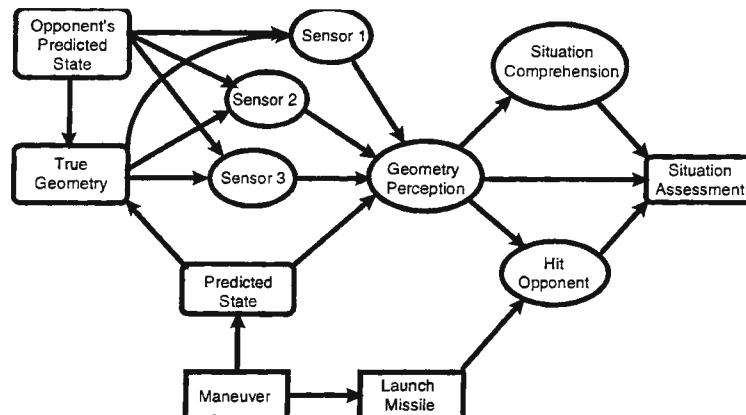
Figure 1: An influence diagram representing decision making during an air combat. Decision nodes are denoted by squares, chance nodes by ovals and deterministic nodes by squares with rounded corners.

Use of the model

An influence diagram is solved by converting it into a decision tree and by selecting the decisions that maximize the expected utility. Results include decision policy suggestions as well as risk and the cumulative utility profiles. Sensitivity analysis determines the most important factors in the model. It measures the impact of changing a deterministic value or a probability distribution on the decision policy. Effects of the pilot's risk attitude as well as the doctrine can be analyzed by modifying the utility function. Furthermore, the value and effect of extra observations from the opponent can be analyzed. The presented influence diagram takes into account all the available information to provide best possible logical decisions in air combat.

References

[1]   Anon., the web site of the Decision Analysis Society, http://www.fuqua.duke.edu/faculty/daweb/ .
[2]   Clemen R.T. (1996) Making Hard Decisions, An Introduction to Decision Analysis, 2nd Edition. Belmont, CA: Duxbury Press.
[3]   Gelman A, J.B. Carlin, S.H. Stern and D.B. Rubin (1997) Bayesian Data Analysis. London: Chapman & Hall.
[4]   Henrion M., J.S. Breese and E.J. Hortvitz (1991) "Decision Analysis and Expert Systems." AI Magazine 12. No. 4. pp.64-91.
[5]   Howard R.A. and J.E. Matheson (1984) "Influence Diagrams." In R. Howard and J. Matheson (eds.) The Principles and Applications of Decision Analysis, Vol. II, pp. 719-762. Palo Alto, CA: Strategic Decision Group.
[6]   Keeney R. and H. Raiffa (1976) Decisions with Multiple Objectives. New York: Wiley.
[7]   Lazarus E. (1997) "The Application of Value-Driven Decison-Making in Air Combat Simulation." Proc. of 1997 IEEE International Conference on Systems, Man, and Cybernetics. pp. 2302-2307.
[8]   McManus J. W. and K. H. Goodrich (1989) "Application of Artificial Intelligence (AI) Programming Techniques to Tactical Guidance For Fighter Aircraft." 1989 AIAA Guidance, Navigation, and Control Conference, AIAA Paper # 89-3525, 1989. pp. 851-858.
[9]   Raivio T., H. Ehtamo and R.P. Hämäläinen (1996) "A decomposition method for a class of pursuit-evasion games." Proc. of the 7th International Symposium on Differential Games and Applications, Vol. 2. pp. 784-795.
[10] Stehlin P., I. Hallkvist and H. Dahlstrand (1994) "Models for Air Combat Simulation." Proceedings of American Institute of Aeronautics and Astronautics Conference. ICAS-94-8.6.2. pp. 2190-2196.
[11] Virtanen K., T. Raivio, H. Ehtamo and R.P. Hämäläinen (1998) "VIATO - Visual Interactive Aircraft Trajectory Optimization." Submitted to IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews.

REAL MULTILEVEL MAINTENANCE SYSTEM PROCESS SIMULATION IN THE FUNCTION OF COST/BENEFIT ANALYZE OF INVESTMENT IN EQUIPMENT AND STAFF TRAINING FOR DIAGNOSTIC

Dr Petar Stanojevic[1], Dr Jugoslav Kodzopelic, Dr Vasilije Miskovic, Dr Radovan Maksic
Military Technical Academy - Department of technical support, Ratka Resanovica bb, 11000 Belgrade, Yugoslavia,

1 - e-mail: spetar@beotel.yu

ABSTRACT

Real multilevel maintenance systems (RMMS) process investigation was made in function of cost/benefit analyze of investment in equipment and staff training for diagnostics. Income of supported business system is directly proportional to a number of working technical items per day. That is the reason why average number of working crucial technical items (machines) per day (ANWTI), or average percentage of working technical items per day was chosen as a basic RMMS effects parameter. This approach makes minimizing of down time a crucial goal for RMMS. Complexity, stohastics and unstationary process character in RMMS and large number of influencing factors made modeling and simulation a base of investigation methodology tools. Investigation results showed that diagnostic accuracy for minimum step of change maximally influenced ANWTI. In the case of diagnostics' accuracy rising up for 10% RMMS can expect rising of ANWTI up for 15%. Diagnostic accuracy does not have significant influence on other needed maintenance resources in RMMS. Cost/benefit analyze lead to conclusion that diagnostic accuracy rising up for 6% will definitely approve all investments in equipment and staff training for diagnostic.

KEY WORDS: Simulation, Multilevel system, Maintenance, Diagnostics, Effectiveness

PROBLEM

Production part of real business system (RBS) consists of four organizational and technology units (OTU), each of them differently located. Distance between OTUs is approximately 20 to 40 km. Each OTU consists of four subunits. Working conditions can be classified as normal and campaign. A large number of different technical items (TI) are used in production process. Two kinds of technically very complexes TIs (machines) could be marked as crucial, because of direct influence on reaching the business goal for the system and influence on other TIs working process. Each OTU has about 160 TIs of that kind. Income of RBS is directly proportional to a number of working TIs per day. That is the reason why average number of working crucial TIs per day (ANWTI) is chosen as a basic RBS effects parameter. This approach makes minimizing of down time a crucial goal for real multilevel maintenance system (RMMS) that exists as a supporting organizational and technology part in business system. RMMS (figure 1) is organized as three levels' system according to complexity of maintenance preventive and corrective technological requests' categories, it's frequency and requested technology equipment. There is a fourth level-as well - Central depot level, and some of maintenance requests have to be done there, but it is not included in RBS. Spare parts supply system is hardly connected to RMMS. According to the crucial goal, management of RMMS wanted to investigate possible ways to improve RMMS efficiency and effectiveness with minimum of investment.
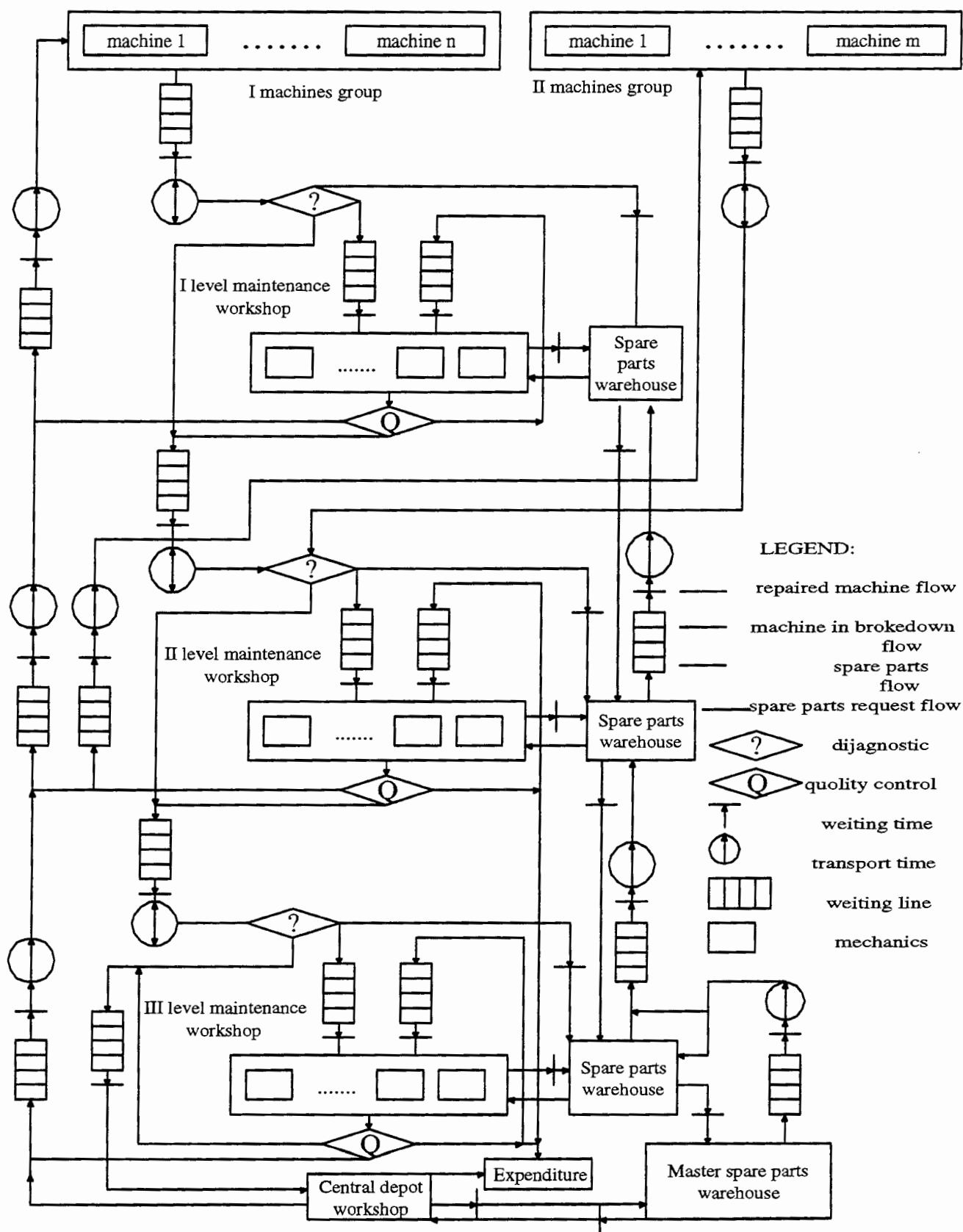
Figure 1. Real multilevel maintenance system model

GOAL

Basic investigation goal was to determinate all influencing factors on processes in RMMS in function of cost/benefit analyze of investment in organizational changes, equipment and staff training for efficiency and effectiveness improvement. There were expectations that diagnostics would be the main influencing factor and that is the reason that first part of investigation was made in function of cost/benefit analyze of investment in equipment and staff training for diagnostics. That part of investigation is shown in this paper.

METHODOLOGY

The reason for Field studies providing in RMMS was to determinate parameters and to investigate character of the processes. Complexity, stohastics and unstationary process character (two types of working conditions) in RMMS and large number of influencing factors made modeling and simulation a basic and only possible investigation methodology tool. As it would be seen RMMS can be treated as Markov process simulation system.

MODELING

A RMMS model was created and it is shown on figure 1. As it can be seen from the figure, there are few different flows: machines in breakdown, repaired machines, spare parts' requests and deliveries. Breakdowns appeared in a stohastics way and preventive maintenance actions are strictly determinate in time. Information flow about machines is following material flow. There is a number of different waiting times, such as transportation, spare part's delivery, diagnostics, quality control, time to repair, etc. They are different for RMMS levels, types of TIs and maintenance actions (every has specific distribution). TIs and spare parts before actions of diagnostics, repairing and delivery makes waiting lines of specific requests. One level of RMMS can provide only maintenance actions for which it has an authorization. As well, type of requested maintenance action is determinate by providing diagnostic procedure. Technological processes for every possible maintenance action were also included in model as specific roots of TIs and parts of TIs between mechanics' working posts, but they were explained elsewhere [1]. Maximum theoretically possible amount of breakdowns and preventive actions and its complexity is taken as a first step point of the investigation. Each RMMS level can be separately investigated as well as RMMS as a complete system.

RESULTS AND CONCLUSION

Diagnostic accuracy (DA) nowadays in RMMS is about 90% at all levels. First step in experiment with diagnostic influence was determination of ANWTI in case of 90% accuracy. Second step was determination of ANWTI in case of DA increasing in 2% steps up to theoretically possible 100%. Results showed that DA for minimum step of change maximally influenced ANWTI among RMMS influenced factors. In the case of DA rising up for 10% RMMS can expect rising of ANWTI up for about 15% (figure 2). There is a slow increasing of repairing possibility among all RMMS levels, but there is no changes in utilization coefficient of employed mechanics (figure 3). DA does not also have significant influence on other needed maintenance resources in RMMS : e.g. number of mechanics on III level workshop (figure 2). Presented results leaded to logical conclusion that diagnostic procedures with higher accuracy would decrease number of TIs and spare parts in flows between different levels of RMMS. That would obviously decrease TIs downtime and rise up ANWTI. There are few possible technological and organizational ways to increase DA. Simple cost/benefit analyze lead to conclusion that all possible solutions that guarantee DA rising up for 6% will definitely approve all investments in equipment and staff training for diagnostics.
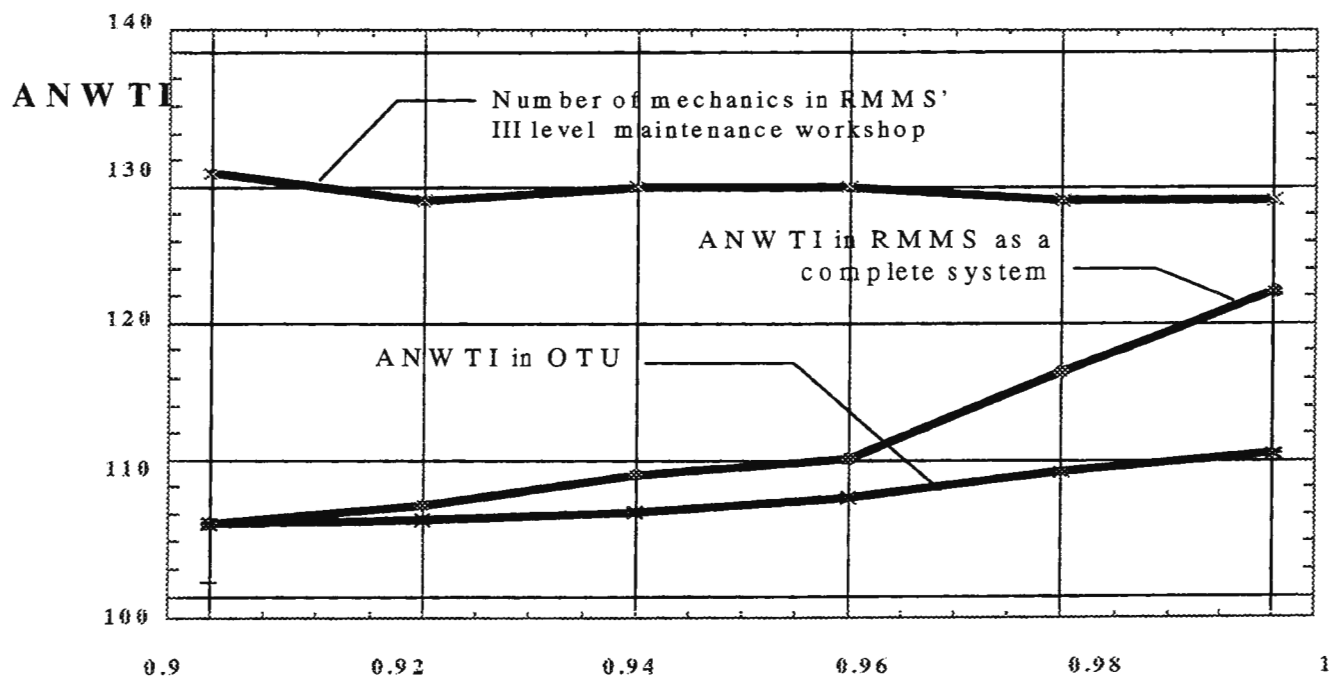
Figure 2. - Probability of accurate diagnostics
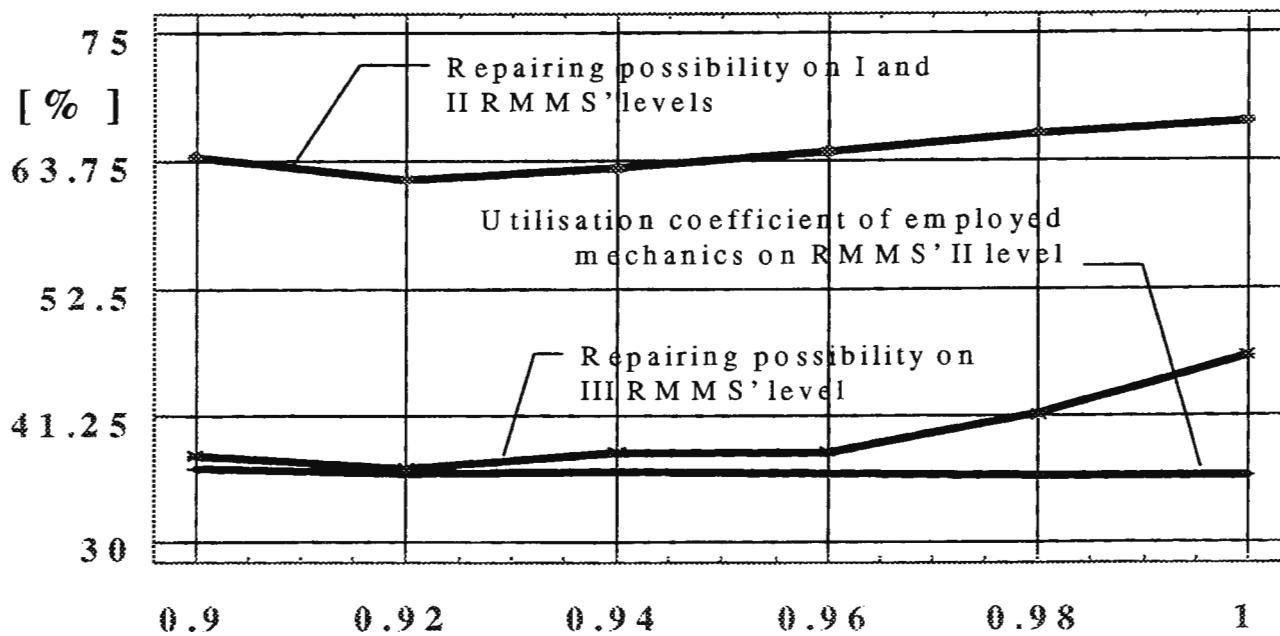
Figure 3. - Probability of accurate diagnostics

LITERATURE:

1. P. Stanojevic: INFLUENCES OF TECHNICAL FACTORS ON MAINTENANCE SYSTEMS ORGANISATIONAL STRUCTURE, doctoral dissertation, Mechanical faculty, Belgrade, 1997.

HYPERMEDIA AND SIMULATION IN PROCESS CONTROL SYSTEMS

Jaakko Oksanen[1]

[1]Valmet Automation Inc., Control Systems, P.O.Box 237, FIN-33101 Tampere, Jaakko.Oksanen@valmet.com

1. ABSTRACT

In this paper, the advantage of hypermedia and simulation systems in the process and process control industry are presented in brief. Practical examples and cases present the current use of these techniques used in today's industry.

2. NEED FOR INFORMATION

Operators in charge of process controls are overwhelmed by the increasing amount of information that must be learned to operate modern factory equipment. For the sake of productivity and efficiency, the operator is responsible for larger and larger process areas, and often alone. At the same time, processes have become increasingly complex due to higher quality requirements and goals. To be able to control these processes, automation systems must conform to more demanding requirements. The degree of automation has become higher and the sophisticated control strategies demand new skill requirements of the operators. The automation system itself is capable of maintaining the process during normal operation, but in more demanding disturbance situations it is the operator who must maintain control and assume the responsibility of restoring the normal operating conditions at the plant. When this occurs, the operator often needs to work with loops that are rarely used. Any earlier training or experience he may have had with these loops is often forgotten during the time and lack of practice. The same is true when starting up or shutting down the operation of an entire plant, or in connection with grade changes that are, in regards to continuous processes, considered as disturbance situations.

When controlling the process, the operator has to deal with large amounts of information. This may sometimes lead to a potentially dangerous situation: a crucial piece of information needed to recover from a disturbance may disappear in the multitude of general data in situations requiring quick action. Thus, the correct information must be readily available where it is needed, especially in disturbance situations where the time available for information browsing may be very limited. Users have the tendency to resort to a trial-and-error method in situations where they do not find sufficient help. In these situations, even though the users' experience and knowledge can often lead them to the correct solutions, the results can sometimes be far from optimum. The average user seek help in documentation only after he has failed to find the solution himself, and at that point, a lot of valuable time may already have been wasted.

The roles of the staff have been changed dramatically over the last few years. Tasks in the mill have become more versatile and there are no longer rigid roles for the staff. The supporting systems (e.g. the control systems) should thus help users by informing

them. The multifaceted information needs lead to severe demands placed on the techniques used in the user interface.

Simulators and simulation techniques help operators in their work. Using these techniques, results of operations can be verified before their actual performance in the mill. Also during the training period before mill start-up, "feedback" on the process can be obtained.

A great deal of paper documentation created to assist the process operator exists, but there are various reasons for not recommending printed material for this particular purpose. Paper documents are difficult and expensive to update, the users are reluctant to resort to paper documents, the documents are not easily found when the user needs them the most, and often, the documentation structure is not the best for the specific reference purpose. Another problem is that paper documentation only supports the linear approach to reading, thereby slowing down the operators as they browse the information. Information searches should be as effortless as any other routine task. The essential information should be located logically so that it can be found in the same place as the corresponding process controls are performed: at the automation system terminal. All information provided must be up-to-date. Thus the policy of changes must be determined.

Hypermedia offers a natural way for the user to navigate in the middle of information. When desired, the information can be read in the traditional linear manner, or the user can utilize the intuitive hypertext links between the information elements. Issues and details that belong to the same entity or concept can thus be interlinked without distorting the overall information structure. Hypertext documentation also supports the direct retrieval of details concerning a certain issue. All together, this technology offers countless views to the information mass which makes the information suitable for many different user groups.

The latest development in information technology enables the implementation of most documentation functions serving the operator on a large industrial scale. The achievements of Internet and WWW technology have opened the door for access to a wide range of information.

3. TOOLS USED IN TODAY'S AUTOMATION

The operator needs various sorts of interactive data to support run-time and problem solving tasks. Sometimes, the information contained in graphical material (e.g. system charts, diagrams, etc.) is as invaluable as the hypertext aids. Thus, an ideal operator support system is a combination of the various information presentation methods.

The following examples describe how hyper/multimedia, internet, and simulator techniques are already used in process control systems.

3.1 Hypermedia and Internet solutions

3.1.1 On-line help for process operators

On-line documentation is designed to serve the plant maintenance and operating personnel, as well as the plant design process. Documentation consists of functional descriptions and other process related information. Descriptions record external functions included in the automation system, such as with some process-related safety regulations. The WWW/Intranet technology used in the Damatic XD*i* documentation system, offers the latest documentation tools and methods available in the industry.
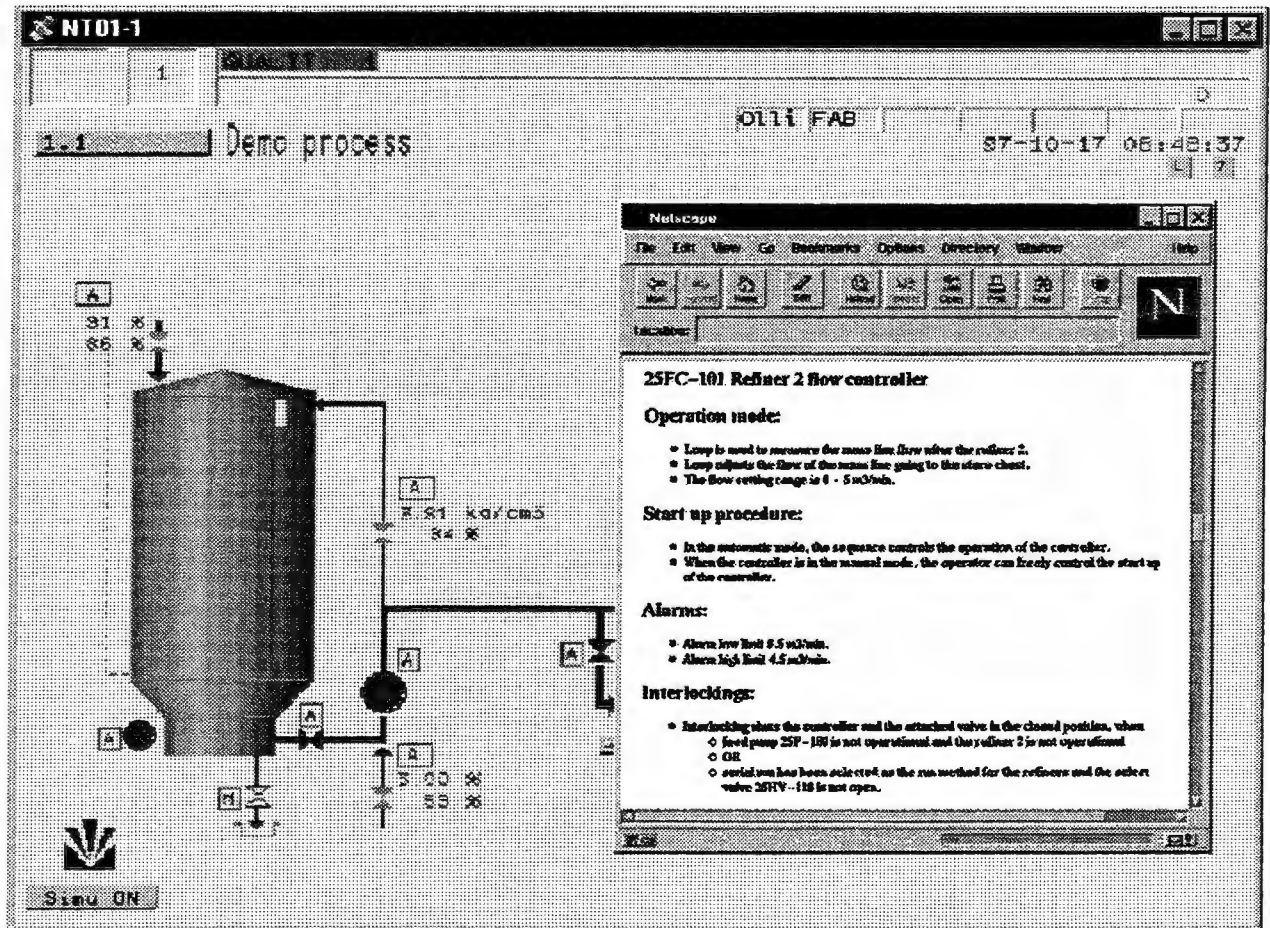
Figure 1. User interface of the Damatic XD*i* system

The user finds the loop-specific documentation easily, just by pointing the loop (e.g., a pump) and pressing a button on the terminal. This will provide the user with focused information on the specific subject. The documentation can also be browsed outside the automation system itself, so that it is, for example, accessible in the offices of the organization, just like the documentation of other departments within the organization.

The documentation is designed to assist the operating personnel in process start-up and operation routines. The complexity of an automation system, with all its controls, settings, and interlocking procedures, forms an invisible layer between the system user and the process itself. It is, however, crucial that the automation system user will, especially in disturbance situations, get quick and clear information concerning the

actual functions of the system. Process disturbances are the most important tests of the usefulness of the documentation system. The information required for process recovery must be easy to access and ready to use. There is no time for browsing through the entire documentation, therefore, the only way to provide focused help in problem situations is to link the documentation to the display elements of the automation system.

From one specific mill The operators report on daily usage of documentation, and the statistics (see figure 2) will also show the use frequency of the documentation. During some weeks, there have been up to 600 searches per week, with the bleaching and brown pulp departments being the most active search service users. The operators used the system most often at the plant start-up phase. This can be explained by the fact that, in that period, the system, equipment, and machinery were new to the users, thus causing the need for assistance.
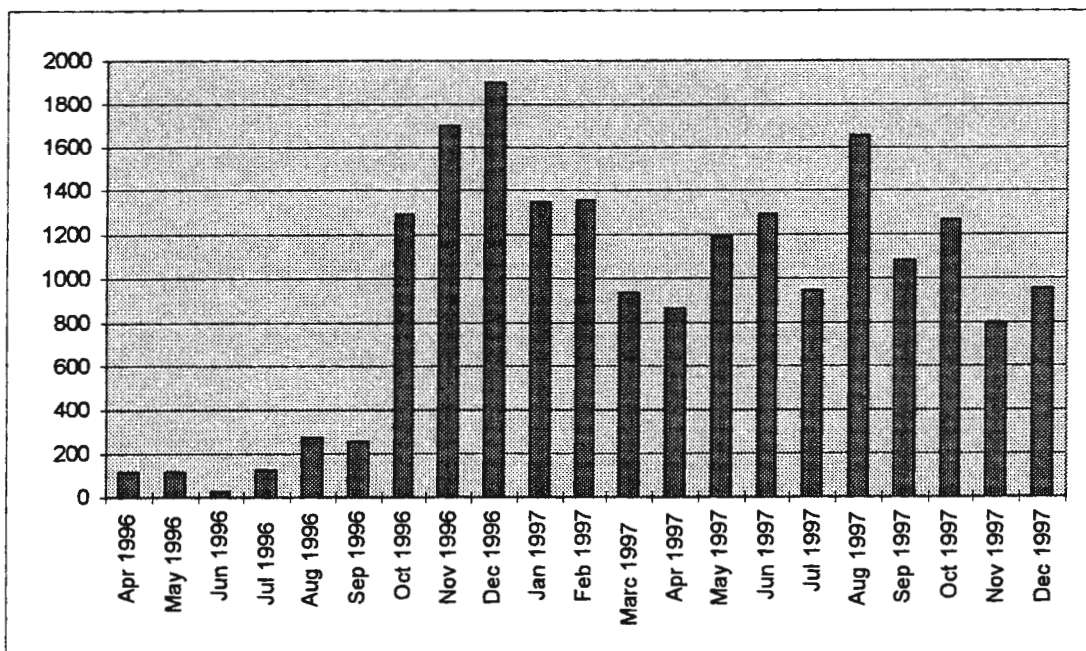
Figure 2. Usage statistics of the web helps

The start-up phase of the pulp processes lasted from October to mid-December. The above chart illustrates the use of documentation at the start-up phases. During that period, the users were only just learning the process and the system functions were still unstable. When the processes stabilised, the need to seek information from the documentation was decreased. It must be noted that even during a normal process run, the documentation was in daily use. The chart also shows how the operational stoppage in May increased the documentation usage, even though the actual stoppage days the use of documents stopped.

3.1.2 Hypertext documentation

Manuals@Web is the full package of Damatic XDi and PaperIQ documentation in a HTML format. Manuals can be located in the customer's server or they can be located in Valmet Automation's server. The manuals can be accessed through an Internet connection. A password is also provided.

In local server, you need 300 Mb of free disk space, if manuals are ordered as a DAT tape. The other alternative is CD-ROM.

3.1.3 Cyber user interface access

Cyber user interface access means that a service person can capture the mouse, keyboard, and display of a remote Windows NT PC. This allows him to teleoperate customer control system worksations (e.g. operation station or engineering worksation) and perform service work as if he would be actually working with the PC at the mill. The concept is based on the PcAnywhere program, which is installed in the mill PC and the PC used by the serviceperson . The connection is made via network modem.
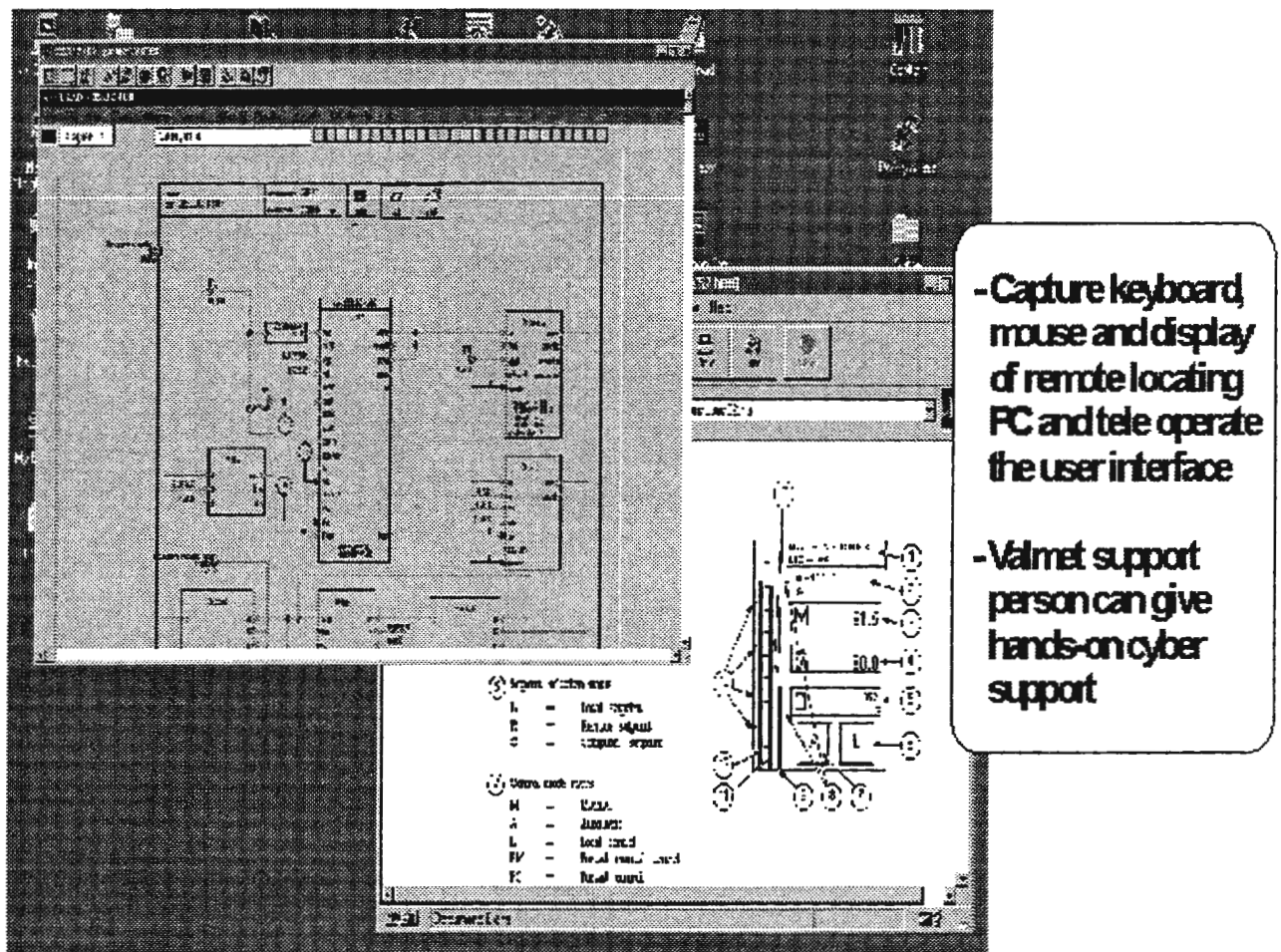
Figure 3. Cyber user interface

3.1.4 Cyber meeting tool

Cyber meeting tool means that a service person and a customer can share a clipboard, called a white board. They can both see all the pictures they copy and paste to the white board and can also write texts on the white board.

In a telephone support situation, a serviceperson can provide better service to the customer, because he can understand customer's problem more easily and give visual advice with the white board. Sometimes one picture communicates more than a thousand words.
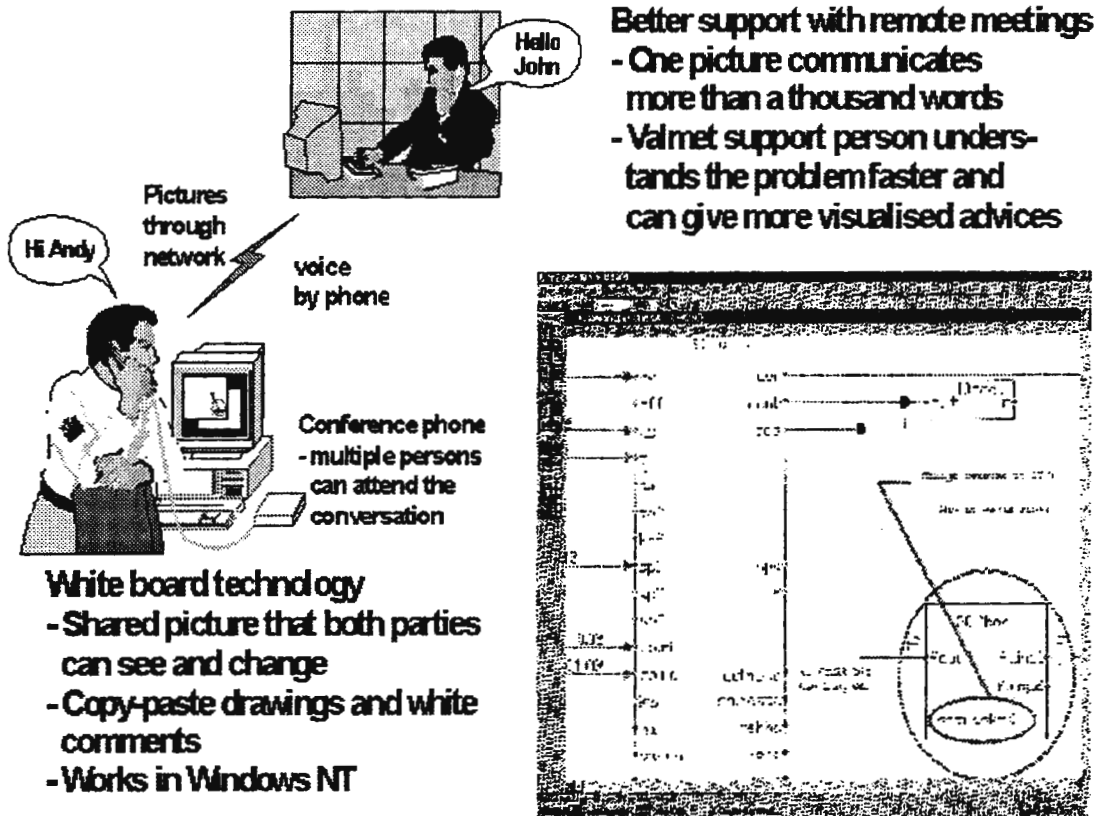
Figure 4. Cyber meeting tool

3.1.5 Technical Notes at WWW

Sometimes technical manuals will may not provide the latest information and experience available in the field. Consequently, an up-to-date database has been established. Technical Notes provides up-to-date product information for customers through the Internet. In Technical Notes, there are short user guides, hints&tips, system faults and how to solve them, as well as examples for more efficiently using one's systems. Information on new updates will also be available. System specialists at the Valmet Automation excellence centres will maintain this database.

3.2 Multimedia tools

3.2.1 Virtual paper machine

The purpose of the paper machine simulator is to simulate the future paper machine and control room. The machine and it's automation is presented from the most challenging and thus interesting points.

The paper machine simulator is located in the Heureka Science Centre in Vantaa, Finland. The "operators" of the paper machine (and the users of the automation as well) are visitors to the science centre -- a normal person who may not know anything of papermaking. The visitor can "run" the machine and/or study some parts of it in greater detail. The virtual machine allows, using one of its three wide screens, a detailed view of some particular part of the machine while the other screens show an overview of the whole machine. The user interface for the application is the normal Damatic XD*i* user interface.

The virtual paper machine is also presented in paper making conferences where there is a great deal of interest in the machine as well as the automation.
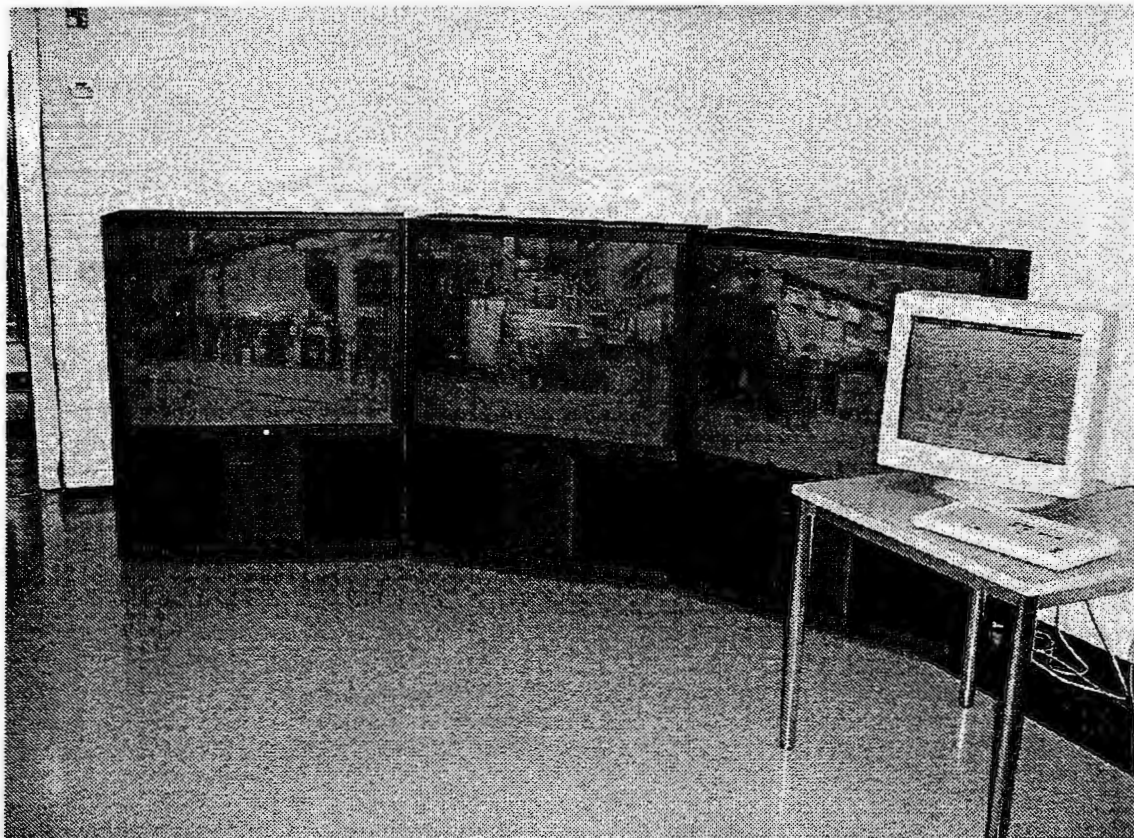
Figure 5. The virtual paper machine

3.2.2 Multimedia training systems

Paper IQ is an advanced paper quality control system developed by Valmet Automation, Inc. The Paper IQ product family consists of a large variety of controls, optimisation packages, sensors etc. for paper producers.

The PaperIQ Multimedia Training offers details of mechanical maintenance tasks and procedures for PaperIQ equipment. Practical exercises give the multimedia user insight into fault correction. Our virtual trainer Matti S. guides the multimedia user through the training sessions. Matti is a professional teacher; if needed he can continue teaching as long as wanted, repeating the difficult parts as many times as needed.

The multimedia training application helps the user to gain a better understanding of the hardware maintenance principles and makes it easier to carry out maintenance tasks on-site. VIVA PaperIQ Mechanical Maintenance multimedia presents the main parts and functions of the PaperIQ frame and sensors with 3D graphics, photos, video clips, animations and spoken text. The entire PaperIQ documentation can be viewed in an HTML format using Netscape. There are also VIVA applications for other process areas such as OptiReel roll handling and OptiLoad calander.

Mechanical maintenance operations for the frame and sensors are shown with three types of training sequences, narrated by the Valmet virtual trainer, Matti.

The multimedia manuals are in English and Finnish, but other languages are also provided. The target users of the multimedia systems are our customers as well as Valmet's own staff.

The pulp mill multimedia represents the products, applications and examples of the specific process phase. User can also simulate the effects of changing various process variables.

There are a number of multimedia tools presenting the whole Damatic XDi system. The tool presents the automation language, the system structure, as well as how to configure a solution with it. The tool is linked to Damatic XDi electronic documentation - Manuals@Web. The tools also contain some exercises to test the users.

All applications are made by the VIVA (Valmet Interactive Virtual Application) tool. The development of the VIVA tool started already in the 1993, since which time Valmet has had multimedia tools for training and simulation for different phases and elements of processes, systems, applications, and project executions.

The multimedia manuals are distributed as CD-ROMs. The other electronic manuals can be distributed as CD-ROMs, data tapes, or from our web site.

Figure 6. Pulp mill multimedia

3.3 Simulators

3.3.1 Damatic XDi simulators

In Damatic XDi systems, several different types of simulators have been implemented. The simulators operate in the fields of energy, pulp and paper. The simulators are made by applying normal functions of the control systems, e.g., controllers, sequences, and logic function blocks.

The following examples gives an overview of the XDi simulators. The simulators have been used for training and simulating the process and control system as well as testing new, enhanced control methods.

In the energy process industry, there are simulators for several types of boilers as well as to the other parts of the plant, e.g. steam networks. In one example, the process is a pulverised coal-firing power plant, which produces electricity and district heat. The process can be divided into the following parts which all are simulated in the system:

- fuel handling and fuel feed (pulverized coal, oil)
- burning air and flue gas system (2 ID fans, 2 FD fans)
- feedwater and steam system (drum boiler, two feed pumps, 3 superheaters)

- turbine (turbine by-pass valve)
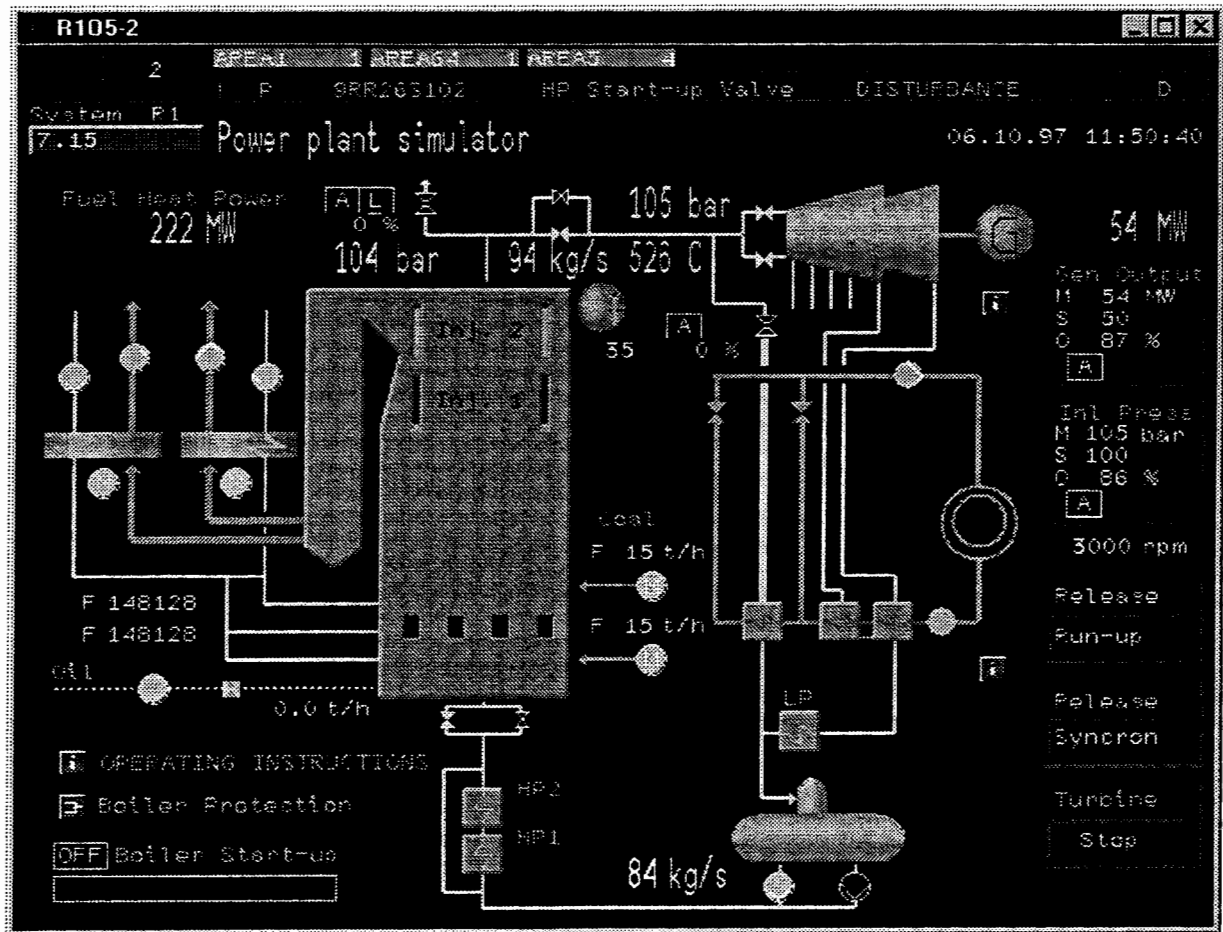
Figure 7. Example of the energy simulator.

In the pulp and paper process industry, there are a large number of simulators covering the whole production line: from the pulp mill's chip feed to the end of the paper machine. The simulators consist also alternative techniques, e.g. both batch and continuos digersters are covered.

The contents of the batch cooking simulator (SemiSuperBatch process) might consist of the following sequences:

- Chip feed
- Warm Impregnation Liquor Filling (WIL-filling)
- Hot Impregnation Liquor Filling (HIL-filling)
- First White Liquor Charge (FWL-charge)
- Heating
- Cooking

- Displacement
- Discharge
- Digester Emptying

4. CONCLUSIONS

The abovementioned examples provide a short overview of the current status of today's process industry. The use of applications mentioned will certainly expand in the near future. The progress of the techniques will give us efficient tools to implement applications.

The Damatic XDi control system offers today a better platform for simulators and documentation: Java language. The Java Virtual Machine has been embedded in process stations thus offering a seamless communication between the simulator and other control systems. This allows for more sophisticated machine independent simulators to be developed for the process industry. Furthermore, XML techniques will initiate a similar evolution of documentation that HTML did few year's ago. The combination of Java and XML will provide us with a new and open platform for information used in tha plant.

5. REFERENCES

Bainbridge, L., Ironies of Automation, Automatica, Vol. 19, No. 6.

Kaarela K., Oksanen J., Takalo J. "An Information Model as a Basis for Hypermedia-Based Plant Documentation", Proceedings of the Third International World-Wide Web Conference, April 10 - 14, 1995, Darmstadt, Germany, In: Computer Networks and ISDN Systems, Vol 27, No 6 (April 1995), pp. 751 - 764.

Kerkelä K., Oksanen J., Paunonen H., Skyttä T., Huhtanen M.: "WWW/Intranet pohjainen sellutehtaan käytön tukijärjestelmä" ("WWW/Intranet based pulp mill operation support system"), Proceedings of Automation 97 - days, September 23-25, 1997, Helsinki, Finland. CD-ROM full paper (ISBN 952-5183-07-6), abstract p. 55 (ISBN 952-5183-06-8).

Paunonen, H.: Roles of informating process control systems. PhD thesis. Tampere University of Technology, Publications 225, Tampere, 1998.

Valmet Automation, http://www.valmet.com/automation/

CYBERBIKE

Virtual Reality and Multimedia in Road Safety Education

D. Wloka[1], M.Zolg[2], C. Winkelholz[3]

[1] University Kassel, FB 16, FG Technische Informatik, D. Wloka, wloka@inf.e-technik.uni-kassel.de
[2] University Kassel, FB 02, FG Technische Elementarbildung, M. Zolg, zolg@hrz.uni-kassel.de
[3] University Kassel, FB 16, FG Technische Informatik, C. Winkelholz, winkelholz@inf.e-technik.uni-kassel.de

Abstract

The progress in the development of new media, like computer graphics, multimedia and Virtual Reality recommends to reflect upon the idea if traditional pedagogical concepts could be enhanced using these technologies.

The project "Cyberbike" tries to develop a learning environment for children in which they can learn how to drive a bicycle in todays traffic situations.

Pedagogical and technical members work together to create the content and to build the software components, using multimedia and virtual reality techniques to support the learning process.

Introduction

48209 children had an accident while participating in road traffic during 1996 in Germany. For 1997 it is estimated that numbers had not changed very much. Many accidents are caused by wrong behaviour of children. Today's traffic volume is so high and situations are so complex that small mistakes very often lead to severe accidents. Children using bicycles are always the weakest part.

In most European countries children are trained in school to participate in road traffic. The high numbers of accidents indicates, that traditional forms of training are no longer suitable to decrease accidents substantially. Additional forms of training have to be found.

Software based education and training systems using new media may be a promising way for the following reasons:

1. Children from 8 – 12 years are learning by doing and repetition. Software based training environments can provide situations never possible in reality. Additional multiple channels to provide information (visual, acoustic, haptic) are possible. It is clear however, that only appropriate content using the technology will cause an effect.

2. Training is especially necessary for critical and dangerous situations, without harming the child. This is not possible with conventional training schemes. It can be done using VR-based scenarios and VR-based training environments.

3. Virtual Reality training can also be used to learn basic qualifications, like estimating the velocity of approaching cars or detecting the direction of movement.

4. The training progress can be measured in such an environment, thus training schemes can be evaluated very effectively.

Problems of traditional road safety education

Road safety education attempts to decrease the number of children been involved in an accident for a long time. In Germany the current concept for the preparation of children to participate in road traffic consists of traditional classroom based learning in cooperation with the police, one exam to acquire a "cycling-licence" and a test-tour in real traffic.

Situations which are typical for accidents can't be tested in this tour in real traffic, because risk is to high. Therefore it is difficult to decide, if the child is able to transfer his theoretical knowledge learned in lessons to behaviour in real situations.

Solving Problems by Multimedia and VR

The learning evironment "cyberbike" consists of a ROad Safety Educational MultiMedia Software (ROSE-MMS) and a ROSE - Virtual Reality Environment (ROSE-VRE).

The ROSE-MMS will teach children, the ROSE-VRE will train children. In this way the "cyberbike"-environment is able to tie together important components of road safety education and helps to judge the qualification of a child to participate in real road traffic.

Pedagogical Concept

Traffic is a complicated occurrence with different and intense information, which must be assimilated very quickly. Roads are build for adults and not for children. Children have different visual, acoustic, emotional, intellectual and kinetic abilities which mostly are neglected.
For example: The field of view of a child is smaller than that of an adult; the perspective perception isn't fully developed.

In a learning environment these differences can be respected and appropriate learning and training methods can be developed to overcome the problems.

No child likes to learn. If one uses new media the child will compare the environment to computer games, which provide action, trilling graphic and most no valuable content.

To overcome this problem a learning concept has been developed which integrates the learing units

Fig. 1: Start-Page of the ROSE-MMS

310

in a story. Four children of different social background are involved in stories. Fig. 1 shows the starting page of the learing software.

The learning child has thus the chance to identify itself with one child of the group and can follow the actions. The child can learn what mistakes could be done and how to behave correctly. The stories take place in a virtual town, which again provide todays different living areas and the typical traffic conditions.

The ROSE-MMS does include lessons about appropriate behaviour in traffic and meaning of traffic signs. Dangerous situations are integrated which are typical to cause accidents in real traffic, like left turn, get in lane, etc.

The ROSE-VRE will provide the virtual town to train the child. Here the child can move through

the roads and dangerous situations will be generated. The ROSE-VRE is the missing element in
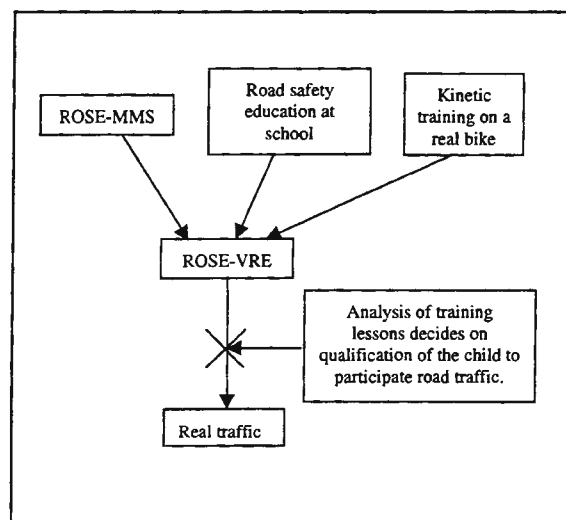
Fig. 2: Integration of the "cyberbike" into traditional road safety education.

road safety education, which allows the child to transfer knowledge learned in theoretical exercises to behaviour which is necessary in real traffic. Fig.2 shows how the "cyberbike" may be integrated into traditional road safety education.

Further by using a ROSE-VRE for training, it is possible to record training lessons, to analyse them afterwards. In this way individual training for single children is possible.

Technical Concept

The ROSE-MM environment will be created using multimedia developing tools. It will integrate text, pictures and 3D-animations.

The ROSE-VRE consists of a real bicycle. The bicycle is equiped with a collection of sensors to provide the input signals. The virtual town is projected on screens, as shown in Fig.4.

Fig. 3: House in the virtual town

Fig. 3 shows as an example a house in the virtual town.

Fig. 4 Concept of a projection based virtual training environment.

The development of this virtual simulation environment is a complex task. A decision to use a client/server concept, object oriented programming techniques and CORBA as a distribution medium has been made. Currently a prototyp is under construction. Fig. 5 shows the system arcitecture.
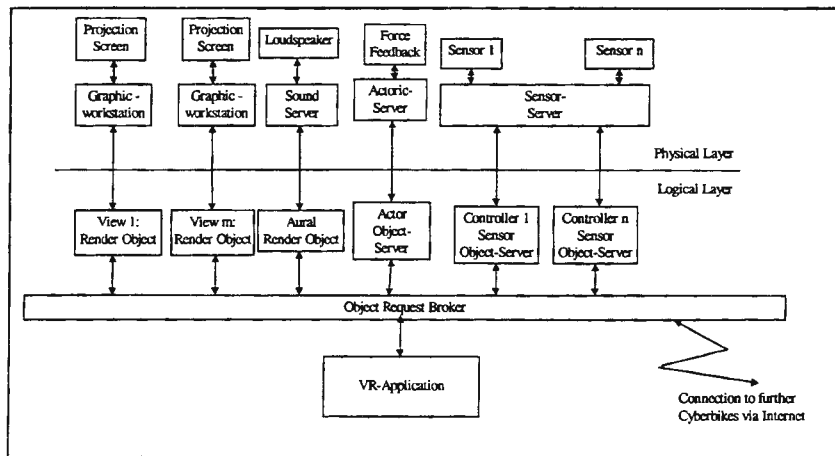
Fig. 5 : Soft-, Hardware architecture of the ROSE-VRE

Conclusion

The combination of new learning schemes (content) combined with new technologies provides a promising approach to educate children. The group hopes, that application of this system in schools will substantially decrease accidents.

References:

VdTÜV 1997 Verband der Technischen Überwachungsvereine (VdTÜV)(1997), Bonn

Statistisches Bundesamt 1996 Statistisches Bundesamt (Ed.) (1996): *Kinderunfälle im Straßenverkehr*, Wiesbaden, 10

Warwitz 1994 Warwitz, S. (1994): *Verkehrserziehung vom Kinde aus* Hohengehren, 32-44

Spitta 1997 Spitta, P. (1997) *"Kinder im Verkehr" Neue Konzepte der Verkehrserziehung in der Primarstufe* VCD (Ed.), Bonn, 60-64

Warwitz 1994 Warwitz, S. (1994): *Verkehrserziehung vom Kinde aus* Hohengehren, 32-44

Spitta 1997 Spitta, P. (1997) *"Kinder im Verkehr" Neue Konzepte der Verkehrserziehung in der Primarstufe* VCD (Ed.), Bonn, 79-84

COMBINED HYPERMEDIA AND SIMULATION IN LEARNING OF PROCESS AUTOMATION

J. Lindfors, L. Yliniemi, K. Leiviskä[1]

[1]University of Oulu, Infotech Oulu and Department of Process Engineering, PO Box 444, FIN-90571 Oulu, Finland,
juha.lindfors@oulu.fi

Abstract

The fast development in the field of process automation together with reduction in teaching resources and increasing number of students forces educational establishments to consider more effective learning methods. The recent development in software technology enables utilization of new technology in education.

This paper describes the utilization of hypermedia technology, combined with simulation, in learning of process automation in the Control Engineering Laboratory at the University of Oulu. The hypermedia material was initially obtained in a COMETT II-project. The project was concerned with the building of single-platform training materials for process automation that will be used both in university courses and in continuing education of people from industry. After completion of the project, the material was included a special module for rotary dryer. This module is the base for the Matlab simulation of the dryer.

Introduction

Process automation has been one of the fastest developing fields of process industry in recent years. This makes for quite high demands on automation personnel. They should be able to learn new technology fast and efficiently. This means that the training of automation must be also interesting and stimulating.The recent developments in the fields of hypermedia, computer, and network technology make it possible to utilize computer based learning to meet these new demands.

In the Control Engineering Laboratory of the University of Oulu, this new situation was encountered in late 80's. It resulted in the application to the COMETT II program, and after positive decision a COMETT II-project "Training material for process automation" with three universities and four industrial partners was carried out during 1990-93.

The overall objective of the project was to develop training material for process automation that can be used in different areas of industry and in different technical institutes both as support material for teaching and for self-learning [1]. The material, consisted of the computer textbooks with hypertext, animations, simulations, sound, and videos, was built using hierarchical multimedia technology. As the result of the COMETT-project seven bookshelves with more than 110 hyperbooks from different fields of process automation were made [2]. Some new hyperbooks from new topics of automation as fuzzy logic and neural networks have been added to the materials after the project. The material runs in a standard PC-environment under Windows. The material can be browsed from WWW using any browser, if a special plug-in, provided for that purpose, is added on the browser.

Hypermedia material

The training material package developed in the COMETT II project, was modularly grouped into series of courses that could be used at different levels of education and in different fields of industry. The basic idea behind the system was that it would be possible to pick a tailor-made module with main "textbook"-part, examples, problems, demonstrations, and audiovisual material for some main topics of process automation. The main modules were as follows:

1. Process measurements and signal conditioning
2. Process dynamics and modeling
3. Controller design and tuning
4. Advanced control methods and optimization.
5. Industrial example: simulation of freeness control (pulp and paper industry)
6. Industrial example: simulation of a PVC-process (chemical industry)
7. Industrial example: simulation of plate rolling mill (steel industry)

The rotary dryer module

The material obtained in COMETT -project includes also a module for learning the basic theory of modeling and control of a rotary dryer.

The behavior of the rotary dryer is simulated in Matlab (Simulink). Both openloop and closed-loop simulations are possible. Openloop simulation is studied during a course of experiment design and closed-loop simulation is used during studies of different control.

The simulation is started from the hypertext site, however, the main user interfaces for simulations are used from Matlab. Using these interfaces, parameters of the process and control can be changed. Figures 1 and 2 show contents of the hypertext interface of the rotary dryer module. Figure 3 shows a screenshot of a simulation.
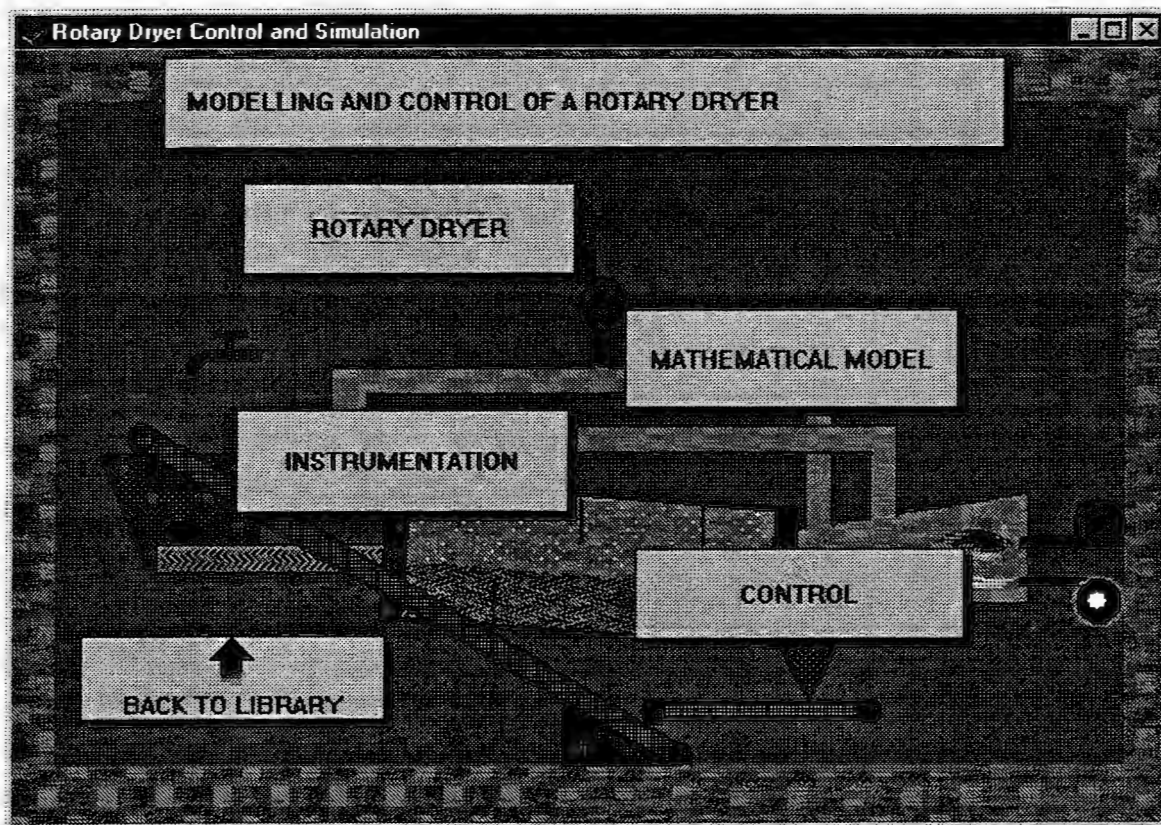
Figure 1. First level contents of the rotary dryer module.
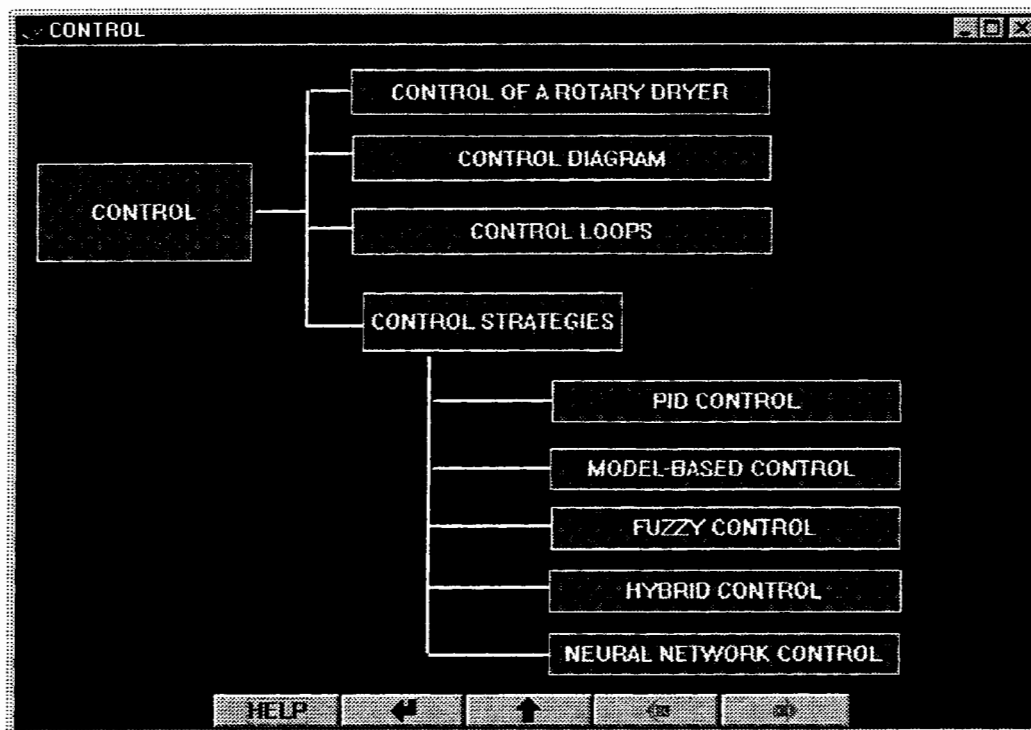
Figure 2. Hypertext contents of controls of the rotary dryer module.
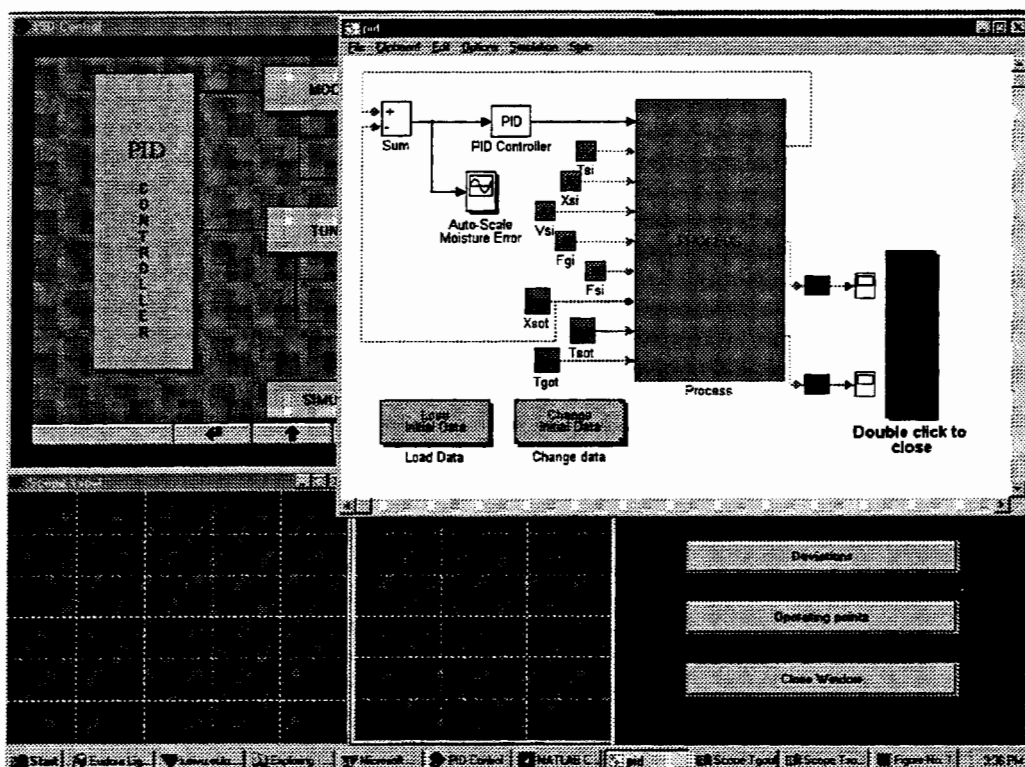
Figure 3. Screenshot from a simulation interface.

Later the rotary dryer module was filled in a book concerning about neural networks and genetic algorithms. Also some additional examples have been included. Asymetrix Multimedia Toolbook 3.0 and its OpenScript language were used for hypertext, simple animations, and simulations. Table 1 summarizes the content of the

rotary dryer module and the table 2 gives a closer overview for the contents of the new fuzzy, neural, and genetic algorithm module.

Table 1: Summary of training material of the theory of the rotary dryer [3].

Books	Includes	Pages
Start	Contents	2
Rotary dryer	Theory	8
	Measurements	10
	Control	9
	Mathematical model	11
	Fuzzy control	5
	Simulation	4

Table 2: Summary of training material of fuzzy control [4].

Books	Includes	Pages
Start	Contents	2
Fuzzy Control	Introduction	41
	The Mathematics of Fuzzy Control	
	Fuzzy Controller	
	Expert Systems and Fuzzy Control	
	Introduction to Adaptive Fuzzy Control	
	A Guide to Fuzzy Application	
	Example	
	Bibliography	
Neural Networks	Introduction	39
	Types of Neural Networks	
	Bibliography	
Genetic Algorithms	Introduction	13
	A Top-Level View of the Genetic Algorithm	
	The Anatomy of a Genetic Algorithm	
	Concepts	
	Bibliography	

A new learning environment

The process itself is a pilot scale dryer in the laboratory hall and it is controlled using a Damatic XD automation system. The automation system is connected to our laboratory network that has also a bridge to the Internet. Because of the connection, it is possible to use also WWW servers for providing information and learning material for operators.

A network server will be used to provide the learning environment. This server includes ToolBook, modified Matlab material for hypermedia use, and a WWW server. Using a workstation, operators studying the control of the dryer, can use hypermedia material from the server to support their studies[5]. The workstation will be a part of a network utilizing the ATM technology [6].

Conclusions

Creating a computer based learning environment in network is challenging in many ways. On the one hand, there is the control engineering approach to the issue; on-line simulation with an automation system, load questions with network, model building and adjustment, and question of tools.

Furthermore, it is not an easy task to integrate different kind networks and programs together, although the operating systems and networks have become more open recently. In addition, there are difficulties in the hypermedia and dynamic linking in order to get the best benefits out of the learning system.

Although the ToolBook material is in principal part ready, the integration of Windows and Unix systems is expected to cause problems. In spite of the ToolBook is running in the local net, launching applications that are pieces of a larger system with system book, etc. can be difficult. The WWW material is more easily available, but can not meet, in this case, all the needs.

References

[1]   Leiviskä K., Yliniemi L., "Training materials for process automation". *Proceedings of the Second Seminar on New Approaches to Computer-Aided learning*, January 9-12, Aalborg, Denmark, pp. 73-77, (1991).

[2]   Lindfors J., "Training Materials for Process Automation in Hypermedia Environment", *Licentiate thesis*, University of Oulu. p. 70, (1993).

[3]   Lindfors, J. (1996) Integration of Hypermedia Material into a Process Automation System. In: *Proceedings of First International ToolBook Virtual Conference '96*, January 18. Internet. 4 p.

[4]   Leiviskä K., (1995) Hypermedia in Simulation and Control. Presentation in SiE/WG Workshop. Brussels, Belgium.

[5]   Leiviskä K., "Hypermedia in Simulation and Control", presentation in SiE/WG Workshop, June 29-30. Brussels, Belgium, (1995).

[6]   Yliniemi L., Lindfors J., Leiviskä K., "Development and experiences on the teaching of process automation via network", *Proceedings of the EAEEIE '96 Conference*, June 12-14, Oulu, Finland, pp. 255-260, (1996).

A MODELLING AND SIMULATION APPROACH TO ERGONOMIC DESIGN OF AUTOMOTIVE COCKPITS

S. Gomes[1], A. Koukam[2], J.C. Sagot[1],

[1]L'Equiped'Ergonomie et de Conception (L'ERCO) Laboratoire d'Ingénierie de Produits (LIP),
[2]Laboratoire de Recherche en Informatique de Sévenans (LaRIS),
Institut Polytechnique de Sévenans, rue du château, 90010 Belfort Cedex, France

Abstract

More and more recognition is being given to the role and importance of modelling and simulation in the design process of a product and its man-machine interactions. Modelling and simulation enable designers to evaluate different design solutions before building the final product. This paper describes a design life cycle that we have adopted in an automotive control project. It also introduces a modelling and simulation phase to evaluate the usability of current functions such as air-conditioning systems or radios. An object-oriented approach is used for modelling and the 3DStudioMAX environment for simulation.

Keywords : Ergonomics, Systemic approach, Object-Oriented Modelling, Simulation, Automotive controls.

Introduction

Recent technological developments in electronics and in-car computerised systems have generated major changes in the "driver-vehicle" interface. Several European research programs have contributed to developing a quantity and a variety of driver assistance equipment such as guidance, navigation and diagnosis systems. The development of such systems in the automotive cockpit introduces the question of road safety repercussions, attention diversions and driver mental cost increase. The purpose of this paper is to study the design of new in-car automotive control systems, based on usability optimisation of current functions such as air-conditioning systems, electrical pane controls, and radios. To do this, we propose a pragmatic methodology which covers part of the design life cycle. The application of this methodology to automotive control design has allowed us to define several innovating prototypes for each function studied. In particular, the modelling and simulation phase has been of great importance. In this paper, we have put the emphasis on this phase of the project. In particular, we show how object-oriented paradigms and statecharts are combined with the 3DStudioMAX environment to model and to simulate many design solutions. This paper is organised as follows. The first part presents the design life cycle used in this project and describes each design phase. In the second part, the modelling and simulation phase is presented and part of the project is used to illustrate the result of this phase. The last section concludes with a brief summary of the main points of this paper and future work.

Design life cycle

In this paragraph we propose a global point of view of an automotive control design project. This project is based on usability analysis of current functions in order to define innovative in-car automotive control systems such as air-conditioning systems, electrical pane controls, and radios. To this end, we have devised a retroactive methodology which covers a part of the product design life cycle from requirement analysis to detailed studies. This methodology is divided into several phases as shown in Figure1. It has been built around a workgroup project involving design, ergonomics, marketing, and engineering fields.
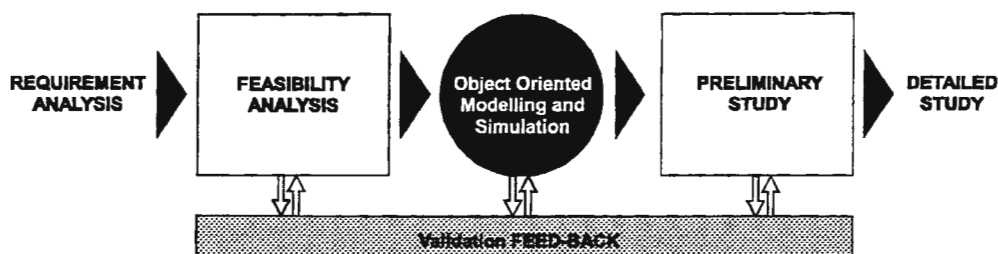
Figure 1 : Design life cycle

After a first phase of requirement analysis introducing drivers' difficulties caused by current man-machine interface complexity (Figure 2), the following phases are then elaborated :

☞ Feasibility analysis which includes several steps : information research on automotive controls, definition of the project's priorities and limits, behavioural analysis in real driving situations, subjective tests on current controls [2]. This phase provides a basis to evaluate the usability of current automotive controls in order to define ergonomic guidelines,

☞ Modelling and Simulation of the Driver-Vehicle-Roadway system (DVR). This phase is based on observed real driving situations, functional and ergonomics specifications drawn previously. Object Modelling Technique (OMT) [6], an object-oriented approach, is used to model the DVR system. The resulting model is then implemented in the 3DStudioMAX environment to evaluate different design solutions.

☞ Preliminary study defines several innovative conceptual solutions [3]. In this phase, a new cooperative and iterative method promoting innovation in three steps is applied : new conceptual solutions research, digital modelling and virtual simulation of the global DVR system including these solutions [5]. These last two steps are applied using 3DStudioMAX virtual reality environment. After several iterations, an optimised solution is selected for detailed study and prototyping phases.

Figure 2 : Example of an automotive cockpit showing the complexity of various in-car systems (from left to right : speedometer, in-car computer, radio and air-conditioning system)

At the end of the feasibility analysis, we encountered several difficulties for designing automotive controls with usability criteria such as manual accessibility, visual localisation, and comprehension. These difficulties were linked to the complexity of modern automobile driving. According to Wierwille [7], the driver provides loop closure within the DVR system, while also performing manual operations and information detection-analysis-implementation for travel. At the same time, the vehicle itself provides comfort and convenience to the driver. The primary driving task involves, in particular :

☞ visual information gathering : looking ahead at the road, glancing in the rear view mirrors, etc.,

☞ manual operations to control the vehicle : hands are used for steering control, changing gears, etc., and feet provide outputs for longitudinal control (acceleration / deceleration).

When a driver performs a secondary task (glancing at instrument or operating controls), that task must be time-shared with the primary driving task presented previously. In order to describe the DVR system and this time-sharing process observed when performing secondary tasks in real driving experimentation (6 drivers, 5

vehicles, 12 in-car current controls, 3 roadway conditions), we applied modelling and simulation. The purpose of this approach is to lay the foundation of an innovation process applied to evaluate new ergonomic solutions during a preliminary study.

Object-oriented modelling of the DVR system

As shown in Figure 1, we have introduced a modelling and simulation phase to analyse different design solutions and to find the most efficient one. Modelling produces an abstract representation of the DVR system that is then used as a basis for simulation. Such representation requires a modelling approach to describe the structural and behavioural aspects of the system.

Practical experience in modelling has taught us that complex systems can be modelled and analysed provided we adhere to some sound principles such as modularity and abstraction [4]. Object-oriented approaches adhere to these principles and provide a better support for building models that are closer to real-world systems. Based on these observations, we have chosen to follow the OMT approach. The structural model is defined as a collection of object classes and their relationships. Figure 3 shows a part of the DVR structural model. Boxes denote object classes and lines drawn between classes represent relationships.
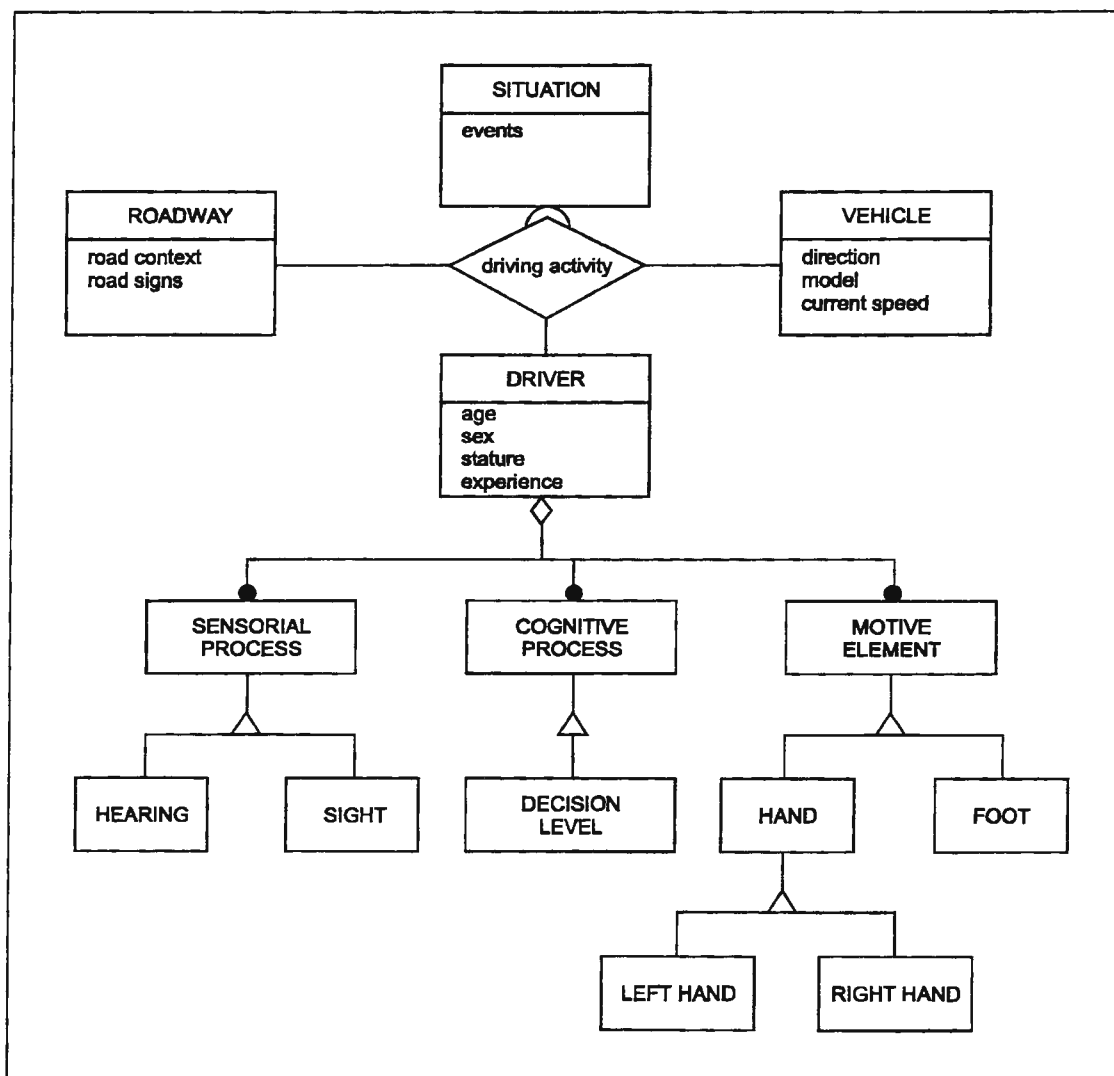
Figure 3 : Structural model example of the Driver-Vehicle-Roadway system using OMT formalisms

The next step consists in building the behavioural model that describes how the object behaves dynamically. This model is achieved through the use of statechart notation; a structured and hierarchical formalism based on finite state automata. Figure 4 shows the dynamic model of the right hand class.
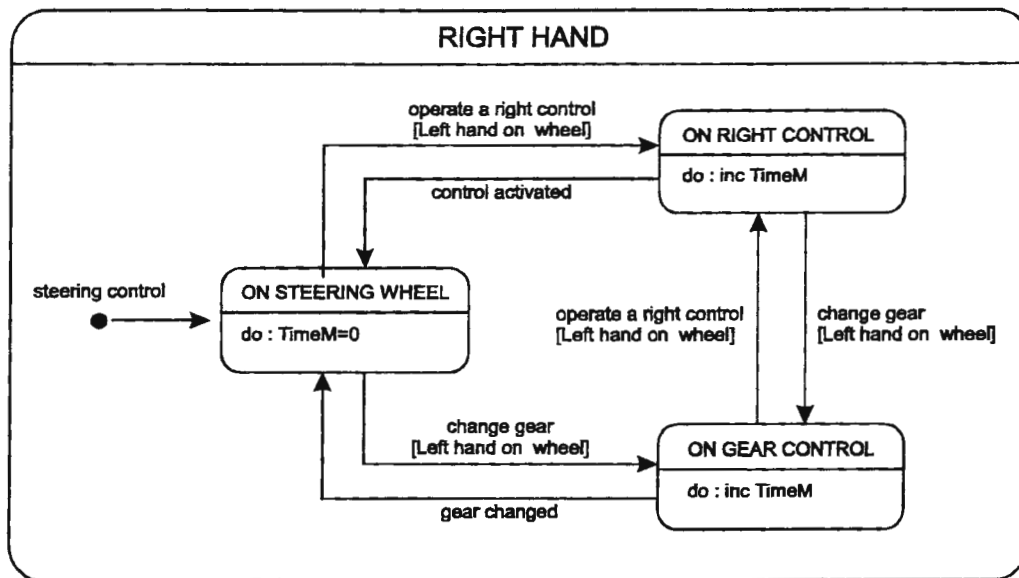
Figure 4 : Statechart presenting the Right hand dynamic model

Transitions between states are denoted by arrows and are caused by events. An expression placed after an event name represents a guard predicate, which is simply a Boolean expression. For example, the transition from the *ON STEERING WHEEL* state to the *ON GEAR CONTROL* state fires when the event *change gear* occurs and the expression *left hand on wheel* is true.

Simulation using the 3DStudioMAX environment

In preliminary studies, we performed a co-operative and iterative innovation process based on digital simulation steps of new automotive control solutions. To do this, we used the 3DStudioMAX virtual reality environment to define a global digital model including an anthropometric mannequin, the future vehicle and a roadway environment. Digital 3DStudioMAX models have been implemented using the OMT models described in the previous paragraph. This allows us to animate an anthropometric mannequin using real observed scenario of secondary visual and manual driving operations in order to define new activities linked to future in-car automotive controls.

To illustrate our design methodology, we can look at the results of our modelling and simulation approach applied to a new in-car radio system design project. As shown in Figure 5, we first defined graphical OMT models of new radio systems. It was then possible to build, on the same defined structure, 3D digital models using the 3DStudioMAX environment (Figure 6). The different 3D radio models are then inserted in a global DVR 3D model to prepare the subsequent simulation steps.
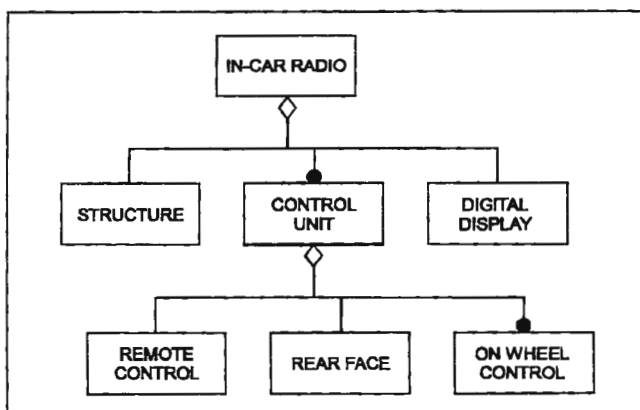
Figure 5 : OMT structural model of an in-car radio solution defined during graphical modelling phase
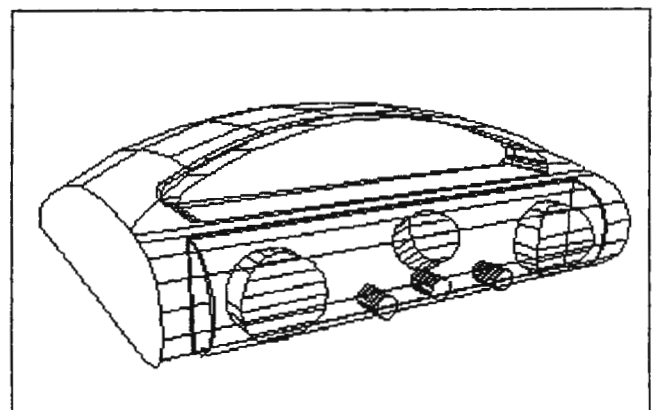
Figure 6 : 3D model example of an in-car radio solution defined during digital modelling phase

Figure 7 shows an example of a scenario used to simulate a secondary radio task. This scenario is built on real observed behaviour situations and analytical predictive relations defined during the feasibility analysis phase. It then becomes possible to compute a virtual graphic animation resulting from the simulation of 3D digital models with different scenarios (Figure 8).
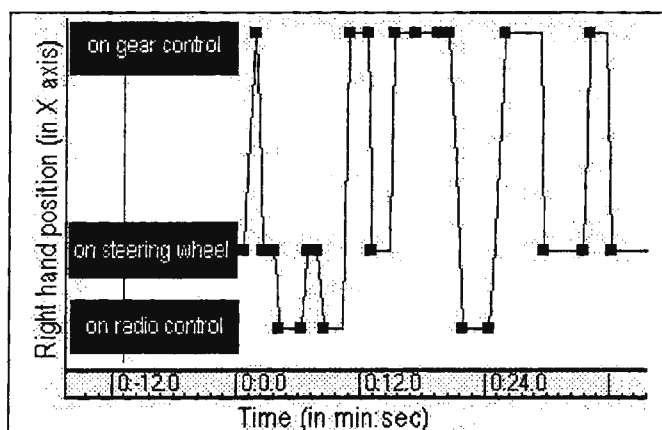
Figure 7 : Scenario example played in 3DStudioMAX environment using the previously defined Right hand states
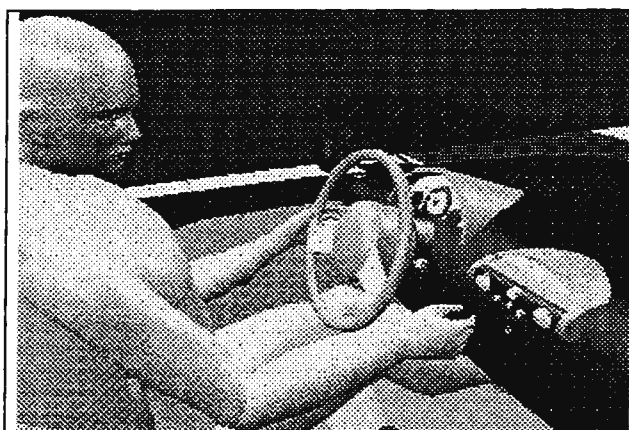
Figure 8 : Behavioural simulation of the global Driver-Vehicle-Roadway system in a virtual reality environment

After this, the virtual films are presented for discussion to the workgroup which suggests changes before running the next iteration : modelling and simulation steps of the new radio concept including the new modifications. After several iterations using our co-operative and iterative innovation process based on digital modelling and simulation steps, an optimised radio concept is selected for detailed studies.

Conclusion

The design methodology, described in this paper, presents our experience with a modelling and simulation approach defined to promote innovation and ergonomics in a concurrent engineering design process. Through a design project of automotive cockpits, and particularly of in-car controls, we have shown how an object-oriented approach and the 3DStudioMAX environment can be used to develop new products. In this project, we dealt with two kinds of actions, integrated in a co-operative and retroactive design process, in order to define new in-car controls increasing comfort and safety. First, in the feasibility analysis, we analyzed driving activity in real conditions when using current controls. We then introduced a modelling phase of the DVR system to prepare, in preliminary studies, virtual simulation of desired activities on the future global system including new controls solutions. In this study, little has been said about object-oriented modelling of cognitive load levels [1] when performing secondary tasks during driving activity. This research area represents a great challenge, but promises substantial rewards in terms of Driver-Vehicle comfort, efficiency and accident prevention.

Bibliography

[1] CHAUVIN C., Modélisation d'une activité de conduite de navire pour la définition de systèmes d'aide, XXXème Congrès de la SELF, Biarritz, p 154 - 161, 1995.

[2] GOMES S., SAGOT J.C., Intervention ergonomique dans la conception d'un poste de conduite automobile, XXXIème Congrès de la SELF, Bruxelles, tome 2, p 149 - 155, 1996.

[3] GOMES S., SAGOT J.C., GOUIN V., Contribution de l'analyse de l'activité dans le processus de conception de produits innovants, 2èmes Journées Recherche et Ergonomie SELF, p 57 - 61, 1998.

[4] KOUKAM A., TARBY J.C, An Integrated model for interactive systems, Human Interaction with Complex Systems : Conceptual Principles and Design Practice, Kluwer Academic Publishers, 1996.

[5] RAUCENT B., JOHNSON D.A., Linking Design and simulation, Journal of Engineering Design, vol 8 - n°1, p 19-31, 1997.

[6] RUMBAUGH J et al., Modélisation et conception orientées objet - Object Modeling Technique (OMT), Editions Masson 1995.

[7] WIERWILLE W., Visual an manual demands of in-car controls and displays, Automotive ergonomics, Ed. Taylor & Francis, 1993, p 299-320.

ROBUST DYNAMIC SIMULATION WITH LINGUISTIC EQUATIONS IN INTELLIGENT CONTROL DESIGN

Esko K. Juuso

Control Engineerig Laboratory, Infotech Oulu and Department of Process Engineering,
University of Oulu, Linnanmaa, FIN-90570 Oulu, Finland,
Esko.Juuso@oulu.fi

ABSTRACT

A new dynamic simulation tool for intelligent control design has been developed on the basis of Linguistic Equation (LE) approach and fuzzy logic. The fuzzy LE approach combines various information sources including mechanistic models, data-driven learning algorithms and expert knowledge to dynamic approximators, state space models and semi-mechanistic models. Understanding of the underlying phenomena and structure of the model are dealed with interactions, and nonlinearities of the system are described by membership definitions tuned by input-output data. Interactions, membership definitions and effective time delays depend on working point conditions. Delays are handled as smoothly changing fuzzy numbers. In the multimodel approach, the smooth switching between various submodels provides an robust platform for testing controller alternatives. The robust dynamic simulator is continuously used in developing adaptive industrial controllers, especially multilevel linguistic equation controllers based on the same approach.

1. INTRODUCTION

Intelligent control provides many new possibilities for the control of complex, nonlinear, partially unknown multivariable systems. Traditionally these controllers have been developed from control. Modelling and simulation can provide many important tools for faster and more accurate tuning of the controllers, especially in slow processes where traditional trial-and-error based on-line tuning is not an efficient solution. However, the tuning of nonlinear multivariable controllers require same properties also in the dynamic models. Simulations can be used for analysis of the system behaviour, design of new processes and control systems. The tradeoff between the necessary accuracy and resulting complexity becomes more and more important when the nonlinear and multivariable behaviour is taken into account. Adaptation to various operating conditions would also be very useful for industrial practice.

Physical (mechanistic, first-principle, "white-box") modelling is based on a thorough understanding of the system's nature and behaviour which is represented by a suitable mathematical treatment. Real systems are usually too complex and poorly understood for a complete mechanistic modelling on an acceptable level of complexity. Gathering necessary knowledge for accurate physical modelling may also be difficult, time-consuming and expensive. Since the models can only represent some special cases, simulations are based on various assumptions and estimated parameters, i.e. the main difficulty is in selecting the models and the parameters for each operating area. The obvious risk of unrealistic simulations is very dangerous in the subsequent steps of analysis, e.g. prediction and controller synthesis cannot be successful.

Data-driven modelling approaches are based on general function approximators ("black-box" structures) which should capture correctly the dynamics and nonlinearity of the system. The identification procedure, which consists of estimating the parameters of the model, is quite straightforward and easy if appropriate process data is available. No process specific knowledge is needed, and actually this a problem for generalization and reliable use of the model since the structure and parameters of these models do not necessarily have any physical or chemical significance. These models cannot be adapted to different operating conditions.

On the other hand, humans can handle complex tasks including significant uncertainty on the basis of imprecise and qualitative knowledge. *Intelligent methods* are based on techniques motivated by biological systems and human intelligence, e.g. natural language, rules, semantic networks and qualitative models. These techniques were introduced by expert systems traditionally used for dealing with the knowledge and experience of engineers and operators. There are various useful methods for knowledge acquisition, but handling of uncertainties is very difficult and time-consuming in conventional expert systems.

Fuzzy modelling and control are extensions of the expert system techniques to uncertain and vague systems. Fuzzy set systems continue the traditions of physical modelling on the basis of understanding the system behaviour. Fuzzy rules and membership functions can represent gradually changing nonlinear mappings together with abrupt changes. *Artificial neural networks (ANN)* introduced new learning and adaptation capabilities by imitating biological neural systems on a simplified level. An ANN can be considered as a fairly complex nonlinear black-box approximator which has similar benefits and drawbacks as conventional identification techniques: easy to use but difficult to explain and relate to physical system properties.

Various modelling approaches try to combine the advantages of the physical and data-driven modelling techniques, e.g. parameters for mechanistic models are approximated by black-box techniques. Since the identification is on a practical level only for linear systems, a lot of work with local linear models is needed. The intelligent methods have extended the toolbox of these *hybrid, semi-mechanistic* or *gray-box* modelling. Fuzzy clustering is an extension of fuzzy knowledge based systems to data-driven techniques. Neuro-fuzzy modelling and identification techniques include fuzzy-logic-based methods to neural computing.

Linguistic Equations (LE) approach combines various intelligent modelling techniques on a unified framework: it was originally developed for handling large knowledge bases in process design [1]; a close connection to fuzzy set systems was important already in the early applications [2]; data-driven modelling properties have brought the approach close ANN techniques. Fuzzy modelling and control was the main application area. Properties of the LE approach are continuously improved and extended on the basis of industrial experience with various application areas, and recently, emphasis is moving from development to direct applications [4, 5, 6, 7, 8].

In intelligent control design, hybrid techniques combining different modelling methods in a smooth and consistent way are essential for successful comparison of alternative control methods. Switching between different submodels in multiple model approaches should be as smooth as possible. For slow processes, predictive model-based technique is necessary at least on the tuning phase. Adaptation to various nonlinear multivariable phenomena requires an highly robust technique for the modelling and simulation.

In this paper, the modelling and simulation is based on the Linguistic Equation (LE) approach. Also other techniques for nonlinear multivariable modelling and simulation are shortly described. Benefits of the approach are demonstrated through two applications on control design.

2. COMPUTATIONAL INTELLIGENCE IN NONLINEAR MULTIVARIABLE MODELLING

Many nonlinear multivariable systems can be transformed into mechanistic models described by differential and algebraic equations. Also data-driven techniques are very useful in modelling, e.g. the linear regression is an efficient method in developing multivariable models. There are more difficulties with nonlinearities: interpreting, generalizing and adapting of nonlinear regression models is not an easy task. In addition to this, there are usually large amount of process knowledge in a qualitative form which cannot be transformed to mechanistic models. This is the area where the Computational Intelligence can provide additional tools: fuzzy modelling is an excellent technique for handling qualitative information, very complex nonlinear models can be constructed by neural networks.

2.1 Fuzzy modelling

Fuzzy modelling integrates qualitative rule-based knowledge to flexible mathematical structures. The expert knowledge expressed in a verbal form is transformed into a collection of if-then rules:

if *antecedent proposition* **then** *consequent proposition.*

Depending on the form of proposition and the structure of the rule base, different types of fuzzy models can be distinguished. The number of necessary rules is much less than in conventional expert systems, but anyway the knowledge acquisition is a problem. Fuzzy models can help in extracting expert knowledge on an appropriate level [9].

Dynamic models can be constructed on the basis of state-space models, input-output models or semi-mechanistic models [10]. In the state-space models, fuzzy antecedent propositions are combined with a deterministic mathematical presentation of the consequent. The most common structure for the input-output models is the NARX /Nonlinear AutoRegressive with eXogenous input) model which establishes a relation between the collection of past input-output data and the predicted output:

$$y(k+1) = F(y(k),...,y(k-n+1),u(k),...,u(k-m+1)), \qquad (1)$$

where k denotes discrete time samples, n and m are integers related to the systems´ order. MIMO systems can be built as a set of coupled MISO models. Delays can be taken into account by moving the values of input variables correspondingly. In the semi-mechanistic models, the modelling task is divided into two subtasks: well-understood mechanisms, e.g. mass and energy balances, are modelled with mechanistic methods, and approximations of partially known relationships and parameters with fuzzy methods.

Fuzzy models can also be constructed from data, which alleviates the knowledge acquisition problem. Various techniques have been to fit the data with the best possible accuracy, but in most cases the interpretation of results is not addressed sufficiently. Fuzzy models can also be considered as a class of local modelling approaches, which attemps to solve a complex modelling problem by decomposing into number of simpler subproblems [11]. Fuzzy clustering can be used for searching an appropriate decomposition of the system. Both the knowledge and data can be combined in fuzzy identification. Also some neuro-fuzzy methods, which represent fuzzy set systems as layered ANN type structures, are widely used in tuning.

The most valuable property of fuzzy models, the ability to provide some insight to the working of the system, should be kept in mind when developing applications. The experts have an important role in model analysis and validation. Interpretability and generalizebility will facilitate adaptation to changing design parameters and operating conditions. Different fuzzy partitions of the anticedent membership functions can be used on different applications (granularity). There are also various methods for estimating the consequent parameters.

2.2 Neural network modelling

Neural computing is a data-driven technique, and therefore, the number of effective parameters should on an appropriate level in order to get a model which can also generalize the results. The most popular neural network architecture is the multilayer perceptron (MLP) which can be considered as a semiparametric method [12]. The correct number of effective parameters can be obtained by various procedures: (1) the number of hidden unit defines how many parameters are available; (2) in the early-stopping regularization the training is stopped when the result starts get worse in respect to the validation set; (3) in the penalty function regularization the complexity of the network is penalized. Also some regularization methods have been developed for using background knowledge, see [12]. Large and complex problems are divided into a set of subproblems by modular hierarchical systems [12]. Neural networks are static mappings in the same way as fuzzy set systems. Dynamic neural network models can be based on similar structures as the dynamic fuzzy models, e.g. the NARX for the input-output models, and mass and energy balances for the semi-mechanistic models. State-space models could be represented as modular hierarchical systems.

2.3 Modelling with linguistic equations

Linguistic Equation (LE) approach provides an unified and flexible environment for combining expertise with experimental data of real systems. Multivariable nonlinear models are constructed in two parts: multivariable interactions are presented as a set of equations for linguistic levels, and linguistic levels are obtained from the values of input variables by membership definitions. The approach consists of nonlinear scaling and matrix manipulation. Each equation,

$$\sum_{j=1}^{m} A_{ij} X_j = 0, \tag{2}$$

where X_j is a linguistic level for variable j, $j = 1, ..., m$, represents a multivariable interaction: the directions and strengths of interactions are defined by coefficients of the interaction matrix. Only the variables with nonzero coefficient belong to the interaction. The original multiplier values $\{-1,0,1\}$ are sufficient for defining the directions, and the strengths can be handled with membership definitions. However, other positive and negative integers has been useful in some applications where the strengths of the interactions are very dissimilar. Nonlinearities are taken into account by membership definitions which consists of two polynomial functions, one for positive and one for negative side labels. With these definitions the values of input variables are mapped to the range -2...2 which correspond to the labels *very low .... very high* (or *negative big ... positive big*); *normal* (or *zero*) is always zero. After the matrix calculations, the outputs are mapped from linguistic level to the real scale.

Linguistic equations can be used in fuzzy form by replacing the right hand side of the Equation (1) with a fuzzy number, i.e. all the variables are not included to the model [9]. The fact that experts do not always agree with the interaction can be taken into account by using several interaction matrices. On the other hand, the directions of the interactions can depend on the working area. In these cases, different interaction matrices have different degrees of membership. In the automatic generation of systems, the resulting equations are in the fuzzy form. Extension to real numbers was introduced because of difficulties to handle variables with dissimilar fuzzy partitions [9]. The FuzzEqu toolbox includes routines for building a single LE system from a large fuzzy systems including various ruleblocks implemented in FuzzyCon or Matlab® FLT. Fuzzy models on any fuzzy partition can be generated from LE models [4].

The development environment includes automatic methods for generating membership definitions and linguistic equations. The system has been implemented in Matlab® environment on a set of three LE toolboxes [9]: LingEqu Toolbox is the basic toolbox for conventional rule-based expert systems; FuzzEqu Toolbox contains extensions to fuzzy set systems; NeurEqu Toolbox is the connecting link to neural networks. The Matlab® toolboxes are embedded to the LE toolboxes: the Fuzzy Logic Toolbox into the FuzzEqu toolbox and Neural Network Toolbox to the NeurEqu toolbox. The FuzzEqu toolbox is widely used in the applications.

3. DYNAMIC MODELLING WITH LINGUISTIC EQUATIONS

The basic form of the LE model is a static mapping in the same way as fuzzy set systems and neural networks, and therefore dynamic models will include several inputs and outputs originating from a single variable. The dynamic behaviour is provided by external dynamic models as state-space models, input-output models or semi-mechanistic models. Rather simple input-output models, e.g. the old value of the simulated variable and the current value of the control variable as inputs and the new value of the simulated variable as an output, can be used since nonlinearities are taken into account by membership definitions. For state-space modelling, each rule correspond to a different working point, and the deterministic equations on the consequent part is transformed into a linguistic equation model. Since the LE model can handle nonlinearities, at least some of the rules can be combined. In semi-mechanistic models, approximation rules are combined into linguistic equations. Effective delays usually depend on the working conditions (process case). Therefore, an appropriate handling of delays will extend the operating area of the model considerably.

3.1 Single model approach

In the single model approach, also variables affecting to the working point of the model are included to the model. In small models, all the interactions are in a single equation. For larger models, the equation system is a set of equations where each equation describes an interaction between two to four variables. The development work starts with an automatic generation of membership definitions which are the used in generation of interaction alternatives. The equations with best fit to the data are selected to the interaction matrix: each new equation introduces at least one variable to the model. Any equation can be rejected or modified on the basis of expert knowledge. For each model, an own delay vector is developed by correlation analysis. The delays should be assessed against process knowledge, especially if normal on-line process data is used [8].

3.2 Multimodel approach

A multimodel approach based on fuzzy LE models has been developed for combining specialized submodels. The approach is aimed for systems which cannot be sufficiently described with a single set of membership definitions because of very strong nonlinearities. Additional properties can be achieved since also equations and delays can be different in different submodels. In the multimodel approach, the working area defined by a separate working point model.

In the FuzzEqu toolbox, the model can include $n \times m$ submodels if there are n working areas and m subareas. Each submodel is a single model whose output is a vector of singleton values. Singleton values of each output variable are aggregated by taking a weighted average. The weight of each submodel is equal to its degree of membership defined by the working point model. The calculation of the degree of membership can be a simple fuzzification procedure or a very complex LE system, e.g. in the web break indicator, the working point model consists of 25 cases, each modelled with about 20 equations [8]. The working point model is usually a steady state model based on direct measurements and some trends. For some variables, changes in signal level or variances and peak values could be useful.

Each submodel has its own delays vector for the variables. In this way the delay is changing with time as the multimodel switches gradually from one submodel to another. For slow processes, also trends are analyzed for some variables. The results are used as additional working point variables. Usually several process cases have a non-zero degree of membership, and these degrees are changing in time, i.e. the delays can be considered as smoothly changing fuzzy numbers.

3.3 Tuning

Tuning is based on real valued linguistic equations and membership definitions. Final membership definitions become consistent with the interactions presented by linguistic equations. Each equation model can be tuned and used to any direction, i.e. any variable can be calculated if sufficient number of other variables is known. For more complex systems, the limitations for solving a matrix equation are taken into account in the development environment. In the tuning of membership definitions, a polynomial function is fitted for the data points and the corresponding linguistic levels obtained by the current membership definitions and interactions. The basic form of the LE model is a static mapping, and therefore dynamic models will include several inputs and outputs originating from a single variable. For the tuning, all these inputs and outputs should have the same membership definitions. In the multimodel approach, submodels are tuned separately.

Testing with independent data is essential in order to use the dynamic models in control design. One-step-ahead testing used in tuning is not enough since the accuracy requirement is very high. A more demanding test is to use the calculated values always as a input on the following simulation step.

4. APPLICATIONS

Dynamic simulators based on linguistic equations are continuously used in development of multilayer linguistic equation controllers. Two application are shortly described in this section. Advanced control methods based on process models, expert system techniques, fuzzy logic and neural networks are of great interest in both processes.

4.1 Rotary dryer

The Control Engineering Laboratory has been developing new control stragies for rotary dryers using model-based control, adaptive control, fuzzy control and neural network based control. The control of a rotary dryer is difficult because of long dead times, long setting times, large disturbances in the input moisture of solids and unmeasurable changes in solids to be dried. The process is highly nonlinear and sensitive to operating conditions. Development of mathematical models is difficult and time consuming. On the other hand, much control experience utilizing operators knowledge is available. Applications have been simulated in Matlab® environment.

The control experiments have been made with a pilot plant rotary dryer [7]. The material is fed to the dryer from a silo with a screw conveyor where it is watered. The drying air is supplied with a blower and it is heated by burning gases from a burning chamber. The dried product is fed back to the silo with a belt conveyor. The process located in Control Engineering Laboratory is connected to Damatic XD automation system. The modelling was based on input-output data including the feed moisture ($X_{s,in}$) , the fuel rate (fuel) and the outlet temperature of the material ($T_{s,out}$). The data was collected from the pilot plant rotary dryer. The values of the feed moisture were filtered with moving average due to the large disturbance. The time delays between inputs and output (about 14 min) were also removed. The time period of data collection is one minute.
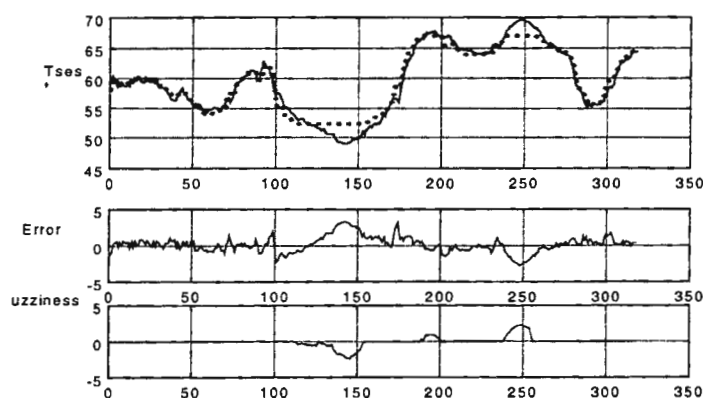
Figure 1. Output of the single model simulator(dashed line) and measured output (solid line).

In the single model approach, the interaction matrix was based on expert knowledge is A = [-1 1 3 −3] with variables $X = [\ X_{s,in}\ \text{fuel}\ T_{s,out}\ T_{s,est}]^T$. The strengths of the interactions are based on the fact that the control variable, fuel feed, has only a minor effect to the temperature level in the normal working area. The calculated and measured values of the output variable after tuning are presented in Figure 1. The fuzziness, which shows how well the equation represents the data, is decreased during the tuning. Tuning reduces the error in those working areas where the most of the data is. Testing will be continued with additional experimental data to find out how general the model is.

Because the single model cannot predict the values of low and high temperatures, the model was further improved using multimodel approach. This is carried out by dividing the data set into three operating areas (*high*, *medium*, and *low*) on the basis of the feed moisture which is not used as an input in the dynamic model. The inputs for the model are then the fuel rate and the temperature of the material ($T_{s,out}$). The model

has membership definitions for each area and they are tuned independently. The interaction matrix A = [0 1 3 –3].

Results of the multimodel with three operating areas were improved especially on high and low temperatures [7]. The multimodel approach gives very good results when operating area is high (moisture is high). In low and medium area results are not so good, but the models work in the right direction. Reason for the inaccuracy is insufficient number of data from medium data. The accuracy can also be improved by further tuning on partly overlapping working areas.

In testing with training data, the delay was removed from data in advance. For controller testing, handling of the delay must be done in the simulation model since control actions will effect to the temperature only after the delay. Using three models instead of one will naturally increase the number of parameters of the system, which means that the fit will be improved. For this reason, testing with additional experimental data is necessary. However, since the direction are consistent with expert knowledge, the results can be considered promising for future use in control design.
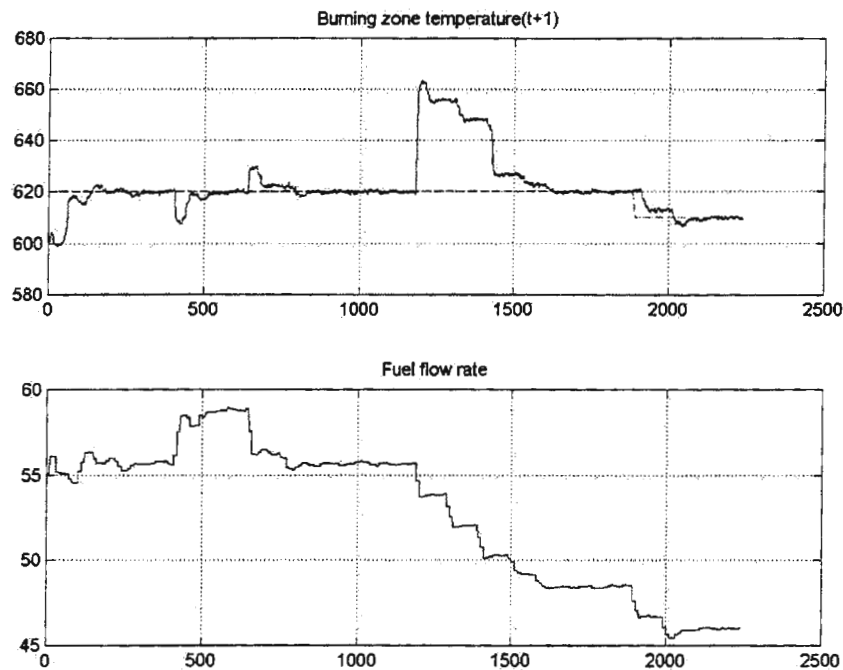
Figure 2. Linguistic equation controller testing with the multimodel simulator.

4.2 Lime kiln

Lime kilns are used in paper industry to convert lime mud to lime in the recovery cycle. Because of the large quantities of lime mud handled, it is desirable to employ an effective control scheme for the economical production of good quality lime. The control of the lime kiln is multivariable by nature. The reaction time in a rotary kiln system is long and varying and there is a large quantity of lime mud and lime in process within the system at any given time. Classical model based supervisory control of process is nearly impossible, and the experience of veteran operator is often the best solution. Control changes should be made in small increments, allowing sufficient time to observe the effect after each change has been made. Overreaction leads to overcompensation, which generally causes oscillations and upsets which may take hours to bring back to normal operation.

Control Engineering Laboratory, University of Oulu has been active in lime kiln modelling and control [4]: the early modelling studies in 70's led to a commercial computerised control system; the models were also included in a steady-state flowsheet simulation package; later sensor based applications and an expert

system application were studied; good results were also gained using fuzzy control in an actual rotary lime kiln in the beginning of 90's. The FLC is implemented in Alcont-II system. Construction of the rule-base and tuning of the controller was done in co-operation with personnel. A new fuzzy modelling technique based on Linguistic Equation (LE) approach was introduced in [4]. In the modelling, data from an industrial scale lime kiln. The LE approach was used in replacing and developing the control rules of fuzzy controller for the lime kiln flue gas fan. This experience is now transferred to a wider control system. The LE approach proved to be a flexible and compact method for developing and testing complex systems.

Extension of the application to the control of the burning end of the lime kiln is now under progress. The burning end controller is a multilevel linguistic equation controller. This new type of controller was first tested on a solar collectors field in a solar power station at Plataforma Solar de Almeria [5, 6]. For this application, simulation is necessary for the controller tuning. In the present model, the multimodel approach with production rate and fuel trend as working point variables is used. The fuel trend effects also to the unsymmetry of the controller. The LE model is now used in controller tuning: an example corresponding to two days operation of the lime kiln is shown in Figure 2. The testing of the multilevel LE controller at UPM Kymmene Pietarsaari mills has been started, and the simulator will be an essential tool in the fine-tuning of the controller.

5. CONCLUSIONS

A hybrid of various adaptive and intelligent methods is a feasible solution for this kind of problems, and the LE approach is a flexible platform for combining expert knowledge, experimental data and simulation results. The simulator operates very well in both applications on the basis of the multimodel LE approach. The robust dynamic simulator based on linguistic equations is an essential tool in fine-tuning of the controllers.

REFERENCES

[1]   Juuso E. K. and K. Leiviskä, Adaptive Expert Systems for Metallurgical Processes. Expert Systems in Mineral and Metal Processing, Proceedings of the IFAC Workshop, Espoo, Finland, August 26-28, 1991, IFAC Workshop Series, 1992, Number 2, Oxford, UK, Pergamon, pp. 119-124.

[2]   Juuso E. K. Linguistic Equations Framework for Adaptive Expert Systems. Proceedings of Fuzziness in Finland *FiF '94*, Plant Design Report Series No. 21, pp. 50-60.

[3]   Juuso E. K., J. C. Bennavail and M. G. Singh (1993). Hybrid Knowledge-Based System for Managerial Decision Making in Uncertainty Environment. In *Qualitative Reasoning and Decision Technologies* (N. Piera Carrete and M.G. Singh, eds), Proceedings of the IMACS International Workshop on Qualitative Reasoning and Decision Technologies *-QUARDET'93*, Barcelona, June 16 - 18, 1993, pp. 234-243, Barcelona, 1993. CIMNE.

[4]   Juuso, E. K., T. Ahola and K. Leiviskä. Fuzzy Logic in Lime Kiln. In Proceedings of TOOLMET'96 - Tool Environments and Development Methods for Intelligent Systems, (Leena Yliniemi and Esko Juuso, ed.), Oulu, 1996, pp. 111-119.

[5]   Juuso E. K., P. Balsa and K. Leiviskä. Linguistic Equation Controller Applied to a Solar Collector Field, Proceedings of ECC'97, Paper 267, 6 pp.

[6]   Juuso E. K. and P. Balsa. Multilevel Linguistic Equation Controller Applied to a 1 MW$_t$ Solar Collector Field, Proceedings of the 1st User Workshop TMR at Plataforma Solar de Almeria, Ciemat, pp. 127-132.

[7]   E. K. Juuso, J. Koskinen, L. Yliniemi and K. Leiviskä. Linguistic Equation Method Applied to Fuzzy Modelling of a Rotary Dryer. In Proceedings of TOOLMET'98 - Tool Environments and Development Methods for Intelligent Systems, (Leena Yliniemi and Esko Juuso, ed.), Oulu, 1998, pp. 145-153.

[8]   Juuso, E. K., T. Ahola and K. Oinonen. Case-Based Reasoning with Linguistic Equations for Web Break Sensitivity Indicator. In Proceedings of TOOLMET'98, pp. 97-106.

[9]   Juuso, E. K. Linguistic Equations in System Development for Computational Intelligence. In Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing –EUFIT'96, Aachen, September 2-5, 1996 (H. –J. Zimmermann, ed.), Aachen, 1996, Vol. 2, pp. 1127-1131.

[10]   Babuska, R., M. Setnes, U. Kaymak and H. B. Verbruggen. Fuzzy Modelling: a Universal and Transparent Tool. In Proceedings of TOOLMET'97 - Tool Environments and Development Methods for Intelligent Systems, (Leena Yliniemi and Esko Juuso, ed.), Oulu, 1997, pp. 1-27.

[11]   Babuska, R. Fuzzy Modeling and Identification. PhD dissertation, Delft University of Technology, Delft, The Netherlands.

[12]   Lampinen, J. Advances in Neural Network Modeling. In Proceedings of TOOLMET'97 - Tool Environments and Development Methods for Intelligent Systems, (Leena Yliniemi and Esko Juuso, ed.), Oulu, 1997, pp. 28-35.

Robust and Central Composite Design as Collaborative Techniques for Production Planning using Simulation

Agostino G. Bruzzone, Piero Giribone
Department of Production Engineering
University of Genoa
via Opera Pia 15, 16145 Genoa
Tel +39 19 97 398 - Fax +39 19 97 600
Email: agostino@itim.unige.it

Abstract

This paper proposes a collaborative analysis method for industrial systems based on the integration of simulation models with Robust and Central Composite Design. The paper describes the proposed approach and presents experimental results concerning an actual industrial plastic moulding plant.

Introduction

The study of complex industrial systems has always been one of the most efficient fields of application for utilising the simulation. The problems to be solved often require the use of advanced analysis methodologies.

In fact, even though the "what if" approach is still widely used from an industrial operating viewpoint, there is a growing trend to replace and/or integrate it with more elaborate methodologies. Obviously, there are many techniques in this field that can be chosen based on the nature of the problem to be solved (optimisation, reverse problems, sensitivity analysis, etc.) and the characteristics of the model being examined.

One particularly effective and flexible technique, that is also widely used in the industrial environment, is D.O.E. (Design of Experiments). This approach can be implemented to develop optimised experimental campaigns to carry out sensitivity analyses and meta-modelling.

The authors propose a methodology that can be applied based on a combination of the CCD (Central Composite Design) and the Robust design. An innovative industrial application of this method is proposed relative to a polymer forming company for production planning analysis. In this way, a support for validating the technique is not only provided by the judgement of the field experts, but also by comparing the results with other detail analyses already carried out by the authors on the simulation model.

This method combines what is certainly high speed and simple application with the enhanced reliability of the results. It is also possible to highlight the phenomena related to the influence of the stochastic factors as well as the average trends concerning the system being examined.

This approach is proposed by the authors to meet the requirements of the industrial experimenters in the field. This is done however without imposing simplifications in advance that may neglect important high-order effects that would invalidate the modelling potential offered by the industrial simulator.

The Integrated Methodology

The design presented in this paper is based on what is mainly an industrial requirement: to analyse the influence of order management logic in a large plant that fabricates plastic piping components.

The number of parameters involved is rather high (there are a few thousand types of products being examined in the industrial case) and even if we consider a family-based logic scheme (an acceptable hypothesis with regard to annual planning), there are more than twenty factors which may have an influence on the objective functions.

In this case, after completing a cognitive analysis through a rapid experimental campaign based on particular hypotheses (which were applicable in this industrial case), it was possible to perform a detailed analysis of the main factors.

However, this methodological approach developed by the authors offsets the high quality of the results with a high computational load and, in any case, enhanced skills by the experimenters (both to verify the hypotheses and to establish the experimental campaign and evaluate the results).

Fig.1 - General System Architecture
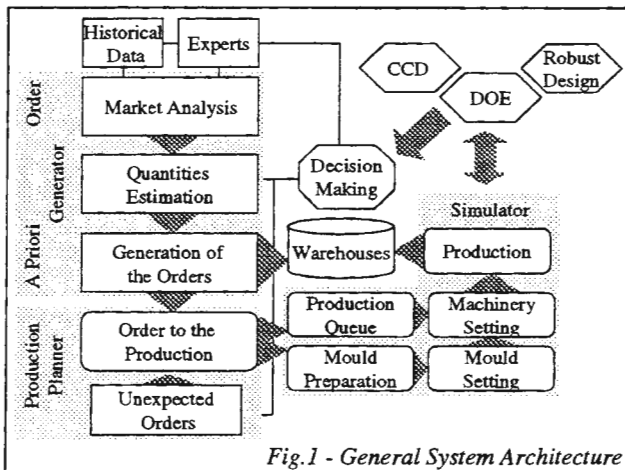
In the industrial world, there are constraints which may jeopardise an approach like the one described: in fact, the time to carry out the experiments as well as the availability of instruments, time and resources to develop, define and analyse the experimental designs, are often unacceptable.

Consider that in the referenced case, the experiment included 36 2^2 factorial designs, and three 2^6 even central composite designs, corresponding to about two computer weeks. In fact, the optimum duration of the simulation run to obtain significant data is determined through an analysis of the evolution of the MSpE. For this reason it was decided to develop a new methodology that, with a limited calculation load, generates valid and especially reliable results. Such results should cover both the aspects concerning the analysis to determine the robustness of the model and those related to RSM (Response Surface Methodology) to support subsequent optimisation phases and immediate valorisation through meta-models of operating hypotheses on the real production plant. This approach is based on the use of two types of experimental designs: robust design and central composite design.

The two methods have numerous points in common since the second one (Robust Design) is derived from specific cases of Fractional Design. For years there has been a rather heated discussion regarding these methods, highlighting the advantages and disadvantages of the two techniques each time.

Instead, in this paper, it was decided to use these two different types of experimentation methods in a combined format.

The two designs are developed in a combined mode to maximise the number of common experimental points and thus reduce the overall computational load. In addition, this balances the number of

experimental tests to provide a usable quantity of equivalent information for the two designs to guarantee that the results can be compared for a mutual verification. Thus an effect analysis is performed (multiple-factor sensitivity analysis) that identifies the greater influence of the variables involved with the CCD and evaluates the trends obtained on the Robust Design.

In the industrial case used as reference each component (product class) involves two different independent variables:

l   re-order point

p   dimension of the production lot

In effect it is quite common to have different independent variables related to the same object for a series of different objectes: this methodology requires a preliminary series of factorial design obtained grouping all the independent variables by the corresponding reference object (in this case a product class):

$$k = n_{oo}\, n_g \qquad (1)$$
$$n_{cd} = k / n_g \qquad (2)$$

n_{cd} number of factorial design

k number of independent variables

n_g number of variables concerning each object

n_{oo} number of objects

So the number of experimental runs becomes:

$$Nf_{tot} = No \cdot n_{cd} \cdot b^{n_g} \qquad (3)$$

Nf_{tot} total number of experimental runs for the factorial designs

No replications required for experimental error identification

b base of the factorial design

The target of this analysis is to identify the influence of the different input factors, so for each independent variable is fixed an equivalent experimental range; all the ranges are centred on the same reference values.

Fig.2 - Influence of Different Factors on the Lead Time

Table I-Experimental Input for 6 x 2² central design

C0		C1		C2		C3		C4		C5		C6		C7		C8		C9		C10		C11	
l	p	l	p	l	p	l	p	l	p	l	p	l	p	l	p	l	p	l	p	l	p	l	p
-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
+	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
+	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	+	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	+	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	+	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	+	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	+	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	+	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	+	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	+	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	+	+	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-	+	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	+	-	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	+	+	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-	+	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	+	-	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	+	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	-	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	+	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	-	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	+	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	-	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	+	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	-	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	+
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	-
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

If it's correct to accept the hypothesis of an uniform distribution of the experimental error, it becomes possible to realise m factorial design with the same centre; in this way it's possible to substitute the factorial design with central design and to estimate the experimental error just based on No replicated runs on the centre of the experimental range.

By this approach the number of the experimental runs for preliminary sensitivity analysis is reduced as follows:

$$N_{tot} = n_{cd} \cdot b^{n_g} + No \qquad (4)$$

N_{tot} total number of experimental runs for the central designs

The experimental matrix of the input corresponding to the case study is proposed in table I, where:

0   reference central value for the indipendent variable

+   higher value of the range

-     lower value of the range

In this case the base for the experimental design is 2 (this is very common in industrial problems due to the necessity to reduce the computational work load), No is fixed equal to 8 and the total of the experimental runs is 56 corresponding to 6 experimental designs 2^2 instead than a single 2^{12} factorial design.

The estimation of the effects is based on contrast methodology:

$$MSE = \frac{\sum_{i=1}^{No}\left(Ys(Xc_i) - Ycm\right)^2}{N0 - 1} \qquad (5)$$

$$MSeff_x = \frac{1}{2^{n_g}}\sum_{i=1}^{2^{n_g}} c_{tab^i} Ys(Xf_i^x)^2 \qquad (6)$$

$$Feff_x = \frac{MSeff_x}{MSE} \qquad (7)$$

MSE   Mean Square of the Experimental Error

Xc_i i-th central replicated input

Ys   Experimental Simulator Output

Ycm   Y Mean for central replicated runs

Xf_i^x i-th run of the sub design involving the x independent variable

$MSeff_x$ Mean Square of the x effect

$Feff_x$ Sensitivity Index for x effect

c_{tab^i} Coefficient in the experimental sign table (+ equal to +1, - equal to -1).
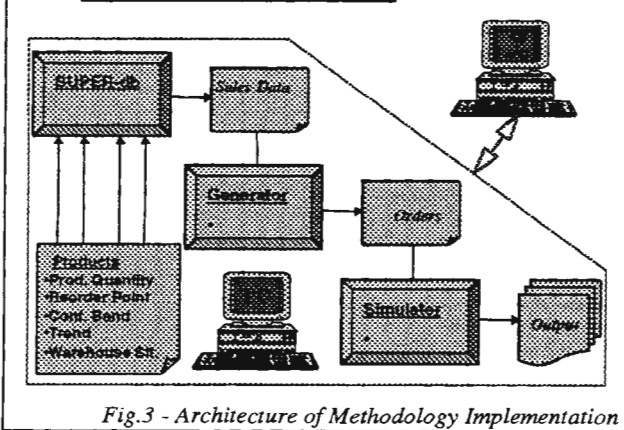

Fig.3 - Simulator General Architecture
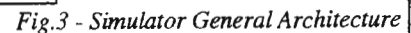
Fig.3 - Architecture of Methodology Implementation

This approach allows to estimate not only the single pure effect of the pure independent variables (in this case l and p) but also the influence of the interactions of higher order (in this case just the combined 2nd order effect l p); the effect due to interactions can be estimate generating in the sign table new columns corresponding to combination the of the pure effects (i.e. column l p will be the result of column l by column p).

The analysis of variance allows by statistical test to verify the significance of each effect; this could be easily obtained comparing the $Feff_x$ and a Fisher distribution with corresponding confidence level and degree of freedoms; the first degree of freedom is one ($Mseff_x$ degree of freedom) in our experimental approach due to the fact the that the experimental factorial point are not replicated; the second degree of freedom correspond to the N0-1 (MSE degree of freedom).

The second step in the methodology is to identify the most important factors (that ones able to pass the statistical significance test) and to apply a single large central composite design on the full set of the corresponding h independent variables.

So it becomes possible to develop two different experimental design (Central composite and robust) based on some common point, centred on the same central condition and with the same range of variation for the h independent variables.

The author proposes to use the contrast methodology on this central composite design and to estimates

$$Contrast_x = \frac{1}{2^{h-1}} \sum_{i=1}^{2^h} c_{tab^i} Ys(X_i^x) \qquad (8)$$

h   number of significant variables

The sensitivity of a pure effect could be estimated also with the Robust Design estimation:

$$MPx = Ysm(Xlx) - Ysm(Xhx) \qquad (9)$$

MPx   Delta on the Mean for x variable
Xlx   experiment with variable x low
Xhx   experiment with variable x high
Ysm   mean value in point X
          for replicated runs performed by
          the simulator on the robust design

This analysis is used to evaluate the effects of the single factors and their interactions, while the deviation between the two evaluations obtained is indicative of the capacity of the CCD to consider higher order effects neglected by the Robust Design:

Due to the fact that all the ranges are equivalent and the project and centred on the same point the two factors can be compared as relative values:

$$Crelx = \frac{Contrast_x}{\max(Contrast_{xi})_i} \qquad (10)$$

$$Mrelx = \frac{MP_x}{\max(MP_{xj})_j} \qquad (11)$$

If this deviation cannot be attributed to combined factors, important information is obtained about the verification of the approach for the case being examined (unsatisfactory experimental samples) and thus attention is focused on a more complete methodology.
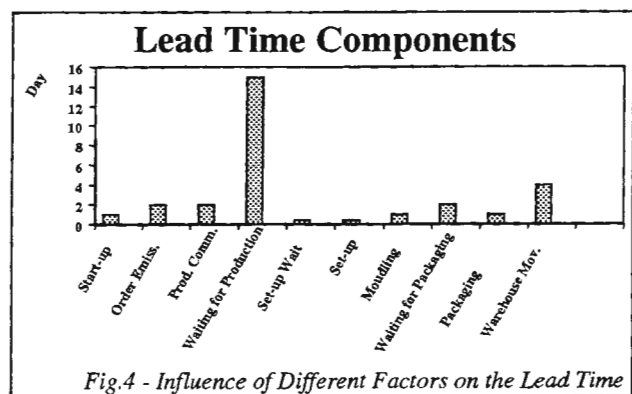

Fig.4 - Influence of Different Factors on the Lead Time

Based on the experimental data obtained, the approximating surfaces of the meta-models obtained from CCD are compared with other interpolating surfaces supplied by the Robust Design.

The deviation between these meta-models and the simulation data is expressed as:

$$Dv(Xeff) = Fccd(Xeff) - Frd(Xeff) \quad (12)$$

$$Den_{td} = \frac{\sum_i U(\delta - |Xeff - X^{td}{}_i|)}{\delta^h} \quad (13)$$

Dv	Deviation CCD and Robust Design
Fccd	Metamodel Output from CCD
Frd	robust design Metamodel Output
Den_{td}	Density of information for the design td (CCD or Robust Design) in the point Xeff
Xeff	Point under analysis inside range of experimental design
$X^{td}{}_i$	i-th experimental point in the design td
δ	Dimension of the reference area for density calculation
U	Heaviside Function

In effect this way to estimate the density of experimental points used by each design is very simple, but have a strange behaviour (i.e. discontinuities), therefore this approach allows an easy computation and in the applications experimented provided good results for δ equal to ½ of homogenous unitary ranges.

The dimension of the reference area is to be fixed equal to half amplitude of the experimental ranges or greater; this set guarantee non null values of the density on the border areas of the experiments.

Based on this value it's proposed to use as metamodel that one that provides the higher density value.

This term, when considered in terms of the density of information of each design, represents an evaluation of the best efficiency of the system compared to the other one. Also in this case a well-defined experimental campaign is expected to provide greater reliability of the model that locally provides more information. With a special additional simulation campaign in the intermediate points, it is possible to verify the overall meta-model as:

$$Mt(X) = \begin{cases} M_{ccd}(X) & Den_{ccd}(X) \geq Den_{rd}(X) \\ M_{rd}(X) & Den_{ccd}(X) < Den_{rd}(X) \end{cases} \quad (14)$$

$$EM(X) = \begin{cases} M_{ccd}(X) - Mt(X) & Den_{ccd}(X) \geq Den_{rd}(X) \\ M_{rd}(X) - Mt(X) & Den_{ccd}(X) < Den_{rd}(X) \end{cases} \quad (15)$$

$$Efc = \int_{range} EM(X) \cdot KDY \cdot dX \quad (16)$$

Mt	Combined Metamodel
Mccd	CCD metamodel
Mrd	Robust Design interpolated metamodel
KDY	factor for convenience of overestimation of the output : equal to 1 for target function that it is more convenient to underestimate from application point of view and -1 for the other ones; i.e. in our case lead time is preferable to be overstimate from industrial risk point of view (KDY=1), while productivity is safer to be underestimate (KDY=1)
Efc	Efficiency of the metamodelling

The Efc is a measure of the safety in terms of output metamodelling proposed by applying this methodology to industrial cases; of course the integral becomes a sum of experimental points (the experimental design points for a basic check or an extended set with additional simulation run for extensive validation).

The Signal to Noise can be used in order to estimate the robustness of the target function on the input perturbations:

$$S/N = -10 \log(y_m{}^2 / s^2) \quad (17)$$

S/N	Signal Noise ratio
ym	measured Signal
s	measured variance

In effect in similar way respect to the output metamodelling, it's possible to perform an analysis of the confidence band provided by the different experimental designs; it's necessary to consider that using CCD the confidence is extended in homogenous way from the central estimation based on the distribution of experimental points, while for Robust Design this estimation is performed on the factorial point.

The Industrial Problem

As already mentioned, the industrial case refers to a large production plant that fabricates plastic components (especially piping components with high added value: valves, fittings, etc.).

Polymer injection moulding is a complex production process involving numerous factors.

The simulator developed by the authors is proposed as the means of reproducing this procedure to provide a support for production planning and to establish new order management policies.

Obviously, this tool can also be used quite efficiently to carry out an operative support analysis for production scheduling and plant reorganisation.

However, because of the condition of the industrial partner, the analysis had to focus on planning aimed at maximising the objective functions.

This was necessary since the sector is considered to be particularly critical and the model must be carefully validated on the medium-term problems before implementing it directly on-line on the plant. The annual planning problem has a significant effect on the quality of the customer service.

Ya -S/N [dB]

Yb -S/N [dB]

Yc -S/N [dB]
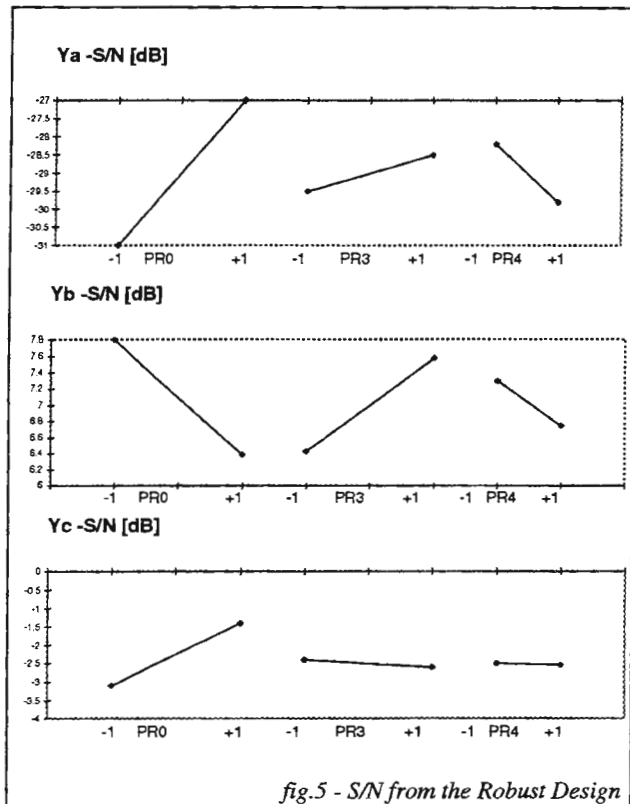
fig.5 - S/N from the Robust Design

In fact, it is closely related to the lead time of the single products: with regard to the total lead time (time elapsed from when the order is received in the plant and when it is prepared for shipment) the part attributable to the moulding queues is often very high (in this case equal to about 80%).

Obviously, planning also effects the use of resources and the company system production capacity. These objectives often conflict and the various performances must be evaluated in terms of a compromise.

The production process is shown in fig. 1 and includes: determination of the annual quantities, generation of production orders according to the production lot and minimum stock level hypotheses,

assignment of orders to moulding machines, mould preparation and production.

Naturally, the system is also influenced by important stochastic components concerning malfunctions (Mean Time Between Failures) and how long such failures continue (Mean Time To Repair):

MTBF, MTTR machine failures

MTBF, MTTR mould failures

A stochastic component with an even greater influence is the evolution in demand during the year and thus the change in the predictable production volumes.

The cost relation in this case is our target function in order to combine the different aspect of the problem (productivity, waiting time, use coefficients):

$$C = C_m + n_M (C_o T_p + C_{no}) + \int_0^{MLT} \left[C_m + \frac{n_M (C_o + C_{no}) t}{MLT} \right] h \cdot dt$$

The influence of the different factors is proposed in figure 2, as it could be seen the production have a great influence on theoveral lead time.

The planning system for the budget analysis currently used manages production in terms of the re-order point (a method that was implemented in the model). The problem is particularly complex to study also because of the large number of products (more than 6000 types, of which about 2000 types are produced within one year out of a total of more than 4000 orders), the great amount of machines present (about 50 presses with different functional characteristics), the mould situation (thousands of moulds with specific connections to multiple presses) and the technological procedures (plastic colour change, normal mould settings, press settings, etc.).

The Simulator

In this case, the simulation is the only tool that can evaluate the consequences of different planning hypotheses. However, management is obviously interested in obtaining information about how the planning management parameters can be changed as efficiently as possible. The authors used an analysis of the historical data and the know-how of experts to identify 12 main groups of finished products. This classification was based on a three-dimensional ABC analysis in terms of sales volume, physical production volume and equivalent machine time.
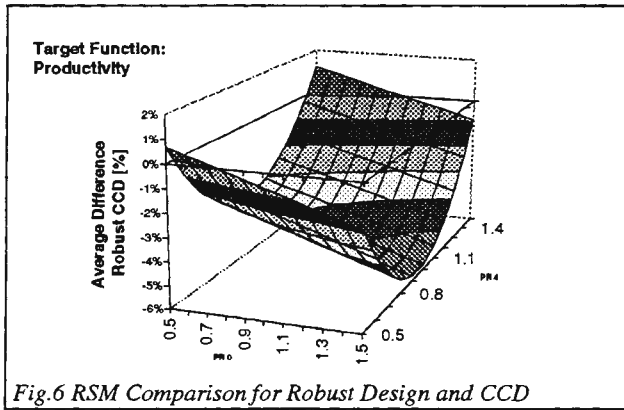
Fig.6 RSM Comparison for Robust Design and CCD

Table II - Confidence Band of the Model	
Output Function	PS [%]
Mean Lead Time	0.824
Standard Deviation Lead Time	4.255
Moulding Efficiency	0.126
Material Change Efficiency	1.787
Setting Efficiency	1.515
Mould Change Efficiency	0.453
Total Productivity	0.984
Effective Production Capacity	2.712
Machinery Breakdowns	2.874
Mould Breakdowns	3.049

For each of these groups, the manageable parameters were considered to be the corrective coefficients of the re-order point and the size of the production lot (defined by management according to the information from the technical and commercial managers). The tool developed by the authors can still be used to change only the products required by the users. Following the analysis performed with the previously mentioned experiment, a ranking list was created of the 24 factors being examined. This was followed by an analysis of the main factors.

Modelling was based on an Object-Oriented Analysis of the problem thus leading to the development of the structure reported in fig.3, while fig.4 presents the system implementing the methodology. A combined simulation model was developed. Such a model operates with discrete events, while evaluating the production technological process with each event for all presses to obtain a correct estimate of the quality and the quantities produced. This is a stochastic type of model. A special analysis of the historical data obtained from the plant helped to identify the type of distribution, while the statistical parameters were determined by means of a best-to-fit analysis.

Implementation was developed using C/C++ library models developed by the authors during their extensive experience in the analog plant modelling sector (i.e. glassworks). This produced an excellent relationship between efficiency of the simulator developed, maintenance capacity and development time.

The Experimentation

The model was provided with a synoptic interface that could represent the plant operating dynamics which was important for the model verification and validation phase. A statistical validation was performed on the results obtained from previous scenarios. Instead, the time evolution analysis methodology of the mean square pure error (MSpE), developed by the authors, was applied to identify the optimum duration and to obtain a good estimate of the effect of the experimental error.

The run time was about 20 real minutes corresponding to one year of production. Model precision was expressed through the relationship:

$$PS = 2 \cdot t_{\alpha,v1} \frac{\sqrt{MSpE(tf)}}{Ymed(tf)} \quad (18)$$

PS	Model Precision
MSpE(tf)	MSpE at the tf time
tf	optimal simulation runtime
Ymed	Mean value of Y at tf time
$t_{\alpha,v1}$	Student Distribution with α confidence, v1 degree of freedom (No)

So PS represents the amplitude of the confidence band respect the average value of the target function under analysis.

Therefore, the relationship PS between confidence band and mean value of the objective function was always widely within the ranges required by the initial specifications (5%), generating a particularly high precision level (see tab. II).

The objective variables considered from among the various factors (use coefficient of each individual press, lead time of each product/customer, effective production capacity for each product, etc.) were the synthetic variables that could highlight the key factors: quality, productivity, potential.

The factors evaluated were:

Ya	Mean lead time among the various products
Yb	Machine mean use coefficient
Yc	Effective Production Capacity with respect to the market request among the various products

Oviously in this robust design the target functions Ya and Yb are subjected to "the smaller the better" criteria, while the Yc uses "the larger the better".

The experimentation was based on the construction of two designs: central composite (2^3 factorial plus 4 central replicated runs) and robust (L4 with 3 replications) in relation to the following 3 indipendent variables (the most important ones based on the preliminary sensitivity analysis):

X_A	PR0	re-order point of product class 0
X_B	PR3	re-order point of product class 3
X_C	PR4	re-order point of product class 4

The results are summarised in the figures which highlight, the influence of the S/N (Signal/Noise) ratio and thus the robustness of the design and the correlation between these inputs and the outputs obtained (fig.5).

The comparative analysis (in fig. 6 and 7) presents the analyses of the effects obtained through a much more complex experimental design. The good approximation provided by this modelling technique can be noted: : both the estimations of mean value and of the significance band are normally quite equivalents (less then 3%).
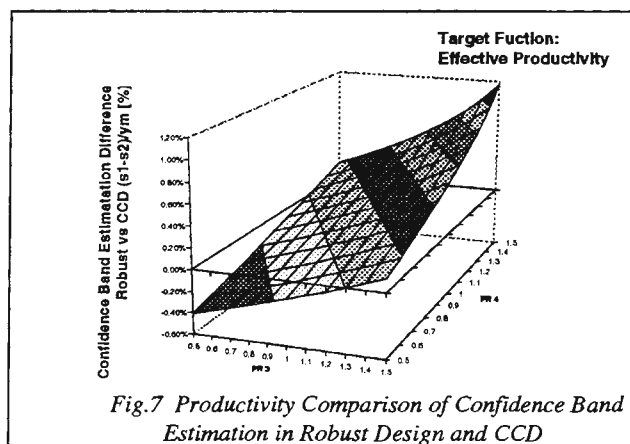
Fig.7 Productivity Comparison of Confidence Band Estimation in Robust Design and CCD

Conclusions

The proposed approach will use existing techniques rapidly but with major guarantees concerning the reliability of the results. The introduction of a new methodological process and indices to verify the models makes it easy to apply the technique.

This method can be easily implemented to automate and accelerate the procedure with respect to the quantity/quality of the information it can provide.

Thanks to its application to a real case, with the level of complexity reported in this paper, the method could be developed with particular emphasis on the needs of the operating field, without neglecting the complexity being examined. The authors intend on evaluating this approach based on new situations and with different types of designs in order to extend its validity and verify its efficiency in the field.

References

- Box G.E.P, Hunter W.G., Hunter J.S. (1978) "Statistics for Experimenters", John Wiley & Son, New York
- Box G.E.P. Hicks W. (1990) "A quesiton of Quality", Harvard Business Review, March April, pp.225
- Bruzzone A.G. (1994) "Manris: Introduction to the Scheduler Package", Bormioli R. Technical Report, Parma
- Bruzzone A.G. (1994) "Potential of Reduced Experimental Designs in Object-Oriented Modeling of a Manufacturing Department for Film Production", Proceedings of ESM94, Barcelona, SP, June 1-3
- Cochran, Cox (1957) "Experimental Design", John Wiley & Son, New York
- Fowlkes W.Y. Creveling C.M. (1995) "Engineering Methods for Robust Product Design, Addison Wesley", Reading MA
- Giribone, P. (1995) "Minimum Aberration Fractionary Designs for Application with Stochastic Simulators of Complex Industrial Plants", Proc. of ESS95, Erlangen, D, October 26-28
- Montgomery D. (1991), "Design and Analysis for Experiments", John Wiley and Son, New York
- Mosca R., Giribone P, Guglielmone R. (1982) "Optimal Lenght in Operational Research Simulation Experiments of Large Scale Production Systems", Proceedings of 2nd Modelling, Identification and Control, Davoz, March
- Mosca R., P.Giribone & A.G.Bruzzone (1993) "Optimum Area Search Techniques Applied to Studies Relative to Plant Problems performed by means of Simulation", Proceedings Simtec93, San Francisco, November 8-10
- Mosca R., P.Giribone, A.G.Bruzzone (1994) "Project and Management of Automatic Yogurt Lines Through Simulation", Proceedings CEEDA94, Bournemouth, April 7-8
- Mosca R., P.Giribone, A.G. Bruzzone (1997) "The Potential in the Integration of Tools for Plant Floor Solutions", Proceedings of MIM'97 IFAC, Vienna, Austria, February 3-5
- Mosca R., Giribone P., Bruzzone A.G. (1997) "Application of Experimental Design Methodologies to Management Re-Engineering in Industrial Assembly Department by Using Simulation", Proceedings of 1st World Congress on Systems Simulation, WCSS'97, Singapore, September 1-4
- Taguchi G. (1987) "System of Experimental Design", ASI & Quality Resources
- Taguchi G. (1990) "Robust Design", Harvard Business Review, March April, pp.229

COMPUTATIONAL INTELLIGENCE IN SIMULATION OPTIMIZATION: A DISTRIBUTED TOOL USING PVM AND ITS APPLICATION IN MANUFACTURING

J.L. Paris, H. Pierreval and L. Tautou
Laboratoire d'Informatique de Modélisation et d'Optimisation des Systèmes,
Equipe de Recherche en Systèmes de Production de l'Institut Français de Mécanique Avancée
I.F.M.A. - B.P. 265,F-63175 Aubière Cedex
Phone: 33 (0)4- 73-28-80-00, Fax: 33 (0)4-73-28-81-00
E-mail: paris@ifma.fr, pierreva@ifma.fr and tautou@ifma.fr.

ABSTRACT

In this paper, we propose a new simulation optimization tool based on distributed Evolutionary Algorithms (EAs). This tool uses PVM to manage distributed searches for solutions on a network of SUNworkstations, organized as an hypercube. One of the major application area of this tool is the optimization of simulated complex systems, where a particular attention has to be paid to the selection of design options (e.g. choice of a transportation device : select the option AGVs or the option conveyor ?) and to the choice of parameters of several types, which can themselves depend on the options selected. We propose several innovations for the EA in order to take into account these design options. The benefits of the tool we propose in this paper are demonstrated through three examples in manufacturing, but the approach can be used in other application areas.

1. INTRODUCTION

Simulation optimization methods have been studied for several years. Recently, some pieces simulation of software have tried to integrate optimization modules, for example Microsaint (Optquest), Witness and Cadence. Unfortunately, the use of a single processor considerably limits the efficiency of the simulation optimization process because of the computing costs required to perform simulation replications and to the large number of points explored in the search space.

In this paper, a new distributed simulation optimization tool, which increases the possibilities of simulation optimization is presented. We propose to use distributed evolutionary algorithms in order to reduce the computing time and to enlarge the regions of the search space explored. This approach has been implemented through the environment PVM (Parallel Virtual Machine), which greatly facilitates the software developments.

One of the major application area of this tool is the optimization of the performances of manufacturing systems where the problem to address is to determine the best combination of set of variables (e.g. size of buffers, transport lot sizes, dispatching rules) in order to optimize a performance criterion. Moreover, when designing manufacturing systems, a particular and difficult attention has to be paid to the selection of design options (e.g. choice of transportation device : select the option AGVs or the option conveyor ?). Each option may itself require the determination of some particular parameters (number of AGVs or speed of the conveyor) and some other (sub)options (which path for the AGVs ?). Depending on the option that have been chosen, the total number of parameters to globally determine can be different. In our approach, several innovations for EAs have been done in order to take into account design options.

In this paper, we will first propose a distributed simulation optimization approach based on EAs. We will then present the corresponding optimization tool that we have implemented using PVM. In the next section, we will introduce the innovations we have carried out to take into account design options in evolutionary simulation optimization. The benefits of our approach will be demonstrated through three examples which were solved using our optimization tool.

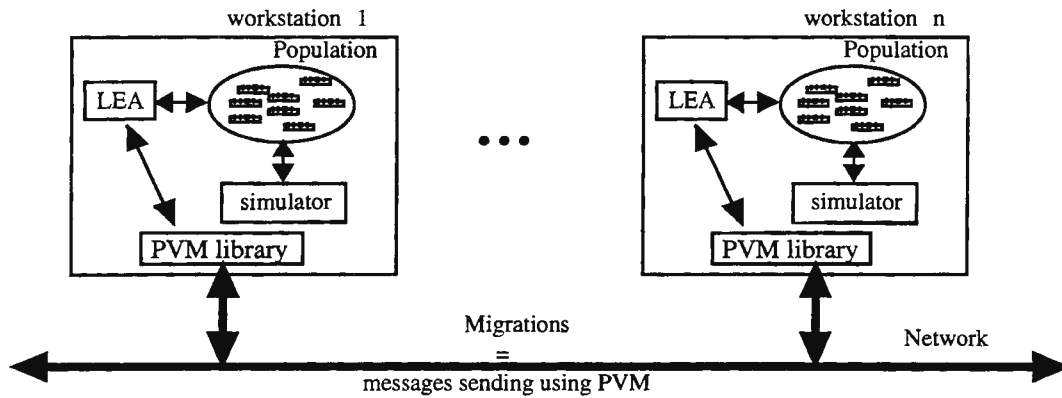# 2. AN ENVIRONMENT FOR DISTRIBUTED EVOLUTIONARY OPTIMIZATION

2.1 Sequential Evolutionary Algorithm

Evolutionary algorithms (EA) include various methods, such as genetic algorithms, evolution strategies, evolutionary programming, genetic programming, etc. A classification of certain of these approaches is given in Bäck and Schwefel (1993) [4]. However, there are many evolutionary systems which are difficult to classify. These approaches share several principles inspired by natural evolution phenomena. One of the essential principles used in several evolutionary approaches is that a well adapted individual has a high probability to survive. These algorithms begin their search for the optimal solution from a set of potential solutions (the *individuals*, characterised by their *chromosomes*), which is called the *population*. Hence, each solution is coded through a chromosome, which is represented by a vector X^k. The components of a chromosome may be binary elements, integers or reals. EAs make the initial population evolve toward a population, which is expected to contain the best solution. For that, they use the following *reproduction-evaluation* cycle: for each iteration, called *generation*, chromosomes from the current population are selected with a given probability favouring the best and copies from these individuals are created. In a simulation optimization approach using an EA, an individual is a vector of input values of a simulation model and its *fitness* is a function of the output variables of the model [12][14][16]. New individuals, produced by this process, are submitted to *mutation* (random modifications) and to *recombination*, which allows the mixing of two individuals to product children. From a generation to another, individuals have globally a higher fitness. The reader can find more details in ([11], [3]). Although EAs are known to be a powerful optimization technique, they are notoriously slow [6]. To cope with this drawback, we are interested in the parallelization of such algorithms for their use in simulation optimization.

2.2 Distributed Approach

Evolutionary algorithms present by nature good characteristics to be parallelized. However, they need some adaptations to be implemented on several processors. An example of bottleneck for parallelization may be the selection process, which may require, at each generation, a comparison of all the solutions performances. Consequently, several methods related to the parallelization of genetic algorithms have already been published ([10][6]). In our approach, the population is divided into several sub populations, which can be managed by their own Local Evolutionary Algorithm (LEA). A processor will be responsible both for the management of the LEA and for running local simulation experiments. Each LEA has its own parameters, i. e., mutation and recombination probabilities. It has to be noted that the random number seeds used by the LEAs to generate their initial local populations and for selection and reproduction are different to ensure the necessary diversity. The local sub-populations can be relatively large and kept relatively isolated from each other, so that such an approach is well adapted to distributed memory computers.

A *migration operator* is used to exchange individuals (called *emigrants*) between populations according to a given migration rate, in order to propagate good solutions and to help unsuccessful LEAs. Migrations occurs when local populations degenerate. A population *degenerates* when the best individual is the same since some generations [5]. Migrations are performed at a given *migration rate* (number of individuals moving to another population / number of local individuals). Some individuals are selected among the best using the wheel principle and sent to each neighbouring LEA. Each LEA has its own migration rate and maximum number of generations without improvement (used to trigger mutations and integrations of immigrants). Hence, LEAs may not evolve at the same speed, especially if they have different population sizes or different recombination and mutation probabilities. Communications between LEAs must be asynchronous. As a consequence, the set of immigrants sent from a LEA to another will wait in a buffer that the receiving LEA degenerates and therefore needs to incorporate new individuals. In order not to waste time, if a LEA needs new individuals to be incorporated and if its buffer is empty, it will then continue for several generations before scanning again the buffer. When several sets of immigrants wait in the buffer, all of them are incorporated in the local population set by set. In order to keep the local population sizes constant, the incorporation of new individuals necessitates to remove the same number of old ones. A selection is therefore performed, favouring the worth using the wheel principle. LEAs have to be connected to each other to form a network in order to be able to send emigrants to other local populations. Cohoon et al [7] have noted that the topology of this network must have a high connectivity and a small diameter to be high performance. They have proposed a hypercube topology.

Figure 1: distributed evolutionary algorithm on a network of workstations.

2.3 Distributed Environment

The system we have developed is implemented on a network of eight SUN workstations. The communication between LEAs are managed by PVM [1]. PVM is an environment able to emulate a distributed memory multiprocessor system on a network of different types of machine using message sending. For example, Reinefeld has used PVM in optimization area [15] ; Schuster and Breitenecker has used it for parallelization of simulation models [17].

Our global application is composed of a master program and n slaves programs written in C language using PVM library. According to the classification proposed by Cosnard and Trystram (1993) [8], this environment is an asynchronous Single Program Multiple Data (SPMD). When PVM is running, the master program creates one slave program per machine. Each slave will be responsible of one evolutionary algorithm. When all slaves are ready, the master program send a signal to each slave to start evolutionary computations. When a slave needs a migration, this slave performs an asynchronous communication to all of its neighbours through the network. This global application ends when every slave reaches a end criterion (generally a number of generations). The master program gets then the results and kills every slave.
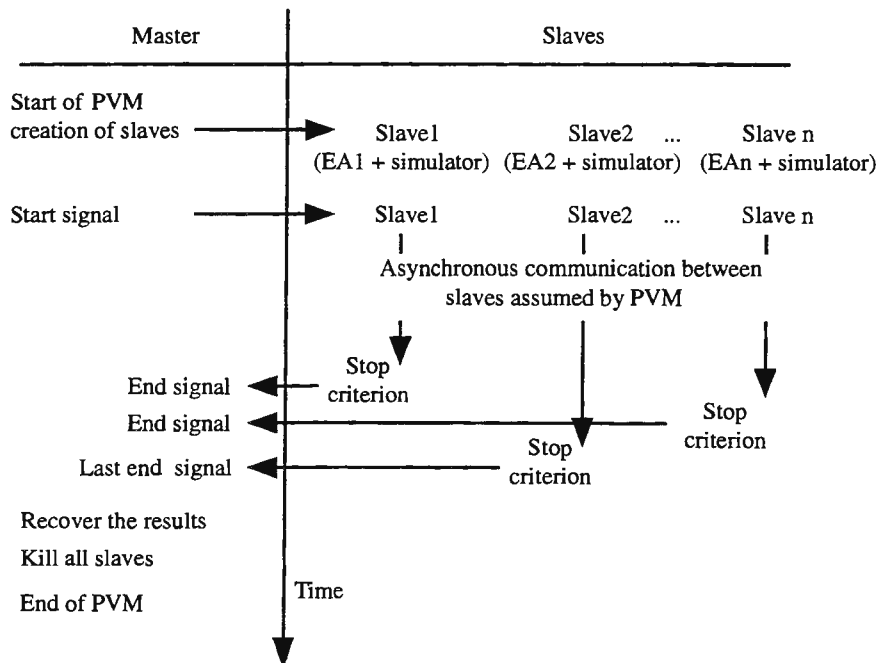
Figure 2: chronological interactions between processors through PVM.

3. DATA STRUCTURE AND OPERATORS FOR EVOLUTIONARY OPTIMIZATION OF THE DESIGN COMPLEX SYSTEMS

3.1 New Data Structure

Most simulation optimization methods are limited to vectors [2], which are no more suitable in the case of design options. In fact, each option may itself require the determination of certain particular parameters and certain other (sub)options. Let us consider the example of a single product manufacturing flowshop system, which can be managed either by a Conwip or a Kanban approach [9]. If the option Conwip is selected, the only variable to be determinated is the number of conwip cards, otherwise, the number of kanbans has to be determinated for every machine. Then, according to the option chosen, the total number of parameters to globally determine can be different.

To describe all the possible configurations of the system, we use a tree where the root is a vector whose components describe the parameters and the options concerning the global system, a node is a vector whose components describe the parameters and the options of a subsystem [18]. For each design choice in the subsystem, there are several branches which represent the different possible options (c.f. figure 3). Thus, a solution is coded as a tree of vectors. The components of a vector are the parameters or the options which characterize the concerned subsystem. A most detailled tree can be seen on figure 5.
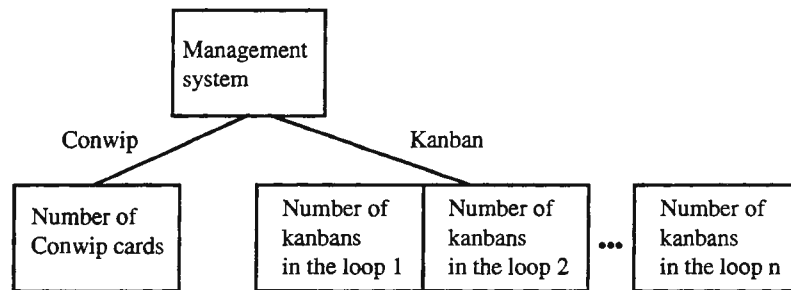
Figure 3: data structure for design options in manufacturing systems.

3.2 New Evolutionary Operators

New evolutionary operators are needed to use the representation of a solution. The general frame of the mutation operator applied to a solution b is the following: choose randomly an element p among elements (options and parameters) of b ; choose randomly a value p' in the domain of p ; replace p by p'. We have implemented two recombinaison operators. The general frame of the first operator is the following: with the same probability choose to cut a branch or a vector ; choose randomly a branch of the first parent (respectively a vector and a position in this vector) ; cut the branch (respectively the vector) ; exchange the subtrees of both parents (respectively the end of the vectors of both parents). The second operator can be expressed as follows: give a number (between 1 and the total number of parameters (tp)) to each element of the tree with a width first search ; choose randomly a position p between 1 and tp ; exchange the elements with numbers superior to p of both parents. More details are given in ([18]).

4. APPLICATIONS TO MANUFACTURING SYSTEMS

4.1 A Flowshop System

The first studied system is an hypothetical flowshop, composed of five machines, which process one type of product [12]. The first machine is supplied directly by the warehouse storage, so that each time an operation is completed, a new job is assumed to be ready to process on this machine. Each machine has an upstream buffer, for job awaiting service and a downstream buffer, for jobs waiting for transportation, which are both assumed to be infinite. There is only one transporter resource for the entire flowshop. Hence, in order to avoid to waste time, the parts are not transported one by one, but are grouped in transport lots, which have different sizes from one machine to another. If the transport lots are too large, the next machine will set idle too long. The transporter is allocated in real time to the machines which have enough parts in their

downstream buffer to be transported to the next machine. Several strategies can be used to allocate dynamically the transporter (Cyclic strategy, Longest waiting time, Largest number in queue, Shortest processing time, Longest processing time). The problem to solve is therefore to determine both the four transport lot sizes and the transporter allocation strategy, so as to minimize the makespan, i. e., the total time needed to complete in this example 500 products. It has to be noted that the decision variables are not independent in this problem. In particular, certain lots sizes may yield good results with a given transporter allocation strategy and bad results with another. Consequently, the best value of each variable cannot be searched sequentially.

The following figure shows the reduction of the makespan during an optimization research with a population size of 30 individuals, a mutation probability and a crossover probability of 0.5, a migration rate of 0.3 and a maximum number of generations without improvement of 5 for each LEA. Single-point crossover and uniform mutation are used. For all values of evolutionary parameters we have tested, the same best value was reached.
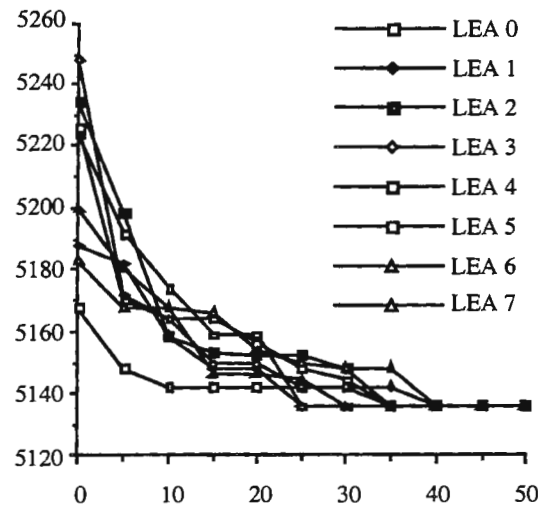
Figure 4: convergence of the objective function

Figure 4 shows that at certain points, migrations have occurred and emigrant solutions have directly contributed to the improvement of local populations and hence to the global convergence.

4.2 Designing of a Flexible Manufacturing System (FMS)

The second problem to address was the optimization of a FMS, which produces products A and B, and assembles those products to give a product C. There are several parameters and options to be determined for designing this workshop: how many stations are needed in the manufacturing cell producing A? Products B can be processed in two ways: with a flexible manufacturing center or with a lathe and a milling machine ; which option should be chosen ? If the flexible manufacturing option center is selected, how many flexible manufacturing centers are necessary ? If a lathe and a milling machine are selected, how many stations (a station is a lathe+a milling machine) are necessary ? and also which storage size between each machine should be chosen ? How many assembling stations are necessary ? Which transportation system should be chosen: conveyor or AGVs ? If the option conveyor is selected: which speed and which length are necessary for each one? If the option AGVs is selected: how many AGVs are necessary: one or two? if one AGV is selected: which lot size is necessary ? If two AGVs are selected, what is the path of each AGV ? If the path is the same for both AGVs then what is the lot size of AGVs else what is the lot size of each AGV ? The following is the representation of a potential solution of this FMS.
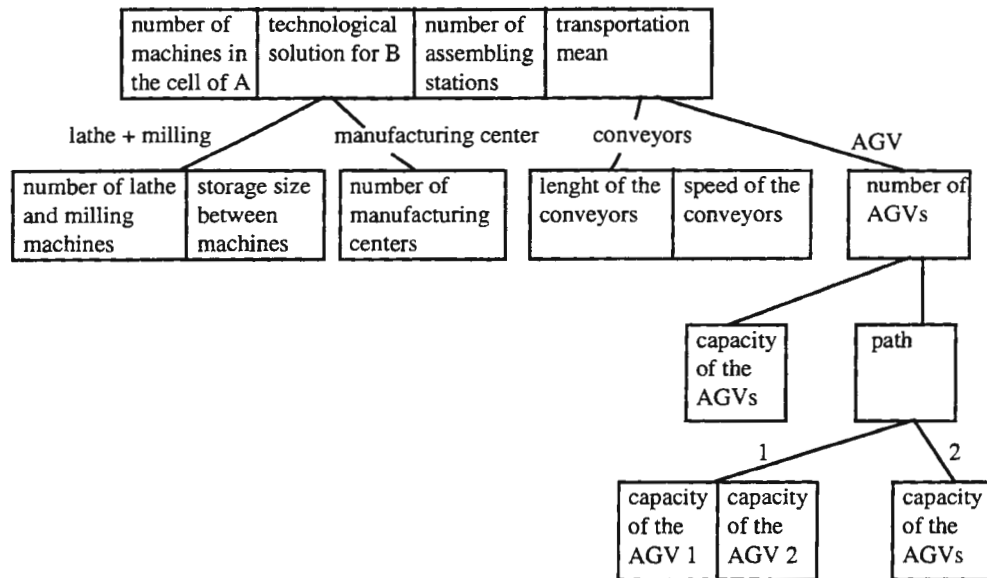
Figure 5: example of design options in a Flexible Manufacturing Systems.

With the worth set of evolutionary parameters tested (population size 25, mutation probability of 0.2 and recombinaison probability of 0.3), the best solution found was obtained in 38 generations.

4.3 A Kanban System

The third problem was to configure a multiproduct Kanban system [13]. This system must be able to respond quickly to manufacturing demands with minimum average WIPs. The system studied was composed of four machines in line, that can process three types of product. The tools switchover times was the same everywhere on the line and for each type of product. Each machine, for each type of product, has a specific and constant time . The transport times of full container and withdrawal kanbans was specific to each section of the line and are the same for each type of product. Three sequencing rules have been modelled: FCFS, SPT and SPT/Late. SPT/Late can be described as follows: if there is no product whose waiting time in front of the concerned machine is greater than a given value, then SPT is used, else the first product, in the order of the arrivals, having a waiting time higher than this given time, is treated. The aim is to avoid that a product remains stuck for a long time because of a high treatment time. The problem to solve was to determine values for the number of kanban between machines, the size of containers associated to each product, the sequencing rule followed by each machine and the size of the safety storages to optimize a production cost including backordered demands, number of kanbans and average number of work in progress. The best configuration found was obtained in 80 generations, with a population of 30, a mutation probability of 0.5 and a recombination probability of 0.8.

5. CONCLUSION

In this paper we have proposed a distributed simulation optimization tool based on evolutionary algorithms. The implementation of the distributed algorithms and communications between each other were managed effectively using PVM. This approach is able to take into account simultaneously parameters (quantitative and qualitative) and design options often encountered in manufacturing systems, thanks to a new data structure and new evolutionary operators.

The proposed method benefits from the powerful capabilities of evolutionary algorithms in optimum seeking and the use of several computers in parallel allows both the computing time to be drastically reduced, which is essential when simulation experiments are used to evaluate solutions, and the regions of the search space explored to be enlarged, giving therefore a better chance to find the optimum solution.

The   benefits of the approach presented in this paper were illustrated through three example: a flexible manufacturing system including design options, a flowshop system and a Kanban system. Meanwhile this approach should be useful to solve a wide range of other simulation optimization problems in other application areas.

6. REFERENCES

[1]   Aversa , R., Mazzocca, N., Villano, U.,1996, Design of a simulator of heterogeneous computing enviroenments, Simulation practice and theory, 4 (2-3), pp 97-117.

[2]   Azadivar, F., 1992, A tutorial on simulation optimization, *Proceedings of the 1992 Winter Simulation Conference,* 198 - 204.

[3]   Back, T., 1996, *Evolutionary algorithms in Theory and Practice* (Oxford University Press).

[4]   Bäck, T., and Schwefel, H.P., 1993, An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation* 1(1), pp1-23.

[5]   Braun, H., 1991, On solving travelling salesman problems by genetic algorithms. In H. P. Schwefel and R. Männer, editors, *Parallel problem Solving from Nature - Proceedings of 1st Workshop,* PPSN 1, Vol. 496 of *Lecture Notes in Computer Science,* pp 129-133, Dortmund, Germany: Springer-Verlag, Berlin, Germany.

[6]   Cantù-Paz, E., 1995, A summary of research on parallel genetic algorithms, *IlliGAL Report N° 95007,* University of Illinois at Urbana-Champaign, july 1995.

[7]   Cohon, J., Martin, W., and Richards, D., 1991, Genetic algorithms and punctuated equilibria in VLSI. in H.-P Schwefel and R; Männer, editors, Parallel problem Solving from nature - PPSN I, Vol. 496 of Lecture Notes in Computer Science, pp 134-144, Berlin, Germany: Springer-Verlag.

[8]   Cosnard, M., Trystram, D., 1993, Algorithmes et architectures parallèles (InterEditions, Paris).

[9]   Gaury, E.G.A. , Kleijnen, J.P.C., Pierreval, H., Configuring a pull production control strategy through a generic model, *Proceedings of the conference on management and control of production and logistics,* pp 223-228, Campinas, Brazil, September.

[10]   Jog, P., Suh, J., Van Gucht, D., 1991, Parallel genetic algorithms applied to the travelling salesman problem, *SIAM journal on optimization,* Vol. 1, n° 4, pp 515-529.

[11]   Michalewicz, Z., 1992, *Genetic Algorithms + Data structures = Evolution Programs* (Springer - Verlag).

[12]   Paris, J.L., Pierreval, H., Tautou, L., 1996, Distributed evolutionary simulation optimization and application on a transport problem in a manufacturing flowshop, proceedings of the 8th European Simulation Symposium Vol 1, pp 310-313, ISBN 1-56555-099-4, Genoa, Italy.

[13]   Paris, J.L., Pierreval, H., 1997, Configuration of multiproduct Kanban systems using a distributed evolutionary algorithm, *Proceedings of the conference on management and control of production and logistics,* pp 192-197, Campinas, Brazil, September.

[14]   Pierreval, H., Tautou, L., 1996, Using evolutionary algorithms and simulation for the optimization of manufacturing systems, IIE Transactions, 29 (3), 181-189.

[15]   Reinefeld, A., 1996, Parallel search in discret optimization problems, *Simulation practice and theory,* 4 (2-3), pp 169-188.

[16]   Salzmann, C., Breiteneicker, F., 1995, Genetic algorithms for discret event simulation, *Proceedings of Eurosim'95,* pp 213-218, Vienna, Austria, 11-15 sept. (Edition Breitenecker et Husinsky).

[17]   Schuster, G., Breitenecker, F., 1995, Coupling simulators with the model interconnection concept and PVM, *Proceedings of Eurosim'95,* pp 321-326, Vienna, Austria, 11-15 sept. (Edition Breitenecker et Husinsky).

[18]   Tautou, L., Paris, J.L., Pierreval, H., 1997, New evolutionary simulation optimization approach for the design of manufacturing systems, *Proceedings of the world congress on systems simulation* (2nd joint conference of international simulation societies), pp 100-104, Singapore.

Development of Emergency Relief Simulator for Near Critical Fluids

Juha-Pekka Pokki [1], Markku Hurme [1], Juhani Aittamaa [2]

1) Department of Chemical Technology, Helsinki University of Technology, P.O.B 6100, FIN-02015 HUT, Finland,
jpokki@cc.hut.fi

2) Neste Oy, Engineering, P.O.B 310, FIN-06101 Porvoo, Finland

Abstract

There are conservative and widely accepted methods to design the pressure relieving devices for processes that operate far away from the thermodynamic critical point but there are no simple methods for processes that operate in the vicinity of the critical point. To design pressure relieving devices and to investigate the applicability of the API 520 procedure for near critical point applications a simulator with rigorous thermodynamics was developed. The simulator operates in two modes. In the first mode the simulator provides the orifice area of the relieving device that can handle the required flow and in the second mode the performance of the process is simulated using the predetermined pressure relieving device. The proposed method is compared with the API 520 procedure and it is found that the API 520 suggests larger cross sectional area for the pressure relieving device than the proposed method. Based on this work the API 520 seems to be a safe and a conservative method in the vicinity of the critical point.

Introduction

The design of a safe operation of a process requires the cause and consequence analysis of potential abnormal events of the process. There are many abnormal events that may disturb the process and cause a dangerous situation. These include fire, reduction of cooling, power failure, blockage in some process equipment of piping, malfunctioning of control system, feed at abnormal rate, etc. The consequence of these might be the increase of pressure in some process equipment. The equipment are designed to tolerate certain maximum pressure which may not be exceeded. The pressure higher than maximum may cause the equipment to burst which must be prevented with pressure relieving devices. These work independently of the control system.

The pressure in the process equipment depends on the vapour pressure of the material and its temperature. If the temperature increases too high the control system may try to cool the material. If this is not possible or adequate some material must be removed from the vessel. The pressure relieving device opens a particular route for the material to escape to a safe place. The design of the pressure relieving device and the outlet line depends on the process. The current and widely accepted procedure to design pressure relieving devices is API 520. [1] It uses the concept of ideal gas and thus can not account non-ideal behaviour, supercritical conditions or multiphase flow.

There are many types and sizes of pressure relieving devices that have different characteristic opening curves. The orifice area is the key parameter for the flow of material. The orifice area large enough to pass the required flow of material must be selected. On the other hand too large orifice area may cause the pressure relieving device to chatter and flutter, i.e. to work incorrectly. Sometimes two or more pressure relieving devices are designed to protect the equipment. The smallest handles minor emergency outflows.

Description of the simulator

The simulation of a relieving system requires a considerable amount of initial data. Data can be divided in two parts, process data and equipment data. The most important process data consist of thermodynamic properties of components and reaction rates of components. All these are function of temperature, pressure and concentration. This simulator includes a large databank and collection of physical property calculation procedures. Because the reaction kinetics is usually determined experimentally the model for reaction kinetics

must be supplied by user. The equipment data includes dimensions and characteristic curve of pressure relieving device. The dimensions needed for safety valve and rupture disk are orifice area and set pressure, for safety valve the pressure where safety valve is totally open and for rupture disk if it is burst or not in the beginning of the simulation. In addition, the phase of the outflow (liquid, vapour or mixed), the design method (proposed or API 520) and operation mode (design or rating) must be selected. By defining the phase of outflow the geometric shape of the vessel is not needed because in preliminary safety analysis the shape of the vessel is not necessarily defined. Because dynamic simulation is an initial value problem the variables need initialisation. The most convenient set of initial values of the variables is temperature, pressure and total composition because the designed operation condition of the process is available. At present the simulator is dimensioned for four vessels, sixteen valves and eighteen flows.

Although thermodynamic properties could be determined by interpolation and the phase could be assumed the simulator calculates the thermodynamic properties and determines the phase of the material rigorously at every time step. At certain conditions material can separate in two distinct phases with different physical properties. The supercritical phase has both vapour-like and liquid-like properties. The determination of physical properties very close to critical point is a challenging task. There are convergence problems because the properties of the coexisting phases become identical at the critical point but the flash procedure requires detectable difference between coexisting phases. The calculation of thermodynamic and physical properties is based on principles of the flash routine presented by Whitson and Michelsen. [2] Their procedure is reliable and provides the state of the fluid and its relative amounts in phases. The thermodynamic model in this work is a cubic equation of state, called Soave-Redlich-Kwong. [3]

The mass of component m_i and internal energy of material U can be computed from general thermodynamics. The relation between internal energy U and enthalpy H is

$$U = H - pV \tag{1}$$

where p is system pressure and V is system volume. The material inventory and internal energy of the vessel depends on flows entering and leaving the vessel. The mass and energy balances are given by equations (2) and (3)

$$\frac{\partial m_i}{\partial t} = \dot{m}_{i,in} - \dot{m}_{i,out} \tag{2}$$

$$\frac{\partial U}{\partial t} = \dot{H}_{in} - \dot{H}_{out} + \dot{Q}_{HE} + \dot{Q}_{FR} - \dot{Q}_{CL} + \dot{Q}_{MX} \tag{3}$$

where m_i is mass of individual component, $\dot{Q}_{HE}$ is net heat flow caused by heating, $\dot{Q}_{FR}$ is net heat flow caused by fire, $\dot{Q}_{CL}$ is net heat flow caused by cooling and $\dot{Q}_{MX}$ is net heat flow caused by mixing. The values of temperature and pressure at time t+Δt can be computed from (4) and (5)

$$m^{t+\Delta t} = \sum_1^{NC} m_i^{t+\Delta t} = \sum_1^{NC} \left(m_i^t + \frac{\partial m_i}{\partial t} \Delta t \right) \tag{4}$$

$$U^{t+\Delta t} = U^t + \frac{\partial U}{\partial t} \Delta t \tag{5}$$

The set of two variables, temperature and pressure, is solved with Newton-Raphson procedure. The convergence is fast because the old values at time t serve as a good initial guess for the iteration. Solution of temperature and pressure is possible because the masses of components define the total molar composition.

As a emergency relief starts the pressure difference causes the material to flow out the vessel. The friction of the flowing material causes a drop in pressure and the phase of the material changes. If the material is on the other hand in liquid phase in the vessel then it might vaporise at the outlet line of the pressure relieving device. The phase change in the outlet line depends on the pressure drop. The drop in pressure is dependent on the phase, speed and acceleration of the material, the dimensions of pipe and the physical and transport properties of the material. In this work the pressure drop in pipe is omitted.

The maximum speed of flowing material in duct is the speed of sound in the flowing material. That flow is called critical flow. If the speed of material is less than the speed of sound the flow is called subcritical flow. The speed of sound can be computed from the thermodynamic model. The speed of sound according to Malanowski [4] is presented in equation (6) where V is system volume, ρ is density of the material and the derivative is the derivative of pressure versus volume at constant entropy. This derivative is difficult to evaluate. A more suitable expression for calculations is presented in equation (7)

$$W_{sound} = \sqrt{\frac{V}{\rho}\left(\frac{\partial p}{\partial V}\right)_S} \qquad (6) \qquad W_{sound} = \sqrt{-\frac{v^2 c_p}{M c_v}\left(\frac{\partial p}{\partial v}\right)_T} \qquad (7)$$

where v is molar volume, c_p and c_v are molar heat capacities at constant pressure and temperature, M is molecular weight and the derivative of pressure versus molar volume is determined at constant temperature. All these terms can be computed from the suitable cubic equation of state. This approach is straightforward for single phase material but the mixture of vapour and liquid is more complex to handle since the speeds of the flowing vapour and liquid may differ. It is possible to assume that speeds of phases are equal, vapour is completely mixed with liquid and the physical property of the mixture are a weighed average of the properties of the liquid and vapour phase.

Designing of the pressure relieving device

The pressure relieving devices can be divided into three main groups. They are spring loaded pressure relieving devices, rupture disks and pilot operated pressure relief valves. The spring loaded pressure relief device opens and closes proportionally to the pressure difference. The rupture disk device bursts only once when the pressure difference exceeds the defined value and stays open after the burst. The pilot operated pressure relief valve is a spring loaded pressure relief valve in which the main valve is combined with and controlled by an auxiliary pressure relief valve. Pilot operated pressure relief valves are not studied in this work. The design of the pressure relieving device can be done in two ways. One way is to find out how much material must be removed from the vessel and calculate the required area of pressure relieving device The other is to specify first the pressure relieving device, simulate the process and check if the pressure stays in a safe level. By combining these two ways in the simulator an effective design tool is obtained.

Design mode

In design mode the performance of the system is specified and the system is designed to meet this performance. The opening pressure is called set pressure. In design mode the set pressure is used as a maximum pressure of the vessel. When the pressure of the vessel is tending to exceed set pressure certain amount of material and energy is removed from the reactor to maintain pressure of the vessel at set pressure. The state of the vessel is calculated with equations (4) and (5). The condition where set pressure p_{set} equals pressure of the vessel p is presented in equation (8)

$$f_{def} = p - p_{set} = 0 \qquad (8)$$

The average outflow of material and energy at an interval of time is equal to the amount of material and energy that must be removed to satisfy the equation (8). The formulation is presented in equations (9) and (10)

$$m_{pset} = m_p - \dot{m}_{out} \Delta t \qquad (9)$$

$$U_{TOT,pset} = U_{TOT,p} - \dot{H}_{out} \Delta t \qquad (10)$$

where m_{pset} is mass of material at set pressure, m_p is mass of material in the vessel, $U_{TOT,pset}$ is internal energy of material at set pressure and $U_{TOT,p}$ is internal energy of material in the vessel. Because outflow of energy is related to the mass of material the outflow of material is solved with Newton-Raphson routine. The iteration is started with a small value of mass flow. Once the convergence is achieved the value of the outflow is stored to be used as initial value at next time step. The proposed method for calculating the area of the valve is equation (11).

$$A = \frac{\overset{\bullet}{n}\, v}{w_{sound}} \qquad\qquad (11)$$

where $\overset{\bullet}{n}$ is molar flow, v is molar volume and w_{sound} is speed of sound. This method requires that pressure difference causes critical flow.

Rating mode

In rating mode the system is specified and its performance is calculated. In this mode the characteristic curve and all dimensions of pressure relieving device is required to calculate outflow, temperature and pressure of the vessel. The rating mode provides information how the performance of the process is affected by the pressure relieving device selected. If critical flow is assumed the mass flow can be calculated from equation (12) because the area as the function of pressure difference is known.

$$\overset{\bullet}{m}_{TOT} = A\rho w_{sound} \qquad\qquad (12)$$

where A is cross sectional area of the valve and ρ is density of the fluid.

Example

Two vessels are protected with pressure relief valves. The system is presented in Figure 1.
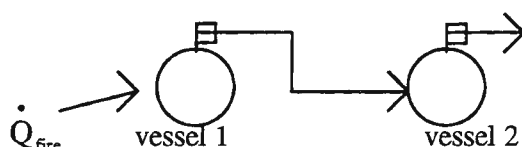
Figure 1. *The system of two vessels (circles) protected with pressure relieving devices (boxes).*

The outflow of the first 20 m³ vessel enters the second vessel and the outflow of the second 20 m³ vessel enters atmosphere. The uninsulated vessel number 1 is exposed to fire for 450 seconds. The heat input is defined according to API 520. The total area exposed to fire is assumed to be equal to the total area of the 20 m³ sphere. The total composition of the mixture in both vessels is 32.8 mol-% propane, 34.0 mol-% n-butane and 33.2 mol-% n-pentane. The initial temperature of the first vessel enveloped by flames is 423.15 K and pressure is 4.5 MPa abs. The mixture critical point tabulated in literature [5] is 428.8 K and 4.188 MPa abs, the initial condition is near the critical point. The initial temperature of the second vessel is 393.15 K and pressure is 0.11 MPa abs. The phase envelope is presented in Figure 7. The initial condition of the second vessel is in vapour region. The set pressure of the relief valve in the first vessel is 5.0 MPa abs and the second 0.3 MPa abs.

The simulation of the design mode is presented in Figures 2 and 3. The required areas are presented as a function of time. Both the proposed method and the API 520 procedure use the same rigorous thermodynamic properties computed at every time step. There are two areas based on API 520 in the Figure 2 because near the critical point the material can be defined either liquid or vapour but only one area based on API 520 in the Figure 3 because the material in vessel 2 is clearly gas. The area based on API 520 is about twice as large as the proposed area. The maximum proposed area of the pressure relieving device of the first vessel is close to the smallest commercial "D" (71 mm²) orifice safety valve and the maximum proposed area of the pressure relieving device of the second vessel is approximately "M" (2322 mm²) orifice safety valve.
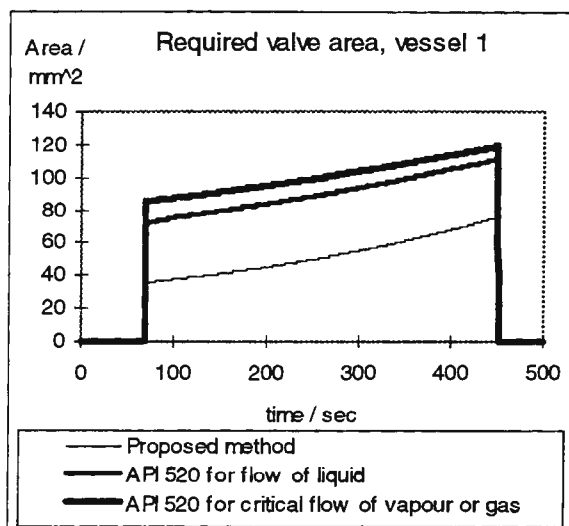
Figure 2. *In the vicinity of the critical point three possible areas are presented*
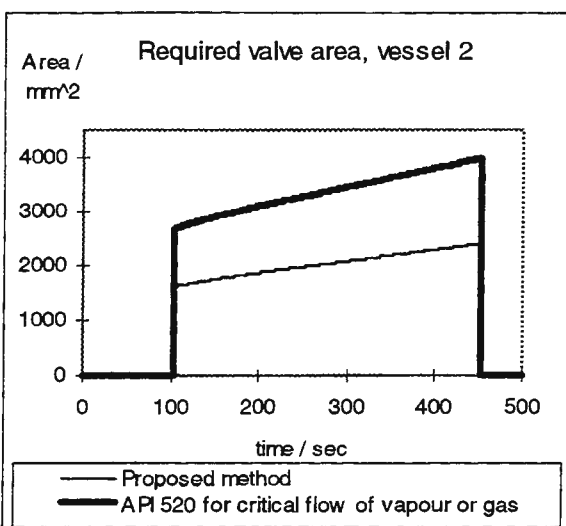
Figure 3. *For the gas filled vessel two areas are presented*

In rating mode the commercial valve areas, "D" and "M", based on proposed method are selected. By simulation the following performance of the system is obtained. The pressure and temperature profiles of the vessels are in Figures 4 and 5.
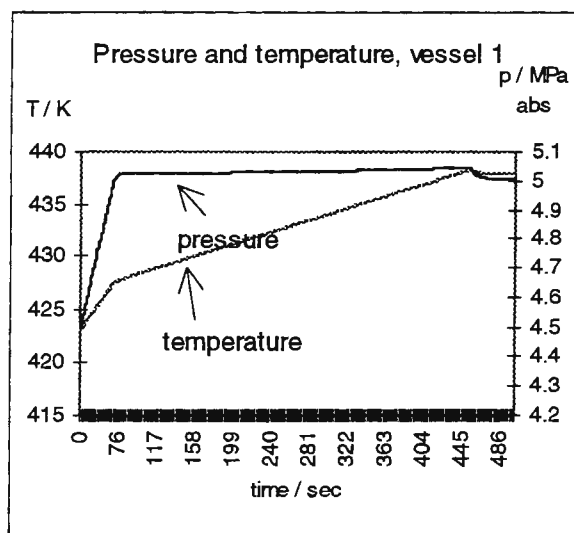

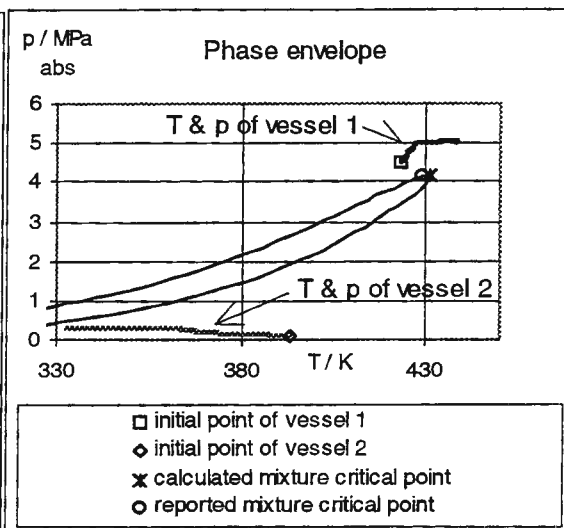**Figure 4.** *Performance of the first vessel with specified safety valve (71 mm²)*
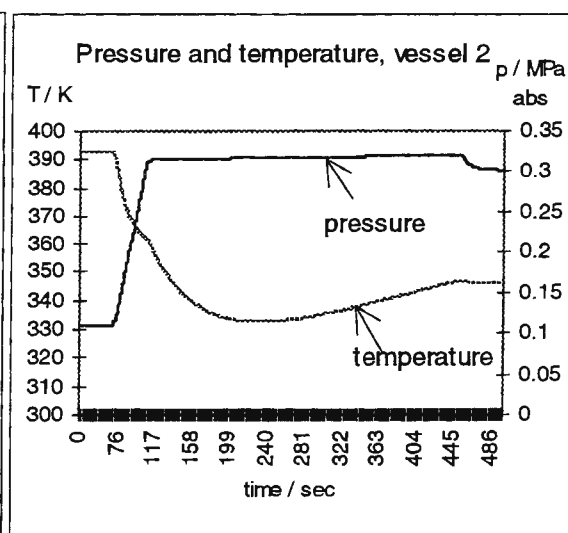
Figure 5. *Performance of the second vessel with specified safety valve (2322 mm²)*

It can be noticed from the Figures 4 and 5 that in the rating mode pressure stays near the set pressure in both vessels. This is an indication that the maximum value of the area of design mode should be selected. This example is valid for this kind of fire. If the fire last longer the safety relieving device must be redesigned. The areas based on API are larger which provides larger safety margin than the proposed method.
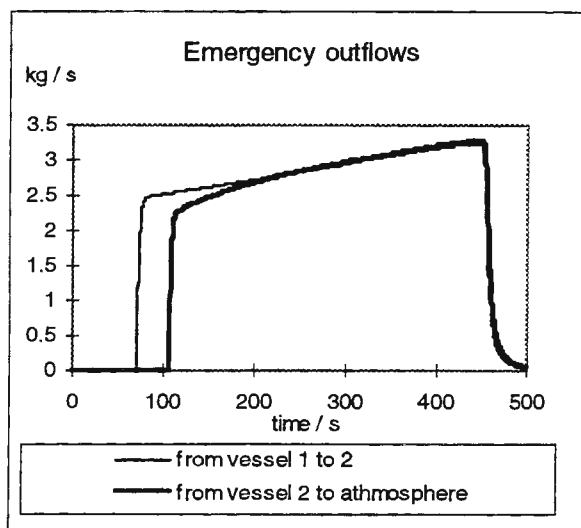
Figure 6. *Emergency outflows caused by fire*   **Figure 7.** *Phase envelope of the mixture studied*

The emergency outflows are presented in Figure 6. The accumulation of material at the second vessel delays the opening of the safety valve of the second vessel. As the fire exhausts the mass flows stop very rapidly and pressure stays at set pressure because safety valves are selected. If the pressure relieving devices were rupture disks the flows would continue until the ambient pressure were reached. The paths of the pressure and temperature of the vessels are presented in Figure 7. The phase envelope is valid for the whole time interval because the composition stays unchanged.

Conclusion

An effective tool for designing a pressure relieving device is developed. The thermodynamic model used in the simulator is applicable to the hydrocarbon mixture of light components. The simulator also provides a comparison of the proposed area to the area based on the API 520 procedure. It is found that the API 520 procedure provides larger orifice area of the pressure relieving device than the proposed method using rigorous thermodynamics. Therefore the method of API 520 seems to be a safe and conservative method in the vicinity of the critical point.

Acknowledgements

References

[1]. American Petroleum Institute, "Sizing, Selection, and Installation of Pressure-Relieving Devices in Refineries," API Recommended Practice 520, Part I - Sizing and Selection, 6th ed., API, Washington, DC, March 1993

[2]. Whitson, C.H. & Michelsen, M.L., "The negative flash", Fluid Phase Equilibria, vol 53, 1989, p. 51-71

[3]. Soave, G., "Equilibrium constants from a modified Redlich-Kwong equation of state, Chemical Engineering Science, vol 27, 1972, p. 1197-1203

[4]. Malanowski, S. & Anderko, A., "Modelling Phase Equilibria", John Wiley & Sons, USA, 1992, p. 39

[5]. Sadus, R.J., "High Pressure Phase Behaviour of Multicomponent Fluid Mixtures", Elsevier, Netherlands, 1992, p. 269

A STEADY STATE RATE BASED REACTIVE DISTILLATION MODEL

Kaj Jakobsson[1], Antti Pyhälahti[1], Juhani Aittamaa[2], Peter Banik[2] , Matti Koskinen[2]

[1]Department of Chemical Engineering, Helsinki University of Technology, PO 6100 FIN-02015, HUT Kaj.Jakobsson@hut.fi
[2]Neste Oy, Engineering, P.O.B. 310, FIN-06101 Porvoo, Finland

Introduction

Reactive distillation (RD) is a combination of a unit process, chemical reaction, and distillation. This type of processes which combines reaction and separation seem to provide promising process alternatives today. The benefits obtained in applying reactive distillation are that the combination of reaction and separation in a single unit can reduce the capital and operating costs considerably and in the case of a reversible reaction the yield of the desired product can be increased.

The applicability the RD process is highly dependent on the properties of the chemical system at hand. A good candidate for reactive distillation is a system with an unfavourable reaction equilibrium and a significant heat of reaction, where the feasible distillation and the reaction temperature ranges overlap.

Esters have been produced using reactive distillation for decades. The most important products are methyl acetate, ethyl acetate and butyl acetate. Worth of mentioning is the recently developed methyl acetate process. In this process high purity methyl acetate is obtained with a single RD column which replaces four separate columns of the conventional process. The most important application of reactive distillation today is the production of ethers used as modern gasoline components. The ethers are methyl tert-butyl ether (MTBE), ethyl tert-butyl ether (ETBE), tert-amyl ether (TAME), and tert-amyl ethyl ether (TAEE). They increase the oxygen content of the gasoline, the octane number is improved, vapour pressure of the gasoline is decreased and the olefin content of the gasoline is reduced. Gasoline containing oxygenates burns cleaner and the reduction of olefin content decreases the amount of ozone produced.

In the Brite Euram III Reactive Distillation -project a rate based steady state model is under development. This paper gives an overview of the main features of the model, describes the solution strategy used, and gives some simulation examples demonstrating the properties of the solution strategy of the model.

Modelling aspects

The number of published models describing reactive distillation is considerable. Pyhälahti (1996) gives a comprehensive description of the models published in the open literature.

The traditional way of modelling a distillation column is the so called equilibrium stage model. In actual operation the stages rarely operate in equilibrium. The usual way of dealing with the departure from the equilibrium is to use stage efficiencies. However, it is not easy to determine these efficiencies.

Another approach is to take heat and mass transfer effects, and multicomponent interaction between the components directly into account in the stage model. In this work the starting point of the modelling is the ratebased model for distillation, see e.g. Taylor and Krishna (1993). The adaptation of this model for reactive distillation has been done among others by Zheng and Xu (1992) and Sundmacher and Hoffmann (1996).

The program contains several hydrodynamic models of the plates. The most important of them are the mixed liquid-mixed vapour and mixed liquid-vapour plug flow models. The implementation of these models is based on the theory presented by Taylor and Krishna (1993) and Hung (1991) respectively. More complex plate models like mixed pool model, and rigorous eddy diffusion model of the plate have also been implemented.

There are several models available for the reaction rates: homogeneous reaction in the liquid bulk, homogeneous reaction in the liquid film based on the theory presented by Kenig and Gorak (1996) and a heterogeneous catalytic liquid phase reaction which may include the effect of the mass transfer resistances in the catalyst pellets based on the theory by Sundmacher and Hoffmann (1996).

The program contains a library of binary mass transfer, pressure drop and liquid hold-up correlations for packed and plate columns available from the open literature.

The large number of additional physical properties needed are computed with standard methods. The implementation of these models is by the IK-CAPE -project. This project has been started by the German chemical industry to standardise the computation of the physical properties in modelling of chemical industry projects.

The solution strategy

For the strategy for solving the highly non-linear set of equations rising from the nonequilibrium model a hybrid method combining relaxation and Newton's method has been developed. Here the relaxation method is understood as a method were the steady state solution is found trough the change of the column state with time. The relaxation method was originally developed by Rose et al. (1958) later modified by Ball (1961). Others developing the method have been among others Ishikawa and Hirata (1972), Jelínek et al. (1973), Jelínek and Hlvácek (1976), Kethcum (1979) presented a combined relaxation-Newton method. Gani and Cameron (1989) have presented the extension of dynamic models of distillation to steady state simulation and Kooijman and Taylor (1995) reports also of the usage of their dynamic nonequilibrium model for finding steady states.

The relaxation method applied follows the ideas of Sundmacher and Hoffmann (1996). The unsteady state form of the equations is derived by modifying the component mass balances and energy balances around the bulk phases to include terms of change in time of the correspondent variables (bulk compositions, liquid and vapour molar enthalpies). The combination of the enthalpy balances to the equation group solved with the relaxation method can cause problems. According to the experience of the authors this happens if the temperatures are used as differential variables. However, it seems that if the enthalpies are used instead of them as differential variables the stability of the problem is better.

This treatment of the equations gives a DAE -system for one stage of the form of equation (1).

$$
\begin{bmatrix}
0 \\
& U_j^V \\
& & U_j^V \\
& & & 0 \\
& & & & 0 \\
& & & & & 0 \\
& & & & & & 0 \\
& & & & & & & 0 \\
& & & & & & & & U_j^L \\
& & & & & & & & & U_j^I \\
& & & & & & & & & & 0 \\
& & & & & & & & & & & 0
\end{bmatrix}
\frac{d}{dt}
\begin{bmatrix}
V_j \\
(Y_{ij}) \\
H_j^V \\
(Y_{ij}^I) \\
Y_{NC,j}^I \\
(X_{ij}^I) \\
T_j^I \\
L_j \\
(X_{ij}) \\
H_j^L \\
(N_{ij}) \\
N_{nc,j}
\end{bmatrix}
=
\begin{bmatrix}
M_{ij}^V \\
(M_{ij}^V) \\
E_j^V \\
(R_{ij}^V) \\
S_j^V \\
(Q_{ij}^I) \\
E_j^I \\
M_{ij}^L \\
(M_{ij}^L) \\
E_j^L \\
(R_{ij}^L) \\
S_j^L
\end{bmatrix}
\qquad (1)
$$

where the vectors (Y_{ij}), (X_{ij}), (X_{ij}^I), (M_{ij}^V), (M_{ij}^L), and (Q_{ij}^I) have the dimension NC and vectors (Y_{ij}), (R_{ij}^V), (N_{ij}), and (R_{ij}^L) are of dimension NC-1. The vapour and liquid inertia terms U_j^V and U_j^L are taken to be constant.

This DAE -system is solved with an already existing DAE-solver LIMEX from Kondrad Zuse Zentrum.

For the Newton method the independent variables and equations are those presented by Taylor and Krishna (1993). The implementation of the Newton's method of solution is a basic one. The Jacobian matrix has the block tridiagonal structure. This makes it possible to solve the linearised subset of the equations by the Thomas algorithm. The computation of the blocks of the Jacobian is done numerically and the determination of the step length in the Newton iteration is done by quadratic interpolation.

The hybrid method was formed by combining the relaxation method with the Newton's method. The flowchart of the algorithm is given in Fig 1.

The algorithm starts by the initialisation of the DAE-system. Then the Limex integrator integrates the problem to a pre-set time t+Δt and stores the independent variables. Using the values obtained from the integrator as a starting point the Newton solver tries to solve the problem. If this is successful, a report is written and the program will terminate. In the case that the Newton solver returns an error message the program reads the solution from the previous integration and returns to integrator and continues the integration. A loop counter controls the maximum times the relaxation/Newton -algorithm is executed.

It seems important to restrict the maximum and minimum values of the variables during the Newton iteration with absolute numbers to avoid the case that the physical property, mass transfer and hydraulic routines get wrong values, for example negative flows or compositions. Such values can make these routines to fail and end the simulation in an error. It is possible to give absolute limits to variables and it is also possible to give the maximum percentage of the change of the variable towards the boundary to restrict the change of variable values for the Newton iteration.
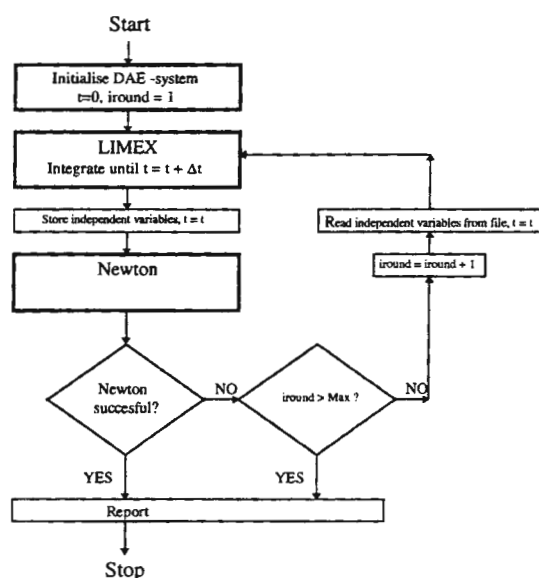
Fig 1. The flowchart of the hybrid algorithm.

Numerical examples

Steady states of four examples of MTBE production were simulated. This system is a good example of possibilities of applying RD. MTBE is currently produced in conventional fixed bed reactors followed by separation section for product purification and recycling unreacted feed methanol after the wash step. Typically the conversion of the isobutene is limited to about 90% because of the chemical equilibrium.

Yield can be increased somewhat with additional reactors, but with reactive distillation almost complete conversion is possible with a smaller number of process units.

The significant qualitative difference of the examples is that in the two first Fig 2. and Fig 3. the amount MTBE product in the reboiler (segment 1) is low. In the two latter simulations Fig 4. and Fig 5. the composition of the MTBE was somewhat higher. According to the experience of the authors Newton's method and its relatives works well when the concentration of the MTBE in product is low. However, finding the solution for the higher MTBE product level requires often tedious nursing of the solution. Similar experiences are also reported by Zheng and Xu (1992).

Two of the runs was done using models for random packed catalytic rings (GPP-rings) described by Sundmacher and Hoffmann (1996) and two with structured catalyst packing filled with ion-exchange resin.

The hydrodynamic model applied was the mixed liquid mixed vapour model. The kinetic model was taken from Rehfinger and Hoffmann (1990) as well as the UNIQUAC interaction parameters for the VLE calculation. The binary diffusivities were calculated in the liquid phase with the method of Tyn and Calus and in the vapour phase with the method of Fuller see e.g. Reid et al. (1987). The vapour and liquid binary mass transfer correlations were calculated for the inert packing and the GPP-rings by the correlation of Onda et al. (1968) and for the catalytic structured packing section the correlations of Rocha et al. (1996) were chosen. The correlations and parameters for many additional physical properties are taken from DIPPR correlations or among methods presented in Reid et al. (1987).

First the problems were tried using the Newton solver independently and the solution was found easily for examples 1 and 2 (Table 1). The two latter examples failed with the Newton solver. However, with a simulation of 15 s (model time) and with a very small inertia terms on the segments the relaxation method could deliver a good starting point for the Newton's method (Table 2). The computer platform used was Pentium II, 200 MHz with 64 MB of RAM.

Table 1. Properties of the computation with the Newton's method

Run	Jacobian Evaluations	CPU-time s
1	9	153
2	8	125
3	Not converged	
4	Not converged	

Table 2. Properties of the computation of the Hybrid solver.

Hybrid solver						
Relaxation method					Newton's method	
Run	Inertia terms	time s	function evaluations	CPU time s	Jacobian Evaluation	CPU time s
3	.0001	15	about 2600	861	3	75
4	.0001	15	about 3200	1063	3	77
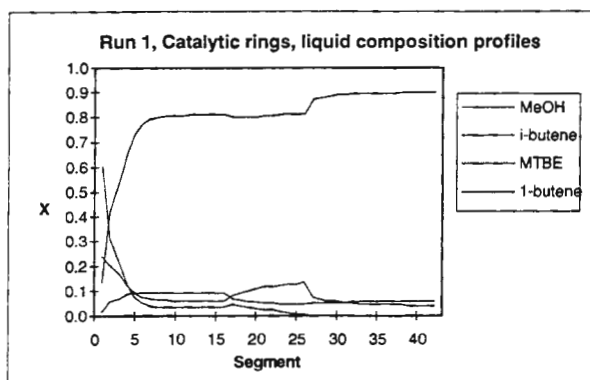
Fig 2. Liquid composition profile of run 1
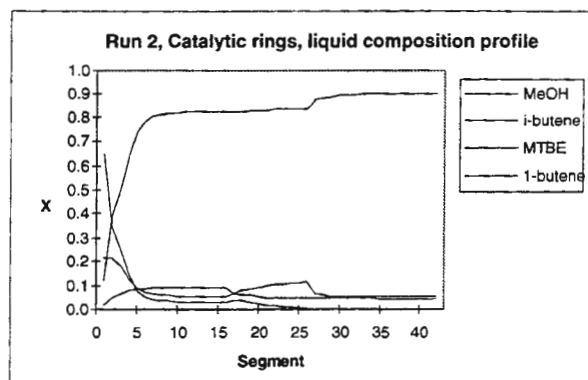
Fig 3. Liquid composition profile of run 2
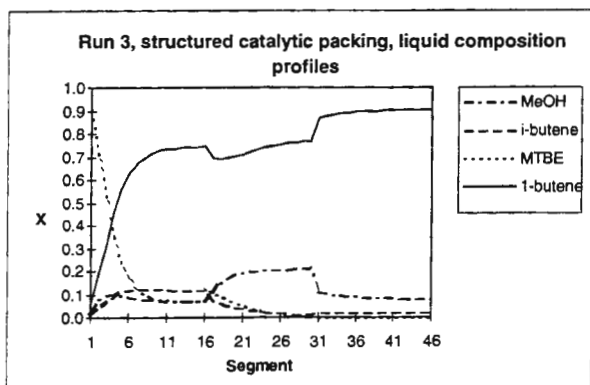
Fig 4. Liquid composition profile of run 3
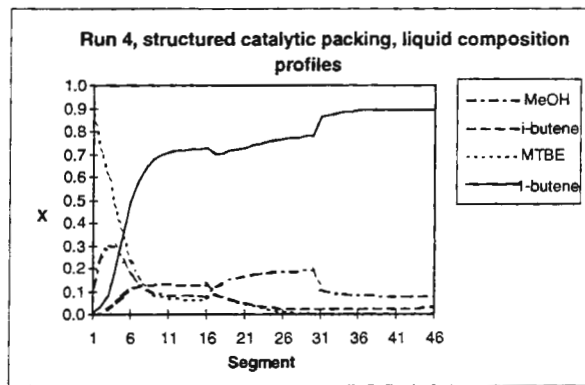
Fig 5. Liquid composition profile of run 4

Conclusions

A rate based steady state reactive distillation model has been developed. A large number of different mass transfer, reaction and hydrodynamic models has been implemented to form an advanced reactive distillation simulator.

For the method of solution a hybrid algorithm combining the best properties of the relaxation and Newton method has been developed. The relaxation method is used in the beginning of the computation to find a good starting point for the Newton's method. The final solution is found by the Newton's method which converges rapidly from a suitable starting point. The hybrid method performed well and has a broader domain of convergence than the Newton solver used alone.

Nomenclature

E_j^I residual of the enthalpy balance over the interface
E_j^L residual of the enthalpy balance around liquid phase
E_j^V residual of the energy balance around the vapour phase
H_j^L specific enthalpy liquid, J/mol
H_j^V specific enthalpy vapour, J/mol
L_j flow liquid bulk, mol/s
M_{ij}^L residual of the component mass balances around the liquid phase
M_{tj}^L residual of the total mass balance around the liquid phase

M_{ij}^V	residual of the component material balances around the vapour phase
M_{tj}^V	residual of the total mass balance around the vapour phase
N_{ij}	mass transfer rate, mol/s
P_j	pressure, Pa
P_j	residual of the hydraulic equation
Q_{ij}^I	equilibrium relations at interface
R_{ij}^L	residual of the mass transfer relations, liquid
R_{ij}^V	residual of the mass transfer relations for vapour
S_j^L	residual of the summation equation liquid
S_j^V	residual of the summation equation vapour
T_j^L	temperature liquid bulk, K
T_j^V	vapour bulk temperature, K
TI_{ij}	interface temperature, K
U_j^L	inertia term liquid, mol
U_j^V	inertia term vapour, mol
X_{ij}	concentration liquid bulk
XI_{ij}	concentration liquid interface
V_j	flow vapour bulk, mol/s
Y_{ij}	concentration vapour bulk
YI_{ij}	interface concentration vapour

Literature

Ball, W.E., 1961, 44th National meeting of A.I.Ch.E., New Orleans.

Gani, R., Cameron, I.T., 1989, Extension of dynamic models of distillation columns to steady-state simulation, *Comp.Chem.Engng*, **13**, 3,271-280.

Hung, J., 1991, *A second generation nonequilibrium model for computer simulation of multicomponent separation processes*, Ph.D.Thesis, Clarkson university, 190p.

Ishikava, T., Hirata, M., 1972, Extractive distillation calculations by modified relaxation method, *J.Chem..Engng.Japan*, **5**, 125-131.

Jelínek, J., Hlavácek, V., Kubícek, 1973, Calculation of multistage countercurrent separation processes-II. Multistage separation by relaxation procedure, *Chem.Engng.Sci.*, **28**, 1825-1832.

Jelínek, J., Hlavácek, V., 1976, Steady state countercurrent equilibrium stage separation with chemical reaction by relaxation method, *Chem.Engng.Commun.*, **2**, 79-83.

Ketchum, R.G., 1979, A combined relaxation-Newton method as a new global approach to the computation of thermal separation processes, *Chem.Eng.Sci.*, **34**, 387-395.

Kenig, E., Górak, A., 1995, A film model based approach for simulation of multicomponent reactive separation, *Chemical Engineering and Processing* , **34**, 97-103.

Kooijman, H.A., Taylor, R., (1995), A nonequilibrium model for dynamic simulation of tray distillation columns, *AIChE J.* **41**, 8, 1852-1863.

Onda, K., Takeuchi, H., Okumoto, Y., 1968, Mass transfer coefficients between gases and liquid in packed columns, *J.Chem.Eng.Japan* **1**, 1, 56..62.

Pyhälahti, A., 1996, *Reactive distillation in literature*, Plant Design Report Series No. 42, Helsinki University of Technology, 107 p.

Rehfinger, A., Hoffmann, U., 1990, Kinetic of methyl tertiary butyl ether liquid phase synthesis catalysed by ion exchange resin-I. intrinsic rate expression in liquid phase activities, *Chem.Eng.Sci.*, **45**, 6, 1605-1617.

Reid, R.C., Prausnitz, J.M., Poling, B.E., 1987, *The properties of gases and liquids*, 4 Ed., McGraw-Hill Book Company.

Rocha, J.A., Bravo, J.L., Fair, J.R., *Ind.Eng.Chem.res.*, **35** (1996) 1660-1667.

Rose, A., Sweeney, R.F., Schrodt, V.N., 1958, Continuous distillation calculations by relaxation method, Ind.Eng.Chem., **50**, 737-742.

Sundmacher, K., Hoffmann, U., 1996, Development of a new catalytic distillation process for fuel ethers via a detailed nonequilibrium model, *Chem. Eng. Sci.*, **51**, 10, 2359-2368.

Taylor, R., Krishna, R., 1993, *Multicomponent mass transfer*, John Wiley, New York, 578 p.

Zheng, Y., Xu, X., 1992, study on catalytic distillation process Part II. Simulation of catalytic distillation process, *Trans.Inst.Chem.Eng.*, **70**, 9A, 465-470.

Acknowledgement

A Dynamical Systems model of combustion kinetics in industrial furnaces

Konstantinos Th. Mavrommatis
Institut für Eisenhüttenkunde der RWTH Aachen

Introduction

The high complexity of the combustion processes is caused by a set of conditions. A high number of chemical components in the overall process, many participating thermodynamic phases, motion, phase transitions and high temperatures are some of them. Quantification paths for the overall combustion processes usually consists of both, experimental and theoretical techniques, which must be complementary to be effective. In computation methodology of combustion processes, due to the above mentioned multiple-caused complexity, Computational Fluid Dynamics (CFD) methods are a very strong tool, which in now in a wide use. It is then evident that a powerfull, allthough itself complex method, like CFD, serves as a tool to support understanding and to quantify the complex combustion kinetics. A high number of adequate boundary conditions is consequently needed in trying to reproduce this complexity of combustion behaviour as realistically as possible. In Pulverised Coal (PC) combustion for example, taking place in metallurgical and in energy aggregates, several hundreds of reaction participants exist simultaneously. It is practically impossible to reproduce their total combustion behaviour in a quantifiable way.

In this contribution an approach is made to model combustion kinetics in industrial furnaces following a different path, which is based on Dynamical Systems (DS) method. In contrast to and complementation of CFD methods usually applied, a set of coupled ODE´s for combustion kinetics of C, O_2, CO and CO_2 is here developed, as these four participants seem to be the essential basis for the early stages of combustion at high heating rates. This is a necessary prerequisite for quantification of the overall kinetics of PC combustion. In a reverse way to the CFD-approach, simplification of combustion in industrial furnaces is faced by the DS-based model in order to enable later effective proces control. This simplification is made through reduction to the above four more essential participants in PC combustion process at high heating rates. Their essentiality is not absolute but is due to the relative target of controlling the combustion process itself in the specific aggregate. The model to be developed will serve as a tool to be applied to different conditions. The target of this work is not to reproduce a specific situation or a set of data achieved by experiment, through suitable modell parameters adjustment, but, instead, to extract informations and the main features of the kinetics of the above four participants, which are contained in the structure of the system of the differential equations itself, i.e. in the dynamical system they define.

To face the overall combustion kinetics on the basis of the chemical dynamics of the four mentioned participants is equivalent to concentrate on main participants and neglecting other ones, eliminating other factors and parameters like grain sizes of the PC and velocity of the grains relative to the gas medium. This restriction should allow to investigate the basic structure of the equations system and their coupling behaviout independently of other parameters, which will complicate the combustion description. However, these additional degrees of freedom should be considered in future modules, but are excluded in this study.

After the development of the mathematical model of the kinetics of the PC combustion at high heating rates, it will be applied to the combustion within the tuyere for PC injection into the blast furnace of ferrous metallurgical operation of iron production.

Reproducibility of expected and measured results supports model validation. Further, questions of stability of the modelled system are investigated. Application of the model to specific combustion process within the tuyere of PC injection into the blast furnace in iron and steel plants contributes further to understanding the overall stability of blast furnace operation. Deduced from the stability considerations of the DS-model, aggregate stability requires operative furnace-intrinsic measures, which are taken in operational practice and which now appear to be in accordance with requirements which are set up by a theoretical model.

The work presented is a part of an actually running research project in the area of on-line monitoring the reactions of PC combustion in metallurgical aggregates.

Development of the mathematical model

The main chemical reactions for PC combustion at high heating rates in the range of 10^5-10^6 Ks^{-1} are controlled by the rate of combustion of solid coal particles. For the whole combustion process only a time of 10-20 ms is available [1]. The main reactions (and their respective enthalpies) are:

1. $C + O_2 = CO_2$ (-94 kcal/mol)   Reaction (a)

2. $2C + O_2 = 2CO$ (-52,8 kcal/mol)   Reaction (b)

3. $2CO + O_2 = 2CO_2$ (-135,3 kcal/mol)   Reaction (c)

4. $C + CO_2 = 2CO$ (+41,2 kcal/mol)   Reaction (d)

Starting from the approach that the rate of change of the concentration is proportional to the actual concentrations of the species taking place in the respective reaction above, the following system of equations can be developed:

$$\frac{d(CO_2^a)}{dt} = k_a(C)(O_2) \tag{1}$$

$$\frac{d(CO^b)}{dt} = \frac{1}{2}k_b(C)^2(O_2) \tag{2}$$

$$\frac{d(CO_2^c)}{dt} = \frac{1}{2}k_c(CO)^2(O_2) \tag{3}$$

$$\frac{d(CO^d)}{dt} = \frac{1}{2}k_d(C)(CO_2) \tag{4}$$

Additionally, due to the above reactions (a) to (d), the following relations can be established:

$$\frac{d(O_2^a)}{dt} = -\frac{d(CO_2^a)}{dt} \tag{5}$$

$$\frac{d(O_2^b)}{dt} = -2\frac{d(CO^b)}{dt} \tag{6}$$

$$\frac{d(O_2^c)}{dt} = -2\frac{d(CO_2^c)}{dt} \tag{7}$$

$$\frac{d(C^b)}{dt} = -\frac{d(CO^b)}{dt} \tag{8}$$

$$\frac{d(C^c)}{dt} = -2\frac{d(CO^d)}{dt} \tag{9}$$

$$\frac{d(CO^c)}{dt} = -\frac{d(CO_2^d)}{dt} \tag{10}$$

$$\frac{d(CO_2^d)}{dt} = -2\frac{d(CO^d)}{dt} \qquad (11)$$

The above relations (5) to (11) simply express the stoichiometric coupling of the species in the reaction equations (a) to (d).

Two different approaches are possible in investigating the kinetic behaviour of the above reactions through mathematical modelling: The first would consist in inserting for the rate coefficients as realistic values as possible into the above ODE´s in order to reproduce real behaviour, which has been or could be measured by experiment. The other one would consist in trying to extract from the above ODE´s as much information as possible about their coupled behaviour in time, neglecting at a first step the specific values of the kinetic coefficients, by setting them equal to unity. Of course, the approach usually made, also made in this work, to set the rate of the reaction in proportionality to the reaction partners, delegates somehow all additional information, besides this proportionality, to the kinetic coefficients, which in general could be functions of the concentrations a/o time. Here the second approach will be followed.

The above set of equations (1) to (11) has been shown to be equivalent to the following equations, in which x_1, x_2, x_3 and x_4 respectively denote C, O_2, CO and CO_2.

$$\frac{dx_1}{dt} = -\frac{1}{2}k^b x_1^2 x_2 - k^d x_1 x_4 + x_1^R \qquad (12)$$

$$\frac{dx_2}{dt} = -k^a x_1 x_2 - k^b x_1^2 x_2 + x_2^R \qquad (13)$$

$$\frac{dx_3}{dt} = \frac{1}{2}k^b x_1^2 x_2 - \frac{1}{2}k^c x_3^2 x_2 + \frac{1}{2}k^d x_1 x_4 \qquad (14)$$

$$\frac{dx_4}{dt} = k^a x_1 x_2 + \frac{1}{2}k^c x_3^2 x_2 - k^d x_1 x_4 \qquad (15)$$

The constants x_1^R and x_2^R denote the rates at which during combustion C and O_2 is externally supplied (injected respectively blown) and in real operation these rates are determined through the operational conditions.

Investigations about the reactions determining the combustion behaviour of Pulverized Coal (PC) at high heating rates are partly contradictory, [2], which is not surprising, taken in mind the extreme operational conditions in plant. The questions targeted here are summarised as follows:

1) What can be deducted from the dynamical system consisting of the above equations, equiped with adequate initial and boundary value conditions, about the structure of the reacting system?

2) Can they reproduce at least qualitatively the observed behaviour of real PC combustion operation?

3) How can they be improved to serve for process control purposes in plant?

4) Does the above system show stability and what should be at this more general level the suggestion for the plant operators?

Application to the combustion in tuyere in PC injection into the blast furnace

The parameters of interest for PC injection which have been investigated are the concentrations of C, O_2, CO, CO_2 and further the O_2/C and the CO/CO_2 ratio and as an operative parameter the (constant) ratio of supplied O_2 to C during combustion. Further the degree of combustion is of practical interest, defined here

as the ratio of the difference of initial and actual, i.e. unburned, C to the initial C concentration, as this is the target of operation to be optimised to avoid blowing a high amount of unburned coal into the blast furnace coke bed, i.e. to avoid decrease of its permeability for gases, thus affecting the overall blast furnace operational stability.

The results of the studies, carried out by a simulation programme which has been set-up in Advaced Continuous Simulation Language, ACSL, [3], starting from the above mathematical model, are summarised in the following figures:

In Fig 1 to 6 the concentrations of the four species C, O_2, CO, CO_2 are plotted in arbitrary units over the time. The parameter PHI is the ration of constant external supply of O_2 to C. A value of PHI=0.6 means that the supply rate of oxygen is 60% of that of carbon. The parameter O_2i denotes the initial oxygen concentration. The figures can be assessed only in relative view.

Concentrations of CO,CO₂,O₂ and C as a function of time(PHI=0.6).
Initial oxygen concentration O₂i=1 .
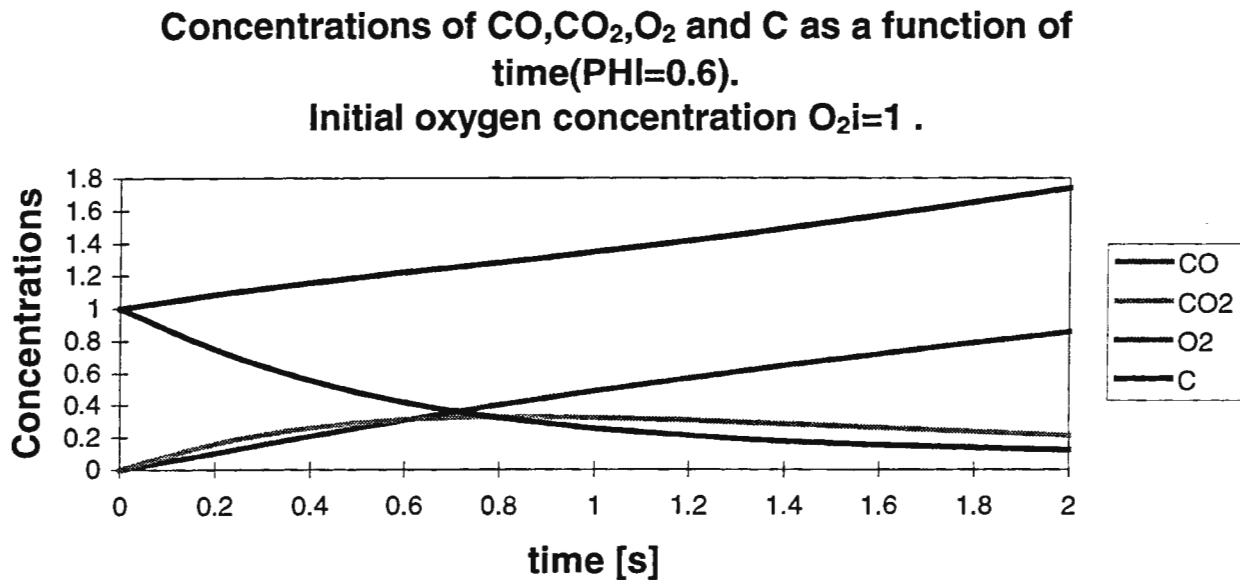
Fig. 1: By 60% relative supply of carbon to oxygen, unburned coal increases, whereas available oxygen decreases. This scenario would cause low gas permeability of the blast furnace.

Concentrations of CO,CO₂,O₂, and C as a function of time (PHI=1.6).
Initial oxygen concentration O₂i=1 .
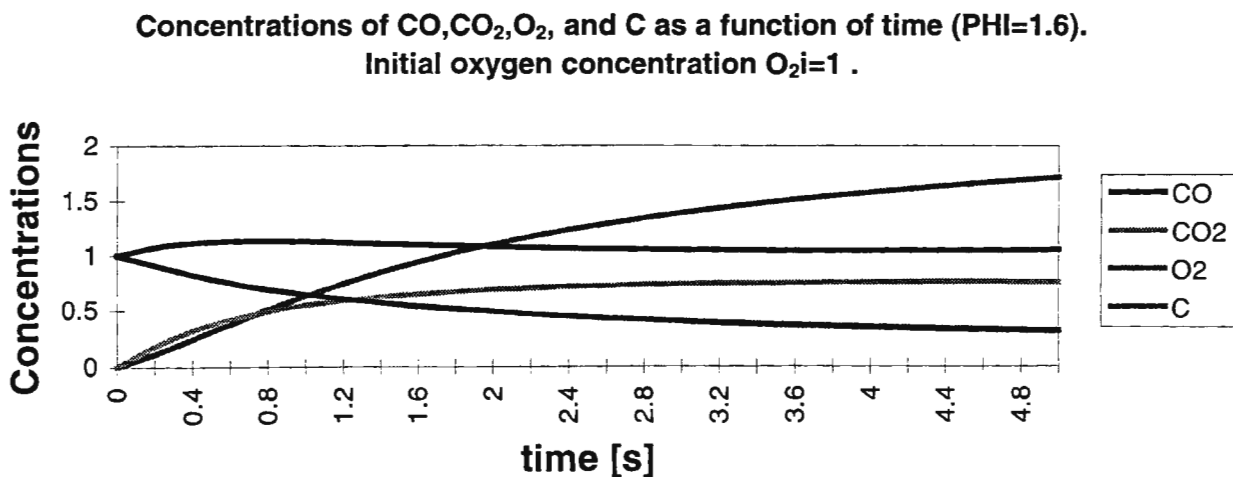
Fig. 2: Coal is constantly burned at 160% oxygen to coal supply, this blast furnace operation would be stable.

Concentrations of CO,CO₂,O₂ and C as a function of time(PHI=0.6). Initial oxygen concentration O₂i=4 .
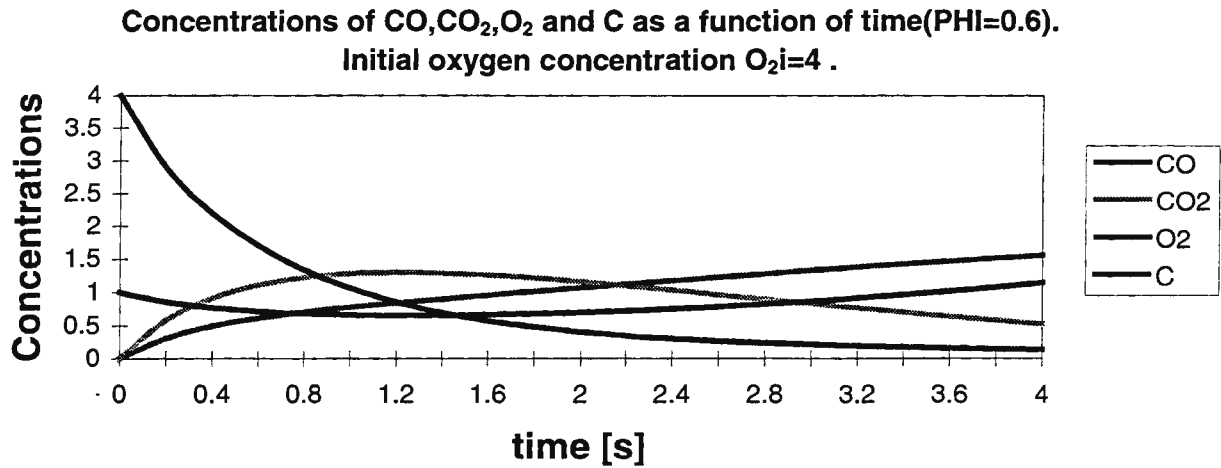
Fig. 3: At 4 times higher initial oxygen content than in the previous two Figures 1 and 2, by 60% oxygen to coal supply, the carbon concentration is firstly decreased and then increases, whereas oxygen is continuously consumed. This operation would be not desirable.

Concentrations of CO,CO₂,O₂ and C as a function of time(PHI=1.6). Initial oxygen concentration O₂i=4 .
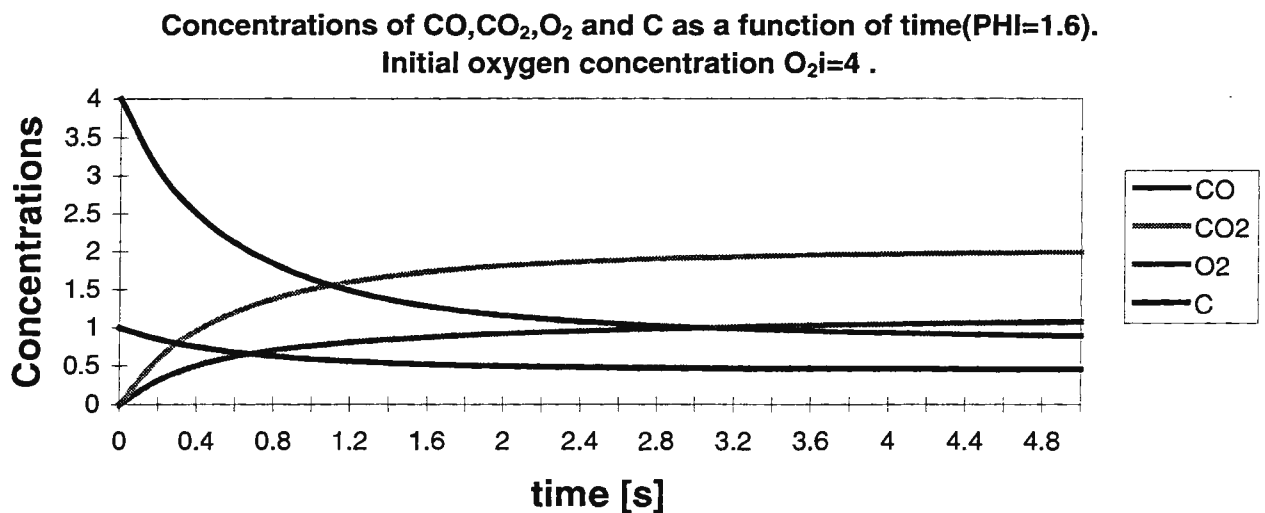
Fig. 4: In contrast to Figure 3, at 160% oxygen to coal supply carbon is burned constantly and the blast furnace would be stable operated.

Concentrations of CO and CO₂ as a function of time(PHI=0.1 - 1.6). Initial oxygen concentration O₂i=1 .
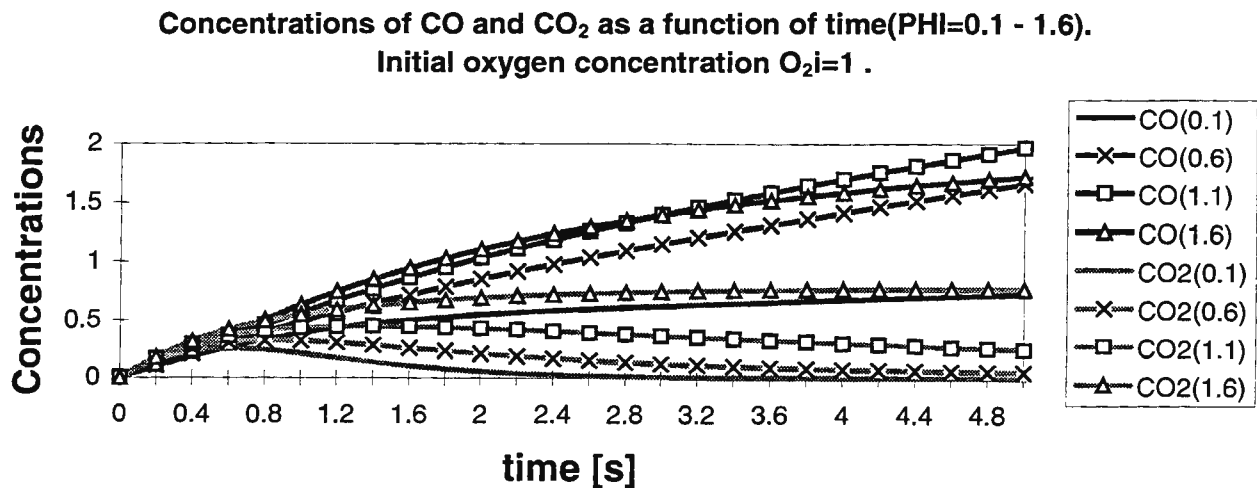
Fig. 5: Increasing the oxygen to coal supply from 10% up to 160% in steps of 50% shows that there is a point where the CO production would increase with smaller rate allthough at higher supply percentage. This indicates a change in the overall combustion processing at that point.

364

Concentrations of C and O₂ as a function of time (PHI=0.1 - 1.6).
Initial oxygen concentration O₂i=1 .
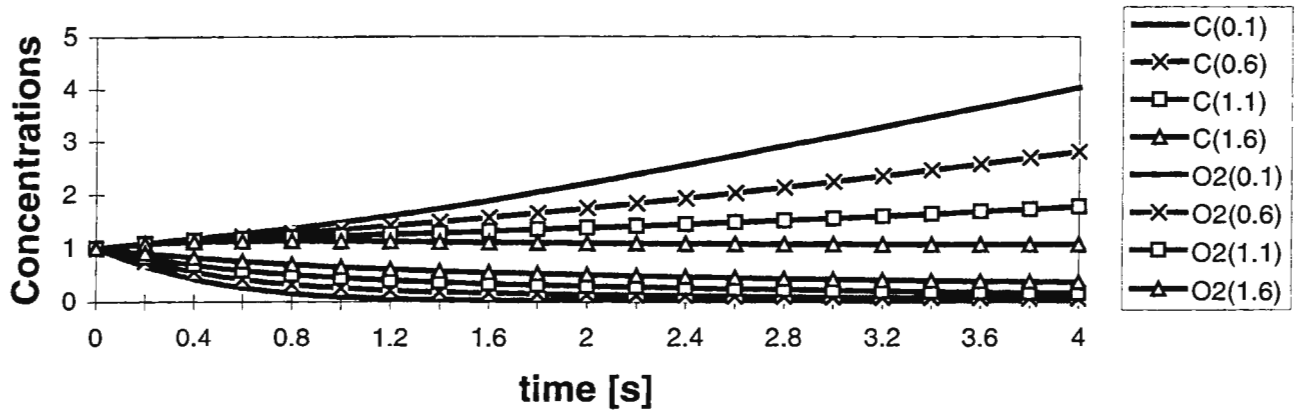
Fig. 6: Summarised are the simulation runs of carbon and oxygen concentration in the tuyere of blast furnace PC injection for relative supply ratios oxygen to carbon, from 10% to 160%. The most stable operation is at the highest supply ration, as carbon is burned constantly.

Concentrations of CO and CO₂ as a function of time(PHI=0.1 - 1.6).
Initial oxygen concentration O₂i=2 .
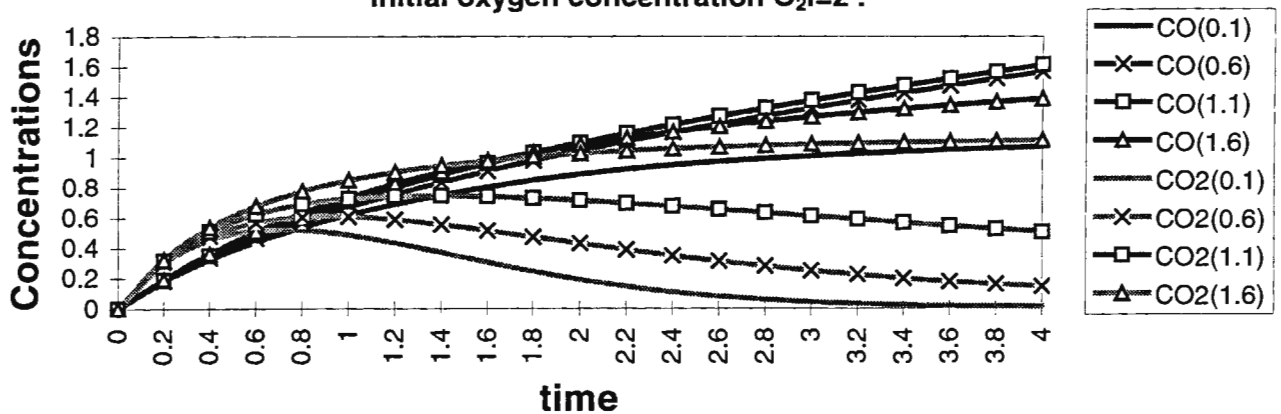
Fig. 7: At higher initial oxygen concentration, double the one in Figures 5 and 6, CO shows the observed change of slower increase allthough higher supply ratio.

Concentrations of C and O₂ as a function of time(PHI=0.1 - 1.6).
Initial oxygen concentration O₂i=2 .
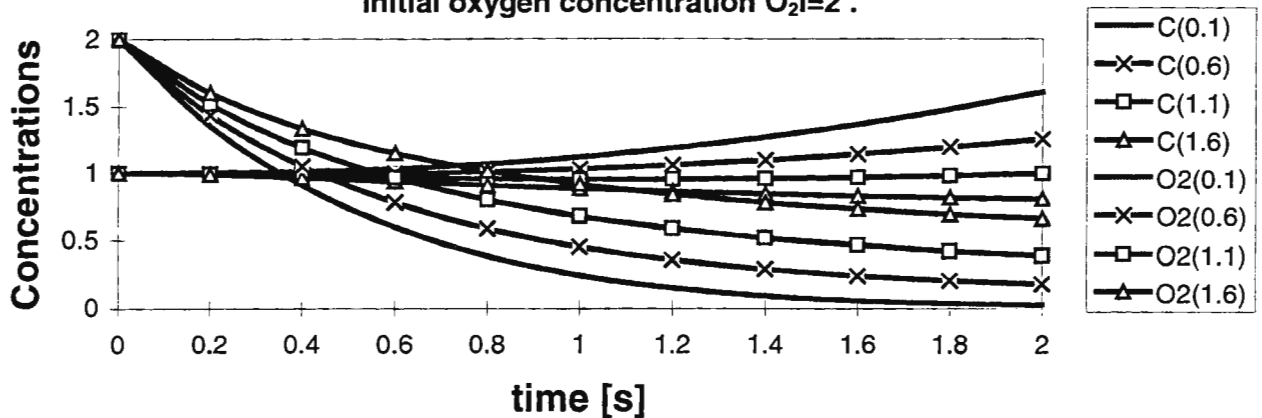
Fig. 8: At double so high initial oxygen concentration, the most stable coal burning takes place at supply ratio of 110%, not at the highest one. This indicates a higher potential for additional coal injection to the same oxygen supply, at least from the point of view of the simulated kinetics.

Concentrations of CO and CO₂ as a function of time(PHI=0.1 - 1.6). Initial oxygen concentration O₂i=4 .
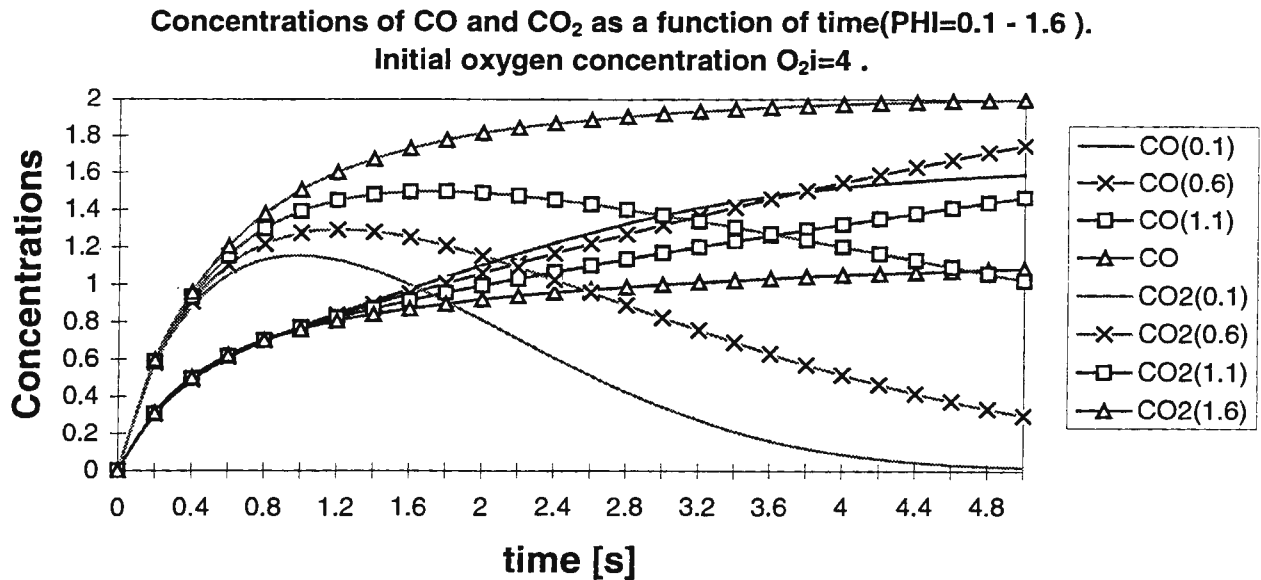
Fig. 9: The initial oxygen concentration is four time so high, and from the different simulation runs for the concentrations of carbon monoxide and carbon dioxide it is indicated that now both have a point of change in their behaviour in the range between a supply ratio of 110% to 160%, through slower increasing and not more decreasing, respectively. The combustion process is producing now relatively more dioxide and less monoxide, i.e. the combustion takes place more completely, as expected from the theory and verified in practice.

Concentrations of C and O₂ as a function of time(PHI=0.1 - 1.6). Initial oxygen concentration O₂i=4 .
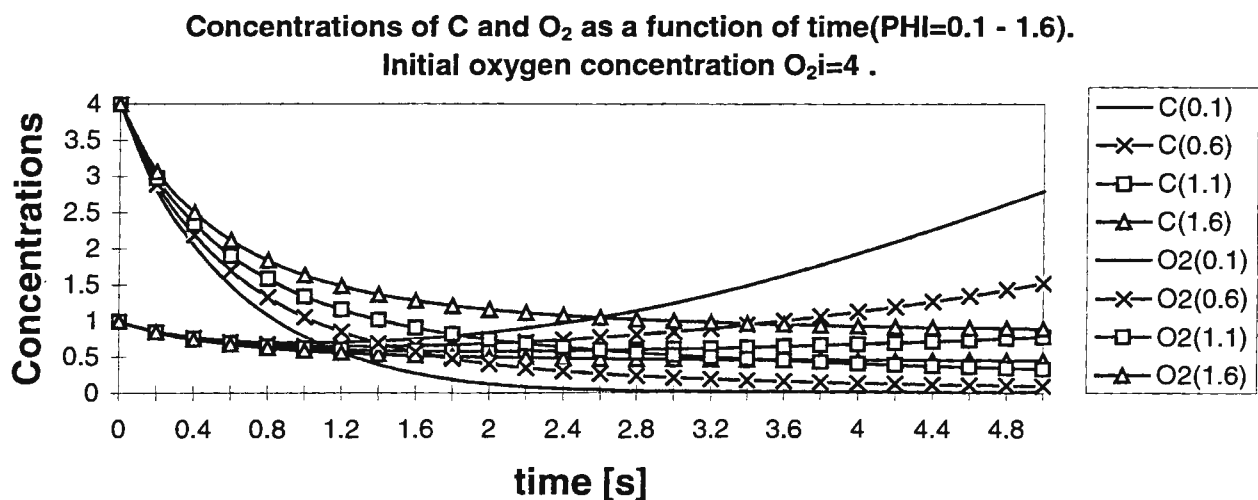
Fig. 10: At high initial oxygen concentration and low supply ratio oxygen to carbon the combustion of coal is incompletely running, the operation is stabilised at high supply rates.

Conclusions

1) From the dynamical system consisting of the above equations and the suitable initial and boundary values conditions it can be deduced that the overall behaviour of the PC combustion system in tuyere of the blast furnace operation corresponding to high heating rates can be reproduced by the above structure and coupling of the equation system for the chemical kinetics

2) The main features and characteristics of the real combustion system adressed, like time behaviour of the concentrations, and of the combustion degree, are reproducible and in accordance with the metallurgical practice. The values given are not absolute and the overall assessment is relative, but the systems structure enables the model to be applied in plant and lab-scale with suitably determined kinetic coefficients.

ETA as a funtion of time(PHI=0.1 - 1.6).
Initial oxygen concentration O$_2$i=4 .Ci=1.0 .
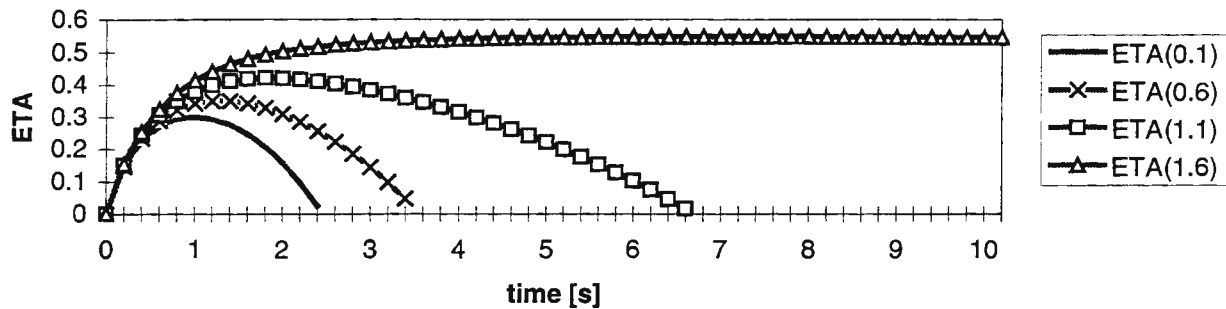
Fig. 11: The degree of combustion, defined as the ratio between the difference of initial carbon concentration and the actual, unburned carbon to the initial one shows an increasing and after passing through a maximum, a decreasing behaviour. A saturation, which corresponds to a constant combustion in time, is achieved only for a higher supply ratio of oxygen to carbon, 160% in these lelative figures and amounts here slightly more than 50%. This behaviour qualitatively shown here is confirmed in practice, the optimum combustion degree being about 70% (ETA 70).

3) Improvement should be possible through considering of coal particles sizes, movement, mutual interaction in the injected stream and their influence on chemical kinetics of combustion process.

4) The system of equations of the developed model indicates qualitatively an instability, which seems to correspond to an intrinsic instability of real metallurgical operation. A full assessment can be given only with validated kinetic coefficients. As the existing knowledge about kinetic coefficients of the corresponding reactions seems poor, further work on their determination and validation is suggested.

Literature

1. Markus Stephan Rudack: Einblasen von Kohle mit Zuschlagstoffen in den Hochofen, Dissertation RWTH Aachen, Shaker Verlag, D 82, Band 3/94, 1994

2. Stefanie Wippermann: Kombiniertes Einblasen von Kohle und Feinerz oder eisenhaltigen Hüttenreststoffen in den Hochofen, Dissertation RWTH Aachen, Shaker Verlag, D 82, Band 9/96, 1996

3. Advanced Continuous Simulation Language, ACSL, Mitchell and Gauthiers Associates, Concord., Mass. 01742, 1986

DYNAMIC SIMULATION OF A CONDENSATION PLATE COLUMN BY DYNAMIC DECOUPLING

Francesco Casella[1], Alberto Leva[2], Claudio Maffezzoni[2]

[1]Dip. Elettronica e Informazione, Politecnico di Milano, p.za L. da Vinci 32, 20133 Milano, ITALY, casella@elet.polimi.it
[2]Dip. Elettronica e Informazione, Politecnico di Milano, p.za L. da Vinci 32, 20133 Milano, ITALY, leva@elet.polimi.it
[3]Dip. Elettronica e Informazione, Politecnico di Milano, p.za L. da Vinci 32, 20133 Milano, ITALY, maffezzo@elet.polimi.it

Introduction

The exploitation of low-energy geothermal resources, consisting of hot-water underground storages, requires a special (new) process to extract pure steam from the underground fluid. The considered process plant is under design and construction and is of really new concept; for this reason, simulation is necessary to check off-design operating conditions, to evaluate plant response in operating transients and to design process control.

The most complex part of the process (see Fig. 1) is devoted to separating the uncondensable gas component (mainly carbon dioxide) from the steam: this is done by a special plate column (under design patent) in which steam is removed from the gas phase by condensation at the highest possible temperature; the resulting hot water is then flashed twice to produce clean vapour. Unlike classical distillation columns, which are usually modelled by mass balances only, the considered process is essentially dependent on energy balances in the different plates.

The paper first describes the basic principles applied for the modular modeling of the condensation column, with use of non-equilibrium mass and energy conservation equations. The resulting model of the 14-plate column (which has also to be integrated with the rest of the process model) is of large scale, consisting of 90 nonlinear DAE's, plus about 200 equations in form of explicit assignements. The DAE's are coupled among themselves; moreover, fast dynamics and algebraic equations are present, so that implicit integration methods have to be applied. Then, the paper presents the application of a special modeling-simulation technique, called *weakening* [1], based on cutting, within any integration step, those variable-equations interactions which are sufficiently weak that the integration stability is preserved if they are treated in explicit form. In other

Figure 1. Simplified plant diagram

terms, in a given equation, a weak variable is given the (fixed) value computed at the previous time step, while the rest of the variables are unknowns in the implicit integration scheme. With this expedient, the DAE system can be rearranged so as to put it in a block-triangular form. In this way, the implicit system of maximum size, to be solved at each time step, has the dimension equal to the number of independent pressure nodes assumed in the column model (15 in this case). The model thus obtained has been implemented in ProcSim [2], a process simulation environment conceived to systematically apply the weakening technique to general thermohydraulic processes. The benefits of the proposed method is obvious: the drastic size reduction of the implicit problem to be solved circumvents possible numerical problems and dramatically reduces the computing time required by the simulator.

General model of the water+CO$_2$ control volume

Both the column plates and the column bottom can be modeled as volumes containing two perfectly separated homogeneous phases: a gas mixture (water vapour+CO$_2$) and a liquid mixture (liquid water with CO$_2$ in solution). To obtain a thermodynamic non-equilibrium model, with different temperatures for the gas and liquid phases, the following assumption is made: the gaseous mixture is always in saturation state, i.e. at
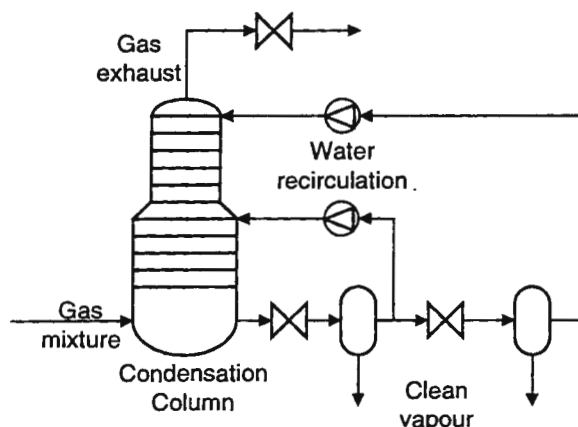
a temperature corresponding to the saturation temperature of the water vapour at its partial pressure, while the liquid water can be out of its saturation state. This is reasonable, since the energy content of the gas mixture depends much more on the latent heat of vapourization of the vapour fraction than on the temperature. The gas mixture is described using Dalton's law: each component behaves as if it were alone in the control volume at the mixture temperature. The CO_2 mass fraction in the liquid phase can also be different from its saturation value, given by Henry's Law; in this case, a mass flow will arise between the two phases towards the equilibrium condition. The heat of solution of the carbon dioxide is neglected.
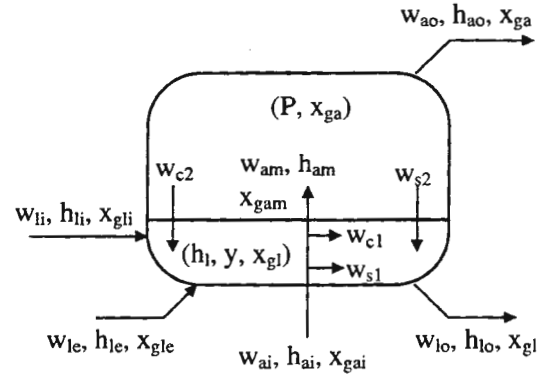
Figure 2. The general volume model

Each flow is described by its mass-flow rate w, its specific enthalpy h and the mass fraction of carbon dioxide x_g (see Fig 2). The gas mixture input flow (ai subscript) enters the liquid control volume, possibly releasing a water vapour condensation flow w_{c1} and a CO_2 solution flow w_{s1}, then enters the gas control volume (am); the gas mixture output flow exits from the gas control volume. The liquid input flow (li), output flow (lo) and optional extra liquid input flow (le) all directly enter and exit the liquid control volume. Moreover, a water vapour condensation flow w_{c2} and a CO2 solution flow w_{s2} may exist at the interface between the two phases.

The proposed model has five state variables: the pressure P, the mass fraction of CO_2 in the gas mixture x_{ga}, the specific enthalpy of the liquid water h_l, the level y and the mass fraction of CO_2 in the liquid mixture x_{gl}. Five balance equation can be written:

$$\dot{M}_a = w_{ai} - w_{ao} - w_{c1} - w_{s1} - w_{c2} - w_{s2} \tag{1}$$

$$\dot{M}_g = w_{ai}x_{gai} - w_{ao}x_{ga} + w_{li}x_{gli} - w_{lo}x_{gl} + w_{le}x_{gle} \tag{2}$$

$$\dot{E} = w_{ai}h_{ai} - w_{ao}h_{ao} + w_{li}h_{li} - w_{lo}h_{lo} + w_{le}h_{le} \tag{3}$$

$$\dot{M}_{wl} = w_{li}(1-x_{gli}) - w_{lo}(1-x_{gl}) + w_{le}(1-x_{gle}) + w_{c1} + w_{c2} \tag{4}$$

$$\dot{M}_{gl} = w_{li}x_{gli} - w_{lo}x_{gl} + w_{le}x_{gle} + w_{s1} + w_{s2} \tag{5}$$

where (1) is the conservation of the total mass in the gas control volume, (2) is the conservation of CO_2 mass in the whole volume, (3) is the conservation of energy in the whole volume, (4) is the conservation of water mass in the liquid control volume and (5) is the conservation of CO_2 mass in the liquid control volume. The left-hand-side terms can be written as a function of the state variables taking into account the geometric features of the volume, the water and steam tables, and the properties of the ideal gas CO_2:

$$M_a = M_a(P, x_{ga}, y) = \rho_a(P, x_{ga}) \cdot V_a(y) \tag{6}$$

$$M_g = M_g(P, x_{ga}, h_l, y, x_{gl}) = \rho_l(P, h_l) \cdot x_{gl} \cdot V_l(y) + \rho_g(P, x_{ga}) \cdot V_a(y) \tag{7}$$

$$E = E(P, x_{ga}, h_l, y, x_{gl}) = \rho_l(P, h_l) \cdot [h_l + x_{gl}h_{gl}(P, h_l)] \cdot V_l(y) + \rho_a(P, x_{ga}) \cdot h_a(P, x_{ga}) \cdot V_a(y) - PV \tag{8}$$

$$M_{wl} = M_{wl}(P, h_l, y) = \rho_l(P, h_l) \cdot V_l(y) \tag{9}$$

$$M_{gl} = M_{gl}(P, h_l, y, x_{gl}) = \rho_l(P, h_l) \cdot x_{gl} \cdot V_l(y) \tag{10}$$

where ρ_l is the density of the liquid water, ρ_a is the density of the gas mixture, ρ_g is the density of the CO_2 in the gas mixture, V_l and V_a are the volumes of the liquid and gas phases, V is the total volume, h_a is the specific enthalpy of the gas mixture and h_{gl} is the specific enthalpy of the CO_2 in solution. The right-hand-side terms are either exogenous variables, or functions of the state variables:

$$h_{lo} = (1 - x_{gl})h_l + x_{gl}h_{gl}(P, h_l) \tag{11}$$

$$h_{ao} = h_a(P, x_{ga}) \tag{12}$$

369

Note that, having two control volumes containing a two-component mixture, six equations should be written: two for the mass of water, two for the mass of CO_2 and two for the energy. However, the hypothesis of saturation for the gas mixture is equivalent to an energy balance equation stating that there is a suitable heat exchange between the two phases, so that the equations actually reduce to five. The remaining equations could be replaced by suitable linear combinations (e.g. the conservation of the total mass in the whole volume instead of eq. (1)), and other state variables could be selected (e.g. the partial pressure of CO_2 instead of x_{ga}); the actual choice has been made according to heuristic considerations in order to achieve the maximum possible decoupling between the different equations and the different state variables.

To obtain a complete model of the condensation column, two versions of the volume model are needed: one for the generic plate, that will be replicated N times, and one for the bottom part of the column.

Model of the column plate and column bottom

In the column plates, the gas mixture coming from below bubbles through the liquid phase (actually a rather thin layer some centimeters thick) before reaching the gas phase: consequently mass and energy flows between the two phases take mainly place in this layer, before the input flow actually reach the gas control volume. A Murphree-like [3] plate efficiency η can be defined:

$$\frac{x_{gam} - x_{gai}}{x_{ga}^* - x_{gai}} = \eta \qquad (13)$$

where x_{gam} is the CO_2 mass fraction of the mixture actually entering the gas phase and x_{ga}^* is the CO_2 mass fraction corresponding to the equilibrium condition with the liquid phase. Application of mass balances for water and CO_2 through the liquid layer leads to the following equations:

$$x_{gam} = x_{gai} + \eta(x_{ga}^* - x_{gai}) \qquad (14)$$

$$w_{c1} = \frac{w_{ai}(x_{gam} - x_{gai}) - w_{s1}(1 - x_{gam})}{x_{gam}} \qquad (15)$$

This choice seems reasonable, since the condensation flow-rate is proportional to the gas mixture flow-rate w_{ai} and to the quantity $(x_{ga}^* - x_{gai})$ which is related to the temperature difference between the incoming gas flow and the liquid layer. If $\eta=1$, an equilibrium model is obtained.

Due to the low solubility and heat of solution (which is actually neglected), the CO_2 flow-rate w_{s1} is a side effect of the process, compared with the main phenomenon of steam water condensation. As a first approximation, it can be thought of as proportional to the gas mixture flow-rate w_{ai} and to the driving force given by the difference between the the CO_2 partial pressure of the gas mixture exiting the liquid P_g and the equilibrium pressure corresponding to the CO_2 mass fraction in the liquid x_{gl}:

$$w_{s1} = k_{s1}w_{ai}(P_g(P,x_{gam}) - H(T_l(P,h_l) \cdot x_{gl}) \qquad (16)$$

where $H(T_l)$ is Henry's constant for CO_2 in water, which is a function of water temperature. When k_{s1} goes to infinity, one obtains the equilibrium condition $P_g = H x_{gl}$, i.e. Henry's law. Since the solubility of CO_2 in water is rather low, the modeling accuracy of eq. (16) is anyway not an issue.

Condensation and CO_2 mass-flows at the interface between the liquid and the gas control volumes are modeled as proportional to their corresponding driving force. The proportionality constants k_{c2} and k_{s2} have to be kept low to obtain a realistic description of the system. In normal plate operating conditions, the mass exchange occurs mainly in the liquid layer, so that w_{c2} and w_{s2} can be neglected; they are included here for completeness:

$$w_{c2} = k_{c2} \cdot \left[T_a(P,x_{ga}) - T_l(P,h_l)\right] \qquad (17)$$

$$w_{s2} = k_{s2} \cdot \left[\frac{P_g(P,x_{ga})}{H(T_l(P,h_l))} - x_{gl}\right] \qquad (18)$$

The column plates used in the plant are fitted with veirs, so that the liquid output flow-rate w_{lo} is a function of the level y; for simplicity a linear approximation is used:

$$w_{lo} = k_l \cdot (y - y_o) \tag{19}$$

The gas mixture flow-rate, instead, depends on the pressure drop between the plate and the underlying volume (plate or bottom), which is the sum of a quadratic friction term and of a hydrostatic term (g is the gravitational acceleration):

$$\Delta P = k_q w_{ai}^2 + \rho_l (P, h_l) \cdot g \cdot y \tag{20}$$

The column bottom has been designed as a simple water reservoir, in which the gas mixture enters directly the gas control volume without first bubbling through the liquid layer: therefore $w_{c1} = 0$ and $w_{c2} = 0$; only the mass exchanges at the interface (equations 17-18) are present.

Since there are no veirs in the column bottom, the liquid output flow w_{lo} is exogenous, and a level control system is needed to keep the water level in the allowed range. The input gas flow depends on the boundary conditions imposed by the feed system.

Analysis of the incidence matrices

The column model is obtained by collecting the plate and bottom equations, while introducing the trivial binding equations between flows exiting one plate ad entering the next (i.e. $w_{ao,k+1} = w_{ai,k}$; $h_{li,k+1} = h_{lo,k}$ etc.). The resulting Jacobian incidence matrix for equations (1)-(5), (20), describing the generic k-th plate becomes the following:

	P	x_{ga}	h_l	y	x_{gl}	w_{ai}	P	x_{ga}	h_l	y	x_{gl}	w_{ai}	P	x_{ga}	h_l	y	x_{gl}	w_{ai}
M_a						X	X	X	X	X	X	X		X				
M_g				X	X	X	X	X	X	X	X	X		X				
E	X		X	X	X	X	X	X	X	X	X	X	X	X				
M_{wl}				X	X		X	X	X	X	X	X		X				
M_{gl}				X	X		X	X	X	X	X	X		X				
w_{ai}							X		X	X		X	X					
	Plate k-1 (upper)						Plate k						Plate k+1 (lower)					

Table 1. Jacobian incidence matrix of the original plate equations

The incidence matrix for the column bottom equations is the same, but without the row and column corresponding to w_{ai} (the input gas flow-rate is a boundary condition) and without the columns corresponding to the lower plate variables, which are again given by boundary conditions.

Looking at the matrix of Table 1, we may notice that all the equations of the column model are mutually coupled because of the following mechanisms:

- the pressure P relative to the k-th volume interacts with the pressures of the neighbouring volumes through the gas mass flow-rates (see the first and last equations);

- the equation set relative to the k-th volume has essentially a full Jacobian matrix, so that the interaction among the plate pressures determines the mutual coupling among all the variables.

Because the hydraulic resistance between neighbouring plates is very small, the coupling among the different pressures along the column through the gas flow rates w_{ai} (whatever the mass fraction x_{ga}) is very stiff, so that implicit simultaneous solution of the column hydrodynamics (the set of M_a- and w_{ai}-equations) is necessary. However, the influence of the other variables (x_{ga}, h_l, y, x_{gl}) on the hydrodynamic subsystem is significantly *weaker* so that one may solve the hydrodynamic subsystem by keeping all those variables at the values taken at the previous time step. This remark leads to neglecting, *within an integration interval,* the

corresponding elements in the (Jacobian) incidence matrix as shown in Table 2. Along the same line of reasoning, some other variables can be recognized to have a *weak* influence in certain equations: for instance, in M_g equation variables h_l, y and x_{gl} of the same plate and variables y and x_{gl} of the upper plate. This has been taken into account to obtained the *"weakened"* incidence matrix of Table 2.

	P	x_{ga}	h_l	y	x_{gl}	w_{ai}	P	x_{ga}	h_l	y	x_{gl}	w_{ai}	P	x_{ga}	h_l	y	x_{gl}	w_{ai}
M_a						X	X					X						
M_g						X	X	X				X		X				
E	X		X	X	X	X	X	X	X	X	X	X	X	X				
M_{wl}				X	X		X	X	X	X	X	X		X				
M_{gl}				X	X		X	X	X	X	X	X		X				
w_{ai}							X					X	X					
	Plate k-1 (upper)						Plate k						Plate k+1 (lower)					

Table 2. Incidence matrix of the original plate equations after weakening

Suppose, for the sake of simplicity, the column has only two plates plus the bottom. The complete model results from the composition of the equations for the column plates, column bottom and the top exhaust valve (whose equation defines the output flow-rate w_{ao} for the top plate as a function w_v of the top pressure). Thanks to the weakening procedure, the complete system of equations can be rearranged in such a way to show a block-triangular structure (the subscripts 1, 2, b and v indicate top plate, middle plate, bottom and exhaust valve, respectively):

	P_1	P_2	P_f	w_v	w_{ai1}	w_{ai2}	x_{gaf}	x_{ga2}	x_{ga1}	h_{l1}	y_1	x_{gl1}	h_{l2}	y_2	x_{gl2}	h_{lf}	y_f	x_{glf}
w_v	X			X														
w_{ai1}	X	X			X													
w_{ai2}		X	X			X												
M_{a1}	X			X	X													
M_{a2}		X			X	X												
M_{ab}			X			X												
M_{gf}			X			X	X											
M_{g2}		X				X	X	X										
M_{g1}	X			X	X			X	X									
E_1	X	X		X	X			X	X	X	X	X						
M_{wl1}	X				X			X	X	X	X	X						
M_{gl1}	X				X			X	X	X	X	X						
E_2	X	X	X		X	X	X	X		X	X	X	X	X	X			
M_{wl2}		X					X	X			X	X	X	X	X			
M_{gl2}		X					X	X			X	X	X	X	X			
E_f		X	X			X	X						X	X	X	X	X	X
M_{wlf}			X				X							X	X	X	X	X
M_{glf}			X				X							X	X	X	X	X

Table 3. Incidence matrix of column after weakening and rearrangement

The matrix thus obtained also describes the actual computational procedure. The system is solved blockwise: first the equations relative to the gas mixture mass-flow, then those relative to CO_2 mass fraction of the gas mixture, then those associated with the liquid flow. Each block of differential equations is solved using a fixed-step, implicit Euler integration algorithm, to ensure the maximum degree of numerical stability. The first block of equations is automatically obtained by the simulation environment by assembling the models of the single components and the topological information about the flow network supplied by the user; the remaining blocks are also sorted by the model builder following the directions of the two streams (gas and liquid) through the column. Note that the first N equations (where $N=3$ is the number of unknown pressures) are algebraic equations of the type $w_{ai,k} = f(P_k, P_{k+1})$ that can be explicitly solved and readily substituted into the next N equations, so that the dimension of the hydrodynamic system to be solved is

actually *N*, not *2N*; the resulting system of nonlinear ODE's is discretized by the Euler implicit method and then solved by the Newton method. If the above-sketched procedure is applied to the real column, which has 14 plates plus the bottom, the total number of equations is 90, while the maximum number of equations to be solved simultaneously is just 15.

Simulator performance

The described column model has been integrated with the other plant components (whose description is outside the scope of this paper) with the same methodology that has been outlined in the previous paragraphs for the different volumes of the column. The full plant model is described by approximately 180 state variables and over 400 output variables. The simulator is numerically stable for integration steps up to 0.6 seconds; beyond that, instability may arise because of the interactions between the weakly interacting variables. With the maximum allowed time step, the simulator almost runs in real time on a Pentium P200 personal computer, while supporting very accurate computation of water-steam properties and an advanced interactive MMI. It is interesting to notice that numerical stability is ensured even if the pressure drops across the plates (less than 0.01 bar) are much smaller than the pressure drop across the top exhaust valve (over 5 bar), meaning that the simulator is able to reproduce the fundamental dynamics of the system even in presence of fast dynamics with associated time constants much shorter than the simulation step. Moreover, it is the authors' opinion that the simulator efficiency could be further enhanced by optimized rewriting of the plate equations.

Example transients

As an example, five transients obtained with the full simulator are reported. The first (Fig. 3) is the response of the top plate pressure to a small step variation at time *t*=0 in the exhaust valve opening. The response shows a dominant time-constant at about 200 s, plus some zero-pole pairs. This means that the column pressure can be easily controlled with a standard PI controller. The second (Fig. 4) and the third (Fig. 5) are the responses of the temperature gap between the top recirculated water and the output gas exhaust, which should be a controlled quantity, to a small step variation of the top recirculation and the middle recirculation valves, respectively. Note that the transient of Fig. 5 shows a rather large

Figure 3. Top pressure transient

direct feedthrough term. Both responses show a rather complex dynamic behaviour. Last, to demonstrate the
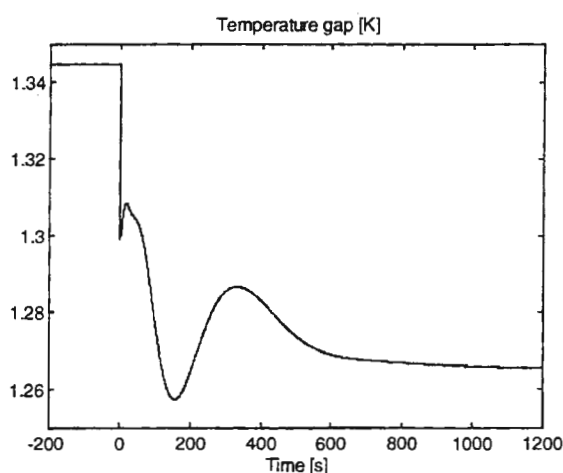
Figure 4. Temperature gap transient
(top recirculation)
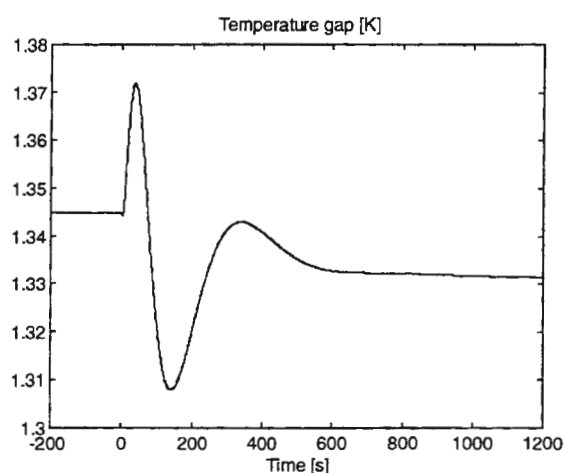
Figure 5. Temperature gap transient
(middle recirculation)

373
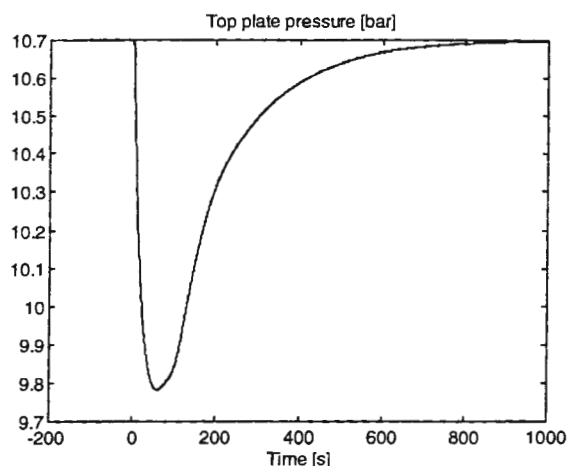
Figure 6. Top pressure transient
(large load decrease)
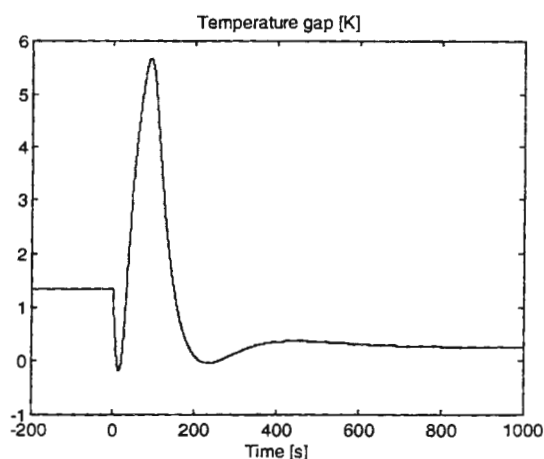
Figure 7. Temperature gap transient
(large load decrease)

ability of the simulator to reproduce large transients, simulation results in response to a 60% step reduction of the input gas mixture flow-rate has been simulated; results are shown in Figs. 6 and 7. This transient has been simulated with PI control on the top pressure, to ensure that the operating limits on the column pressure are not violated, and it is of particular interest, since it corresponds to a particularly critical pump trip event in the real plant.

Conclusions

The paper has discussed a methodology to build a simulation model of a condensation plate column, that meets the engineering requirement of reproducing process complexity while ensuring numerical robustness and limited computational burden. The methodology, called *weakening*, amounts to defining a number of variable-equation relationships as "weak", basing the selection on the peculiar dynamic properties of the process under consideration. With the application of this methodology (considered in detail for the plate column) the whole geothermal process under design has been modelled with a DAE system of about 600 variables but limiting the dimension of the largest implicit system to be solved at 24. The weakening paradigm has also been implemented in a systematic way in the Process Simulation environment *ProcSim*, specifically conceived for thermohydraulic processes (including multi-component flows), so that model decoupling specifications are given by the user in standardized forms; for instance, the user is required to structure the "hydraulic network" of the process in terms of pressure nodes and flow branches, by using library models which are already composed by twin submodels, a hydraulic submodel and a thermal submodel. Moreover, the hydraulic network can be partitioned in "weakly" interacting subnetworks, by very intuitive semantics.

Thus, the resulting simulator is sufficiently flexible and light to be run on a PC, but without losing model fidelity which is essential for engineering simulators. The potential of the developed simulator has also been shown by reporting the simulated transients both relative to small perturbations and to a very large disturbance corresponding to an abrupt 60% load reduction.

References

[1] F. Casella, C. Maffezzoni: "Exploiting Weak Interactions in Object-Oriented Modeling", *EUROSIM Simulation News Europe*, March 1998.
[2] A. Bartolini, A. Leva, C. Maffezzoni: "A Visual Environment for Building Process Simulators", Proceedings ITEC '96, Le Hague 1996, pp. 381-389.
[3] W. L. Luyben, "Process Modeling, Simulation, and Control for Chemical Engineers", McGraw-Hill.

MODELLING AND SIMULATION OF HEAP LEACHING PROCESSES

Zoltán Cziprián [1], Sirkka-Liisa Jämsä-Jounela [2] and Pedro Munoz[3]

[1] Helsinki University of Technology, Kemistintie 1, 02150 Espoo, Finland, czip@cc.hut.fi

[2] Helsinki University of Technology, Kemistintie 1, 02150 Espoo, Finland, sirkka-1@cc.hut.fi

[3] Universidad de Santiago de Chile, Alameda 3363, Santiago Chile, pmunoz@lauca.usach.cl

Abstract

Ore leaching techniques combined with solution treatment procedures offer exceptional conditions to produce high purity metal from very low-grade ore sources. Essential percentage of the world copper production is based on the heap and dump leaching techniques. The use of computer simulation, is now a basic tool for process design and optimization, as evidenced by its increasing use of practicing engineers through commercially available simulator software. This paper discusses the modelling and simulation of the hydrometallurgical processes. The simulation example of a copper heap leaching process is presented and evaluated.

Introduction

Engineers, in charge of designing equipment for the mineral processing plants have to be able to select the most suitable equipment for each application and to predict the performances under different process conditions. With the increasing capacity of microcomputers, the software companies would like to provide computer tools for plant operators to obtain the best results from the equipment and process. In order to allow the mineral processing engineers to concentrate on their task, the simulation software should offer a user-friendly interface combined with quick and accurate calculations.

Hydrometallurgical processes are becoming more and more important in the mineral processing industry, especially in treatment of low grade ores, due to their low capital investment for a given metal extraction rate and due to the low operating costs as well. Simulation offers the advantage of predicting the behavior of the processes without making excessive tests at the real process. Simulations are quicker, more safe and economic than the process experiments.

The benefits of the use of the simulator software for the design and optimization of the hydrometallurgical processes have been proved through the several industrial applications.

At the Laboratory of Process Control and Automation from the Helsinki University of Technology, one project was started to build the simulation software for teaching purposes. This simulator is designed to include hydrometallurgical and mineral processing units. Process control and automation elements will be implemented as well. Earlier versions of the simulator gave good results in the field of mineral processing. Recently the research is focused on heap leaching process modelling and simulation. In the first stage, several simulations were carried out using a commercial simulation-software, called METSIM, and real heap leaching plant data from Codelco, Chile. The simulation results are compared with the plant data. In the second stage, the development of unit models required for heap leaching process simulation was started. These models are still under development, however the main modules of the simulator are presented.

Heap leaching

Leaching is a process conducted with a solution in which there is a solubility of the dissolving minerals or compounds in the material being leached sufficient to obtain an acceptable concentration of valuable metals in solution. The principle of heap leaching is presented in Figure 1. In case of heap leaching, the crushed and prepared ore is placed in a form of heaps on gently sloping flat ground covered with an impervious pad, and sprinkled with lixiviant.

The leaching time usually is three to six months for oxide ores, and ten to twenty months for sulfides. The *water table level* usually is not high even it can be missing, contributing to substantial oxidation of sulfide minerals. A typical configuration for copper production including the heap leaching is shown in Figure 2.

The heap leaching process involves heterogeneous chemical reactions on the mineral surface catalyzed with bacterial activity. There are several main bioprocess-engineering factors that affect the kinetics of bacterial leaching of sulfide copper ores, namely bacterial population, mineral and particle size, nutrients and inhibitors, temperature, pH, oxygen and carbon dioxide supply, leaching kinetics and operational mode [2,3,4]. Under water level, conditions are not oxidizing unless ferric ions are present. Sulfide minerals are usually not leached or are leached very inefficiently in absence of oxygen. Channeling, compaction and blinding-off the heaps can occur as a result of moderate changes in pH, temperature or solution composition, reducing the recovery rate of the copper. Mineralogy of the ore determines leachability and the nature of unwanted side reactions and precipitation.
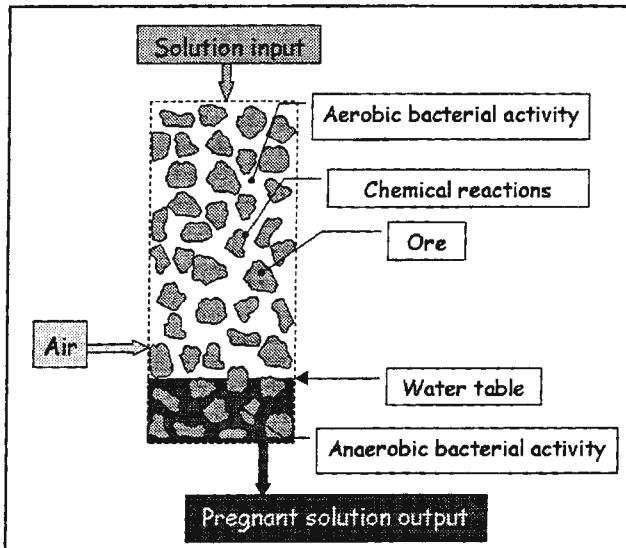
Fig. 1.

Vertical section through a heap

Fig. 2.

Typical flowchart for copper production based on heap leaching
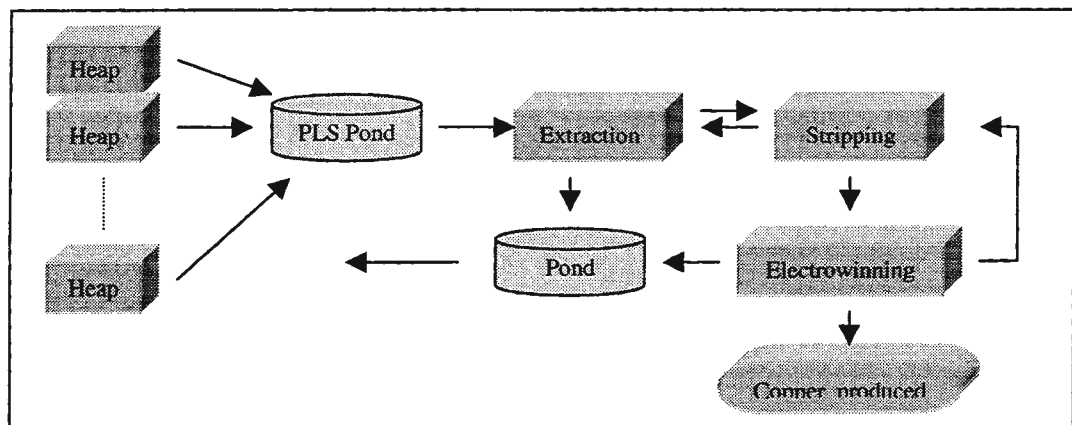
Modelling of the heap leaching process

The key-point of the heap leaching process is usually the heap itself. The study of the heaps can be divided in two main parts. First is the definition of the preprocessing steps, in order to prepare the ore for the heap leaching. The preprocessing step usually includes comminution, acid treatment, agglomeration and addition of biological nutrients. The second part is to determine the configuration of the run time leaching, including the parameters as heap dimension, geometry and aeration. The amount and concentration of the percolation acid should be also defined.

In practice, during the feasibility studies of a copper production plant, hundreds of tests are carried out with different ore preprocessing and acid percolation. These tests are performed in a small scale related to the desired heap size, usually in boxes and columns. The experts should define the right preprocessing parameters and then predict the behavior of the full-size commercial heaps based on the test results [5,6,7].

The most important is to predict the expected metal extraction versus time. Due to this reason, the dynamic calculations are also needed. Simulation provides a tool for these tasks.

For the metal extraction from the pregnant leach solution (PLS), simple and reliable models are available. This paper discusses the modelling and simulation of the heap leaching process. The percolation is assigned as an input, and the PLS is considered the output of this model. The input-output representation of the heap leaching process is presented in Figure 3. The PLS is a solution with the metal content, usually 5-10 gram/liter.
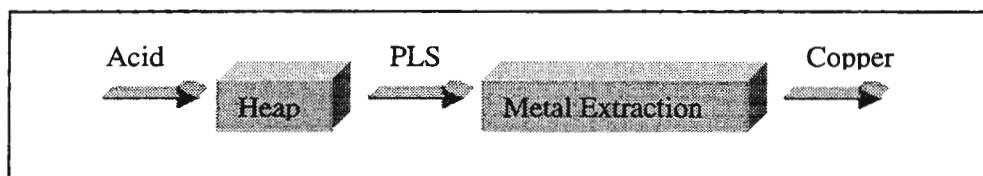
Fig.3. Input-output representation of the heap leaching process

The mathematical model has the preprocessing values (e.g. particle size distribution, temperature, pulp density, chemical and biological properties) assigned as model parameters. The configuration values of run-time leaching are also considered the model parameters. The parametric description of the heap leaching process may need two kinds of models, a *micromodel* that deals with the size changes of the solid particle during the leaching and a *macromodel* that deals with the system changes [8]. The changes in the size distribution and the shape of the particles in the heap, as result of the leaching process, are computed by means of the micromodel. The macroproperties as temperature, volume, pressure, pH and solution composition will change as well, therefore a macromodel is needed. The chemical reactions and the biological activity should be also included in the mathematical model.

Many approaches have been adopted to simulate the heap-leaching processes using biological models, which yields not accurate predictions due to the highly complicated calculations [9,10].

This research uses a parametric macro model, where chemical reaction kinetics is estimated based on the *column test data*. Biological factors and particle size changes are not directly included in calculations. All leaching characteristics, as ore crushing, comminution, agglomeration, pretreatment, biological factors, temperature and pressure are concentrated in the column test data. Several tests are carried out during the feasibility studies and each test yields an extraction curve. Based on the extraction curve, the estimation algorithm computes the kinetics of the dominant fast and slow chemical reactions. In this study, the abstraction has been made that the leaching conditions for the real full-size heap are same as for the test.

Simulation of a heap leaching experiment using Metsim

The flowsheet of the copper production based on heap leaching is presented in Figure 2. The process units in the production chain are the followings: *Heap, PLS Pond, Solvent Extraction and Electro Winning* and *Raffinate Pond*. The heap is percolated with lixiviant pumped from the raffinate pond. Fresh acid is added to the pond in order to keep the acid concentration in the lixiviant at constant level. The output of the heap is the PLS, stored in the PLS Pond. The Solvent-extraction and Electro-winning (SX-EW) are integrated into one process unit called *Phase-splitter*. This block has assigned as outputs the produced copper cathodes, gaseous Oxygen and the recycled lixiviant. The regenerated lixiviant is pumped back to the raffinate pond.

After the definition of the preprocessing steps, the column test is carried out. From the experimental data, the kinetic parameters of the undergoing chemical reactions are computed, using an approximation method. The column test configuration parameters for this simulation example are listed in the *Table 1*.

In the first step, the simulator calculates the amounts of fast and slow reacting copper with their reaction rates based on the column-test extraction curve. The kinetics of the undergoing main chemical reactions are described by integrated rate laws presented with the estimated kinetic parameters in Table 2. These values are further used to compute the dynamics of the chemical reactions. The simulated input and output curves are presented in the Figure 4, while the experimental data is presented in Figure 5. The computed copper extraction data are very close to the experimental values, and similar results were obtained in case of other heap leaching column test simulations as well. The research will continue with full size heap leaching simulations, where the complexity of the dynamic calculation is estimated to increase considerably.

Table 1. Configuration parameters of the heap leaching column test

Total mineral	0,95	[tons]
Acid cure	13	[kg/tons]
Initial copper content	1,02	[%]
Heap area	0,27	[m^2]
Height of lift	2,78	[m]
Leaching time	60	[days]
Dominant fast reaction	$CuO + H_2SO_4 \rightarrow CuSO_4 + H_2O$	
Dominant slow reaction	$3CuFeS_2 + H_2SO_4 + 8,5O_2 \rightarrow 2CuSO_4 + Fe_2(SO_4)_3 + H_2O$	

Table 2. Coefficients calculated from the column test data

Amount of fast reacting copper	A_1	69,46%
Amount of slow reacting copper	A_2	30,53%
Fast copper reaction rate	R_1	0,1581
Slow copper reaction rate	R_2	0,03406
Calculation formula	Recovery = $A_1[1-(1-R_1)^t] + A_2[1-(1-R_2)^t]$	
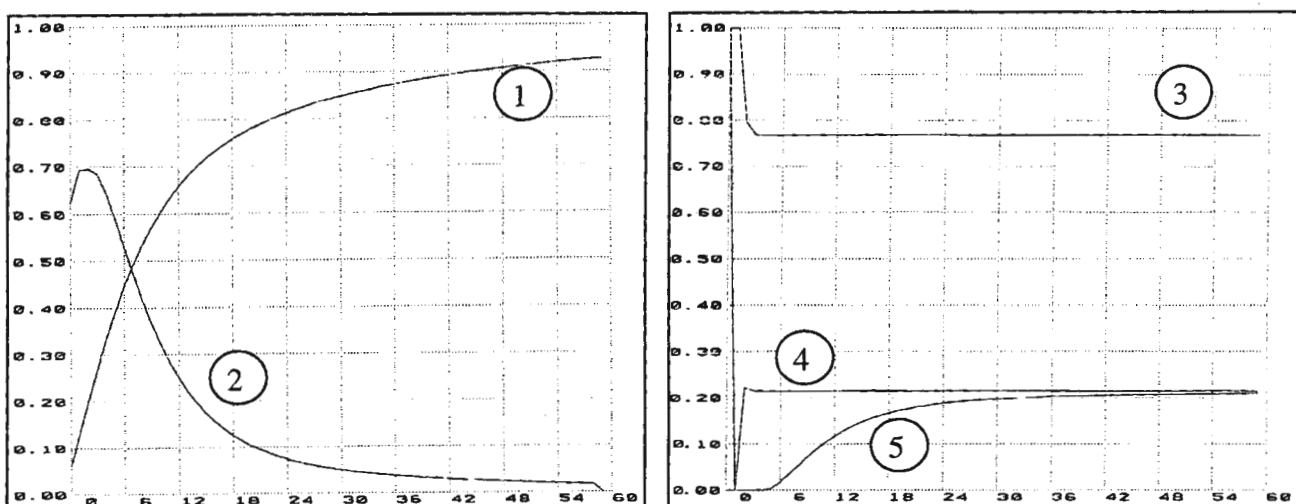
Fig. 4. Simulation results:

1. Total copper extraction   Range: 0..100 kg
2. Copper extracted per day   Range: 0..10 kg
3. Irrigation acid concentration   Range: 0..20 g/l
4. Irrigation acid mass   Range: 0..5 kg/day
5. PLS acid content   Range: 0..5 kg/day

Fig. 5. Experimental data
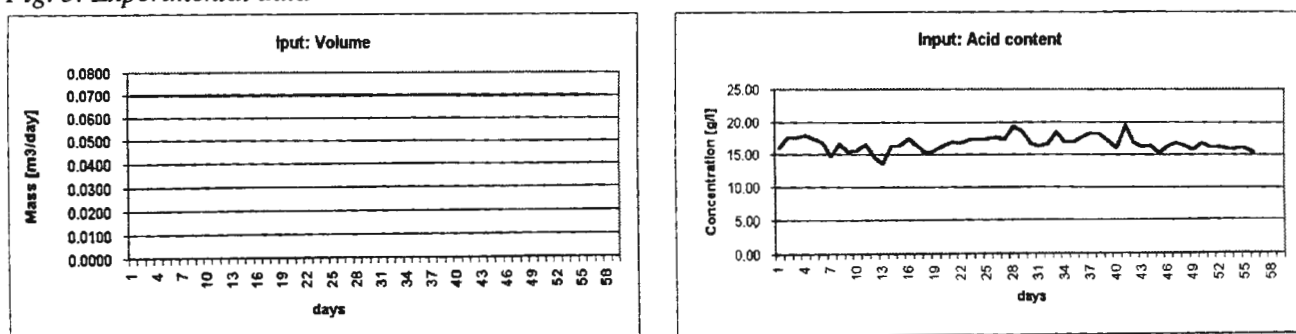
a. Irrigation acid volume   b. Irrigation acid concentration
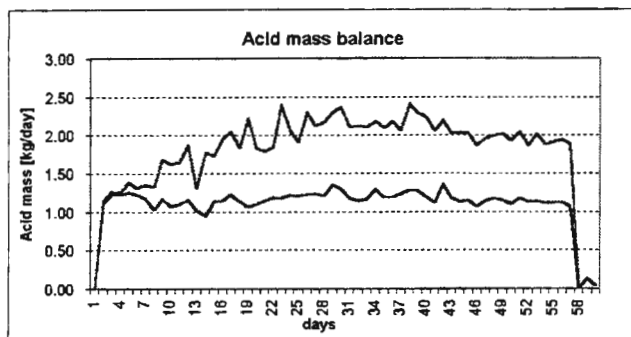
c. *Input and output acid mass*   d. *Copper extraction*
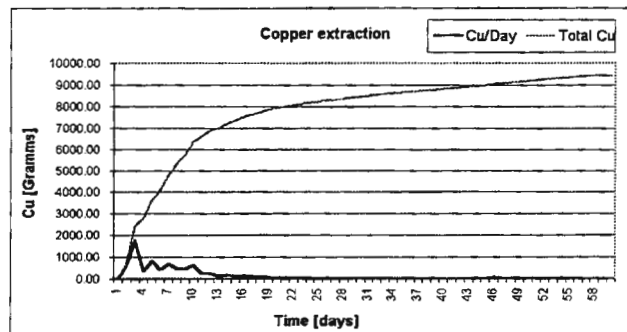
Experimental values contain oscillations, mainly caused by measurement errors, however the trend of the curves is same as the simulated ones. Due to the small size of the test column, the copper extraction was far to high, comparing with commercial size heap leaching. Unwanted side effects as channeling, blinding off and reaction rate modifications caused by temperature changes and uneven distribution could not happen, explained also by the relative small size of the experiment. These factors shall be taken in consideration during the full size heap leaching simulations.

The Metsim simulation software package provides a reliable calculation tool for the hydrometallurgical engineers, although the user interface is very old fashion and not handy at all. In case of a large flowsheet, the user can easily make a connection fault, and this task of numerically specifying the connections should be automated. These are the main sources of errors and waste of time. The software cannot be used in network and the chemical database is fixed. The printing is working with bad resolution and the installation of a new printer driver can easily fail. The monitor settings are old fashion and cannot use the high resolutions provided by the new monitors. The menu-driven interface nowadays is going to be replaced by dialog boxes and control fields. In many cases, if an input data is missing or having a value outside of the permitted range, error message is not generated. The mathematical models are making impossible the comparison of different models for the same operation unit. Although, already exists simulation software packages which satisfy most of these requirements the problem of creating customized mathematical models remains without answer.

General description of the new simulation software, SimCon

The simulator, under development, provides the tools, model libraries, database and solution algorithms for full-scale modelling and simulation of the hydrometallurgical processes. The mineral-processing engineer has to be able to represent the material flow including the chemical and physical properties and the operation units with their important parameters as well. The **flowsheet builder** module provides the basic graphical tools for the flowchart editing and data input. The graphical functionality is realized in this module. The representation of the **circulating mass** is implemented using *streams*, and their properties can be fitted by the user with the help of the "chemical data base". The database is an ordinary MS Access database file, and the connection is realized through 32 bit ODBC. For the **operation units** this simulator provides one comprehensive set of the mathematical models gathered in the "Unit library". The main modules of the simulator are presented in Figure 6. The data that are sampled from an industrial process never exactly satisfy the steady-state mass balances. The need exists for the mass balance calculation in the mineral and hydrometallurgical processes. For this purpose, the software is designed to include a mass-balancing module. During the whole development period, the objective was to highlight the features of the simulation environment considered important by simulation practitioners. The highest emphasis is on a consistent and user-friendly **graphical interface,** main weak point of the earlier simulators in the field of hydrometallurgy and mineral processing.

Modelling and computer simulation usually requires a large number of different alternatives and parameters. Often the variations in a design are relatively small and they need just minor extent. An important issue in simulation software design therefore, is to enhance the code and model reusability. Several approaches have emerged to support the software reusability. The simulator utilizes *object-oriented design and programming*

which, ensures that further development can be done without extensive programming work, just adding the desired new elements [1]. This is especially important regarding the purpose and future of the software.
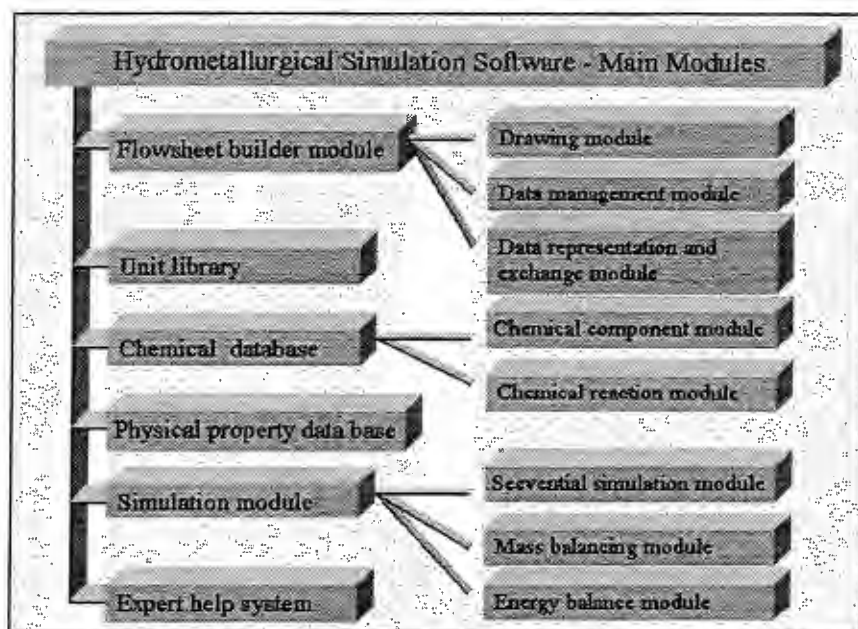

Fig. 6.

Main modules of the simulator

Conclusions

Simulation software has become a very important tool for the design and improvement of the mineral processing plant. The value of simulation is judged in the terms of its utility for the practicing engineers. However, the quality of the simulation software package can be measured through the incorporated models and the plant data available for the model calibration and parameter estimation. There exists the necessity for the development of fundamental mathematical models that can be customized to the specific applications.

It also provides the capability to conduct experiments on the simulator rather than at the plant. It rapidly assesses many possibles operating changes and identifies the one or two that are the most appropriate for the possible plant testing and assists in the evaluation of the equipment.

The presented simulation was based on real plant data. The simulated results are very close to the experimental data, which let us to conclude that simulation can be used as an efficient tool in the field of hydrometallurgy. Earlier software products usually have a very difficult user interface, therefore exists the need for a calculation tool with a user-friendly graphical interface combined with quick and reliable calculations.

References

[1]   *Praehofer,H.*, Object Oriented, Modular Hierarchical Simulation Modelling: Towards Reuse of Simulation Code, Eurosim no.17, Elsevier Science B.V., Amsterdam, (1996) , s.5-10

[2]   *Peters, E.*, Heap and dump leaching of copper, Copper'91 "Copper Hydrometallurgy Short Course", Ottawa, (1991), s.1-9.

[3]   *McCready,R..,G.,L.*, Bacterial leaching, Copper'91 "Copper Hydrometallurgy Short Course", Ottawa, (1991), s.2-13.

[4]   *Acevedo, F.*, Bioleaching of copper minerals, Copper'95 "Copper Hydrometallurgy Short Course", Santiago-Chile, (1995), s.1-6.

[5]   *Kelly,R..,J., Zárate, G.*, Heap leaching of copper oxide ores - Scale Up between Laboratory and Pilot Plant Results, Copper'95 - Cobre'95, Vol.3., Chile (1995), s. 743-756.

[6]   *Natunen, H.*, Copper Heap Leaching - Solvent Extraction - Electrowinning Process at the Zaldívar Mine, Copper'95 - Cobre'95, Vol.3., Chile, (1995), s. 757-772.

[7]   *Canello, R., Schnell, H.,A.*, Piloting of the Bacterial Leach Process at Quebrada-Blanca, Copper'95 - Cobre'95, Vol.3. Chile, (1995), s. 773-780.

[8]   *Peters, E.*, The Mathematical Modelling of Leaching Systems, JOM, February (1991), s. 20-26.

[9]   *Box, J.,C.*, A general model for the reaction of several mineral and several reagents in heap and dump leaching, Hydrometallurgy, 16, Elsevier Sc. Publishers, Amsterdam (1986), s. 77-92

SIMULATION OF THE CUMULATIVE CAPACITY OF CERAMIC ROTATING DISK FILTERS

Ville Kanninen

Helsinki University of Technology, Department of Chemical Technology, Laboratory of Process Control and Automation, Kemistintie 1, FIN-02015 TKK, Finland, Ville.Kanninen@hut.fi

1 Introduction

Ceramic rotating disk filters offer many advantages over traditional rotating disk filters. They require much less maintenance and their energy consumption can be up to 90 % lower than that with traditional disk filters, which together means low operating costs per produced concentrate tons. Ceramic filter technology is based on high-tech sintered alumina membranes, which have uniform micropores to create capillary action (Ekberg and Woitkowitz [1], Ekberg and Haartti [2], Lotzien et al. [4]).

Owing to the mechanical structure of the ceramic rotating disk filter, the basin slurry density can also undergo slow dynamic changes in cases, where the feed slurry density remains constant. This means that in order to determine the cumulative capacity of a ceramic filter during an operation period, dynamic changes have to be taken account. In this article, dynamic simulation is used to study the effects of slurry density changes on the cumulative capacity of a ceramic rotating disk filter. Simulation runs are used specifally to determine the effect of disk rotation speed and the effect of mounting gap length of filter cake scrapers on the cumulative capacity of an operation period.

2 Derivation of the model equations

2.1 Differential equations for basin slurry density and volume

Starting from the dynamic total mass balance of the basin (1), it is possible to solve the change in density to obtain a differential equation for basin slurry density (2).

$$\frac{d\left(V_{SLURRY}\,\rho_{SLURRY}\right)}{dt} = F_{SLURRY\atop FEED}\,\rho_{SLURRY\atop FEED} - F_{FILTRATE\atop WATER}\,\rho_{FILTRATE\atop WATER} + F_{WASH\atop WATER}\,\rho_{WASH\atop WATER} - \dot{M}_{CONCENTRATE} \tag{1}$$

$$\frac{d\rho_{SLURRY}}{dt} = \frac{1}{V_{SLURRY}}\left[F_{SLURRY\atop FEED}\,\rho_{SLURRY\atop FEED} + F_{WASH\atop WATER}\,\rho_{WASH\atop WATER} - F_{fILTRATE\atop WATER}\,\rho_{FILTRATE\atop WATER} - \dot{M}_{CONCENTRATE} - \rho_{SLURRY}\,\frac{dV_{SLURRY}}{dt} \right] \tag{2}$$

The dynamic volume balance of the basin is needed in order to take into account changes in slurry volume.

$$\frac{dV_{SLURRY}}{dt} = F_{SLURRY\atop FEED} - F_{FILTRATE\atop WATER} + F_{WASH\atop WATER} - F_{CONCENTRATE} \tag{3}$$

where the volumetric flow of the concentrate is as calculated as

$$F_{CONCENTRATE} = \frac{X_{CONCENTRATE}\dot{M}_{CONCENTRATE}}{\rho_{MINERAL}} + \frac{\left(1 - X_{CONCENTRATE}\right)\dot{M}_{CONCENTRATE}}{\rho_{WATER}} \tag{4}$$

2.2 Filtration equations

Equation (5) is a basic form of the simplified steady-state capacity model for the a rotating disk filter (Kanninen [3]). Equation (6) is a steady-state equation for the cake height in the rotating disk filter. These steady-state filter capacity models can be used in the dynamic simulation, because filtration phenomena occur much faster (time constant, one round time of the disks 5-90 seconds in the simulations) than changes in the basin slurry density (time constant in the range of 30-60 minutes in the simulations).

$$\dot{m}_{cake.dry} = N_{DISKS}\left(r_{outer}^2 - r_{inner}^2\right)\omega \frac{1}{\alpha}\left(-R_{disk} + \sqrt{R_{disk}^2 + \frac{4\alpha \Delta p C_{slurry} \arccos\left(\frac{-L}{r}\right)}{\mu \omega}}\right) \tag{5}$$

where

$$\alpha = \frac{1}{(1-\varepsilon)\rho_{solid} B_{cake}} \tag{6}$$

In deriving these models, it was assumed that in the filtering phenomena occuring on the ceramic plate surface, the division of slurry into solid and liquid fractions follows directly the ratio of solids to water in the slurry as defined by the pseudo-concentration C_{slurry}

$$C_{slurry} = \frac{\text{mass of solids in slurry } [\text{kg}]}{\text{volume of water in slurry } [\text{m}^3]} \tag{7}$$

The solid fraction is stratified on the surface of the cake, and the liquid fraction flows through the cake and ceramic plate. When the ceramic plate rises above the slurry, the formation of the solid cake ceases and the drying phase starts. As a result of capillary phenomena in the microporous ceramic plate, only water flows through the ceramic plates (there is no airflow). At the end of the drying phase, a small amount of water (5 % in these simulations) remains in the cake. This means that, in the separation, the solid fraction passes almost completely to the formed cake (solid concentration in the filtrate water flow is typically less than 50 ppm), and the major part of the liquid fraction passes to the filtrate water and the minor part to the formed cake. Using the assumption that division of the slurry into solid and liquid fractions follows directly the ratio defined by the pseudo-concentration C_{slurry} gives the following relationship between the steady-state liquid and solid flows

$$\frac{\dot{m}_{cake.dry}}{C_{slurry}} = F_{FILTRATE \atop WATER} + \frac{(1-X_{CAKE})\dot{m}_{cake}}{\rho_{WATER}} \tag{8}$$

The relationship between cake mass flow and dry cake mass flow before the scrapers is

$$\dot{m}_{cake.dry} = X_{CAKE}\,\dot{m}_{cake} \tag{9}$$

Combining equations (8) and (9) and solving for $F_{FILTRATE \atop WATER}$ gives

$$F_{FILTRATE \atop WATER} = \left[\frac{1}{C_{slurry}} - \frac{1-X_{CAKE}}{X_{CAKE}\rho_{WATER}}\right]\dot{m}_{cake.dry} \tag{10}$$

2.3 Scraper equation

In ceramic rotating disk filters, scrapers are used to remove the cake from the disks. To avoid mechanical abrasion, the scrapers are usually mounted such that there is a small gap between the scrapers and the filter disks. After cake removal, thin layer of residual cake is therefore left on the surface of the ceramic plates. The residual cake returns to the basin slurry as a result of ceramic plate backflow washing, which is performed after cake removal. The recycling cake flow causes a slow dynamic increase in slurry density to a new steady-state value, which is higher than that of the feed slurry density. On the other hand, because the backflow wash water dilutes the slurry, it is also possible to end up at a steady-state, where the basin slurry density is lower than the feed slurry density. Equation (11) takes into account the effect of scraper mounting gap on the simulated system.

$$\dot{m}_{concentrate, dry} = \left(\frac{h_{cake} - h_{scraper}}{h_{cake}}\right)\dot{m}_{cake, dry} \tag{11}$$

3 Simulation model

The simulation model was constructed and the calculations performed using commercial software, Matlab 5.1 and Simulink 2.1 in the Windows NT Workstation 4.0 operating system environment. The simulation model is divided into several subsystems (figure 1).
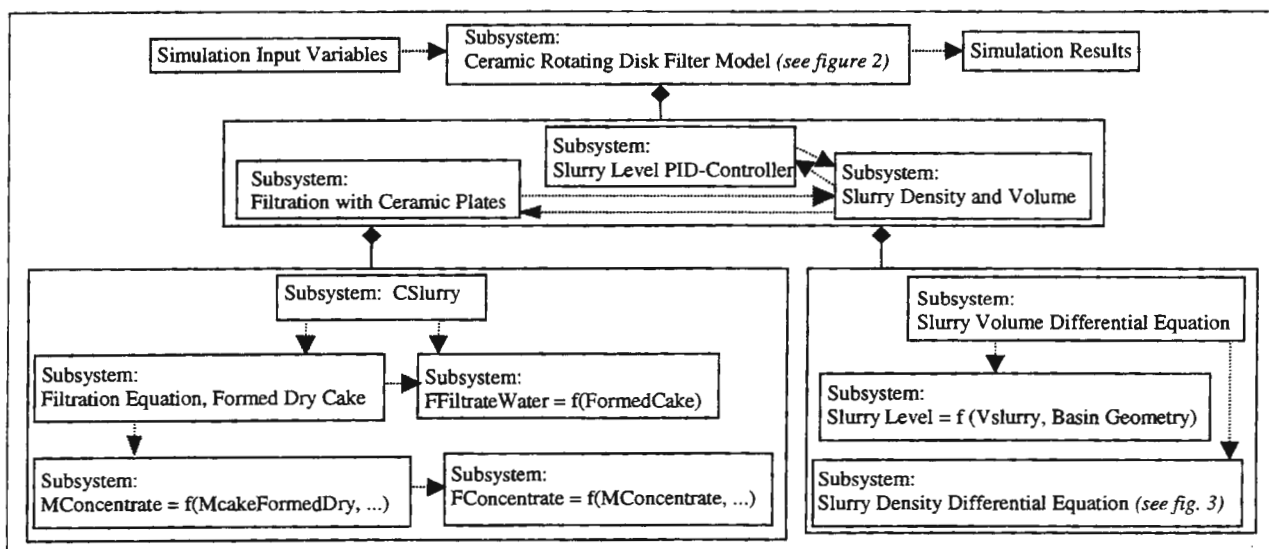
Figure 1. Simplified overview of the Simulink simulation model structure

Figures 2 and 3 are presented as examples of the Simulink subsystem block diagrams used in the model. Figure 2 shows the connections between the filtration model, basin slurry model and slurry level PID controller. Figure 3 presents the slurry density differential equation subsystem.
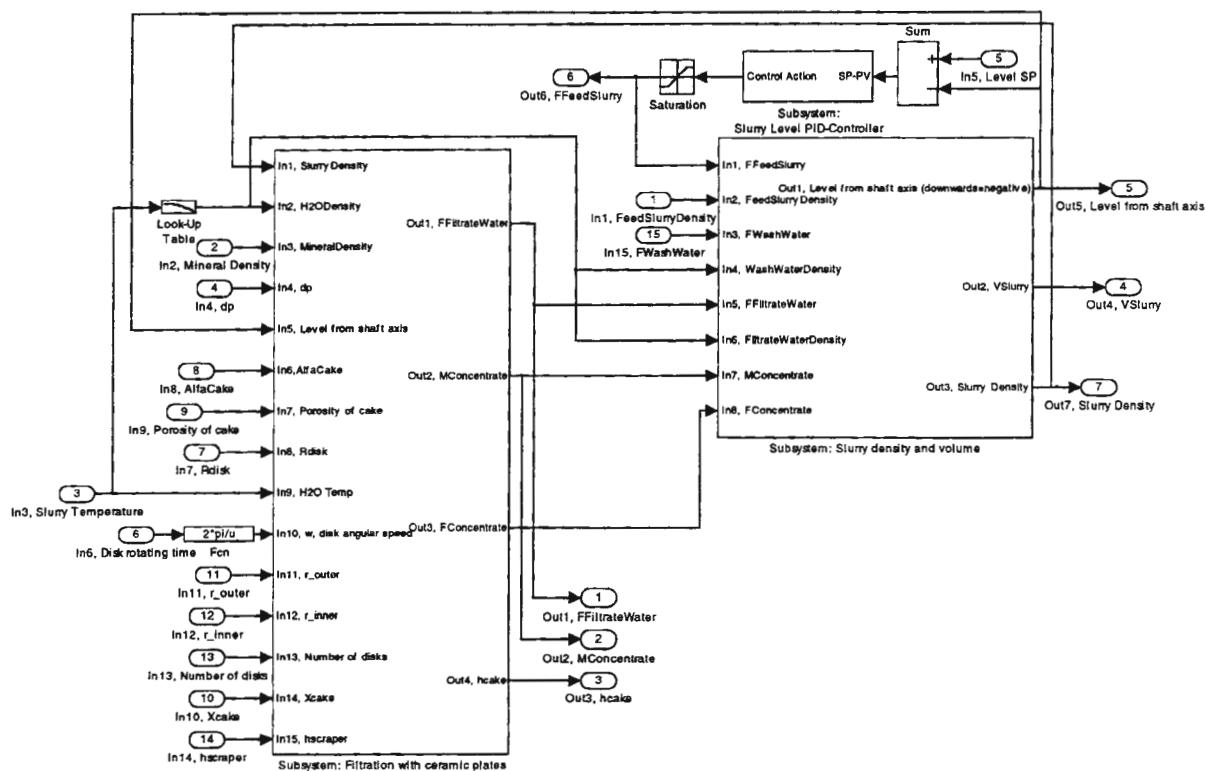
Figure 2. Connections between the filtration model and slurry density and volume model

dSlurryDensity/dt =
(1/Vslurry) * (FFeedSlurry*FeedSlurryDensity + FWashWater*WashWaterDensity - MConcentrate - SlurryDensity*dVSlurry/dt
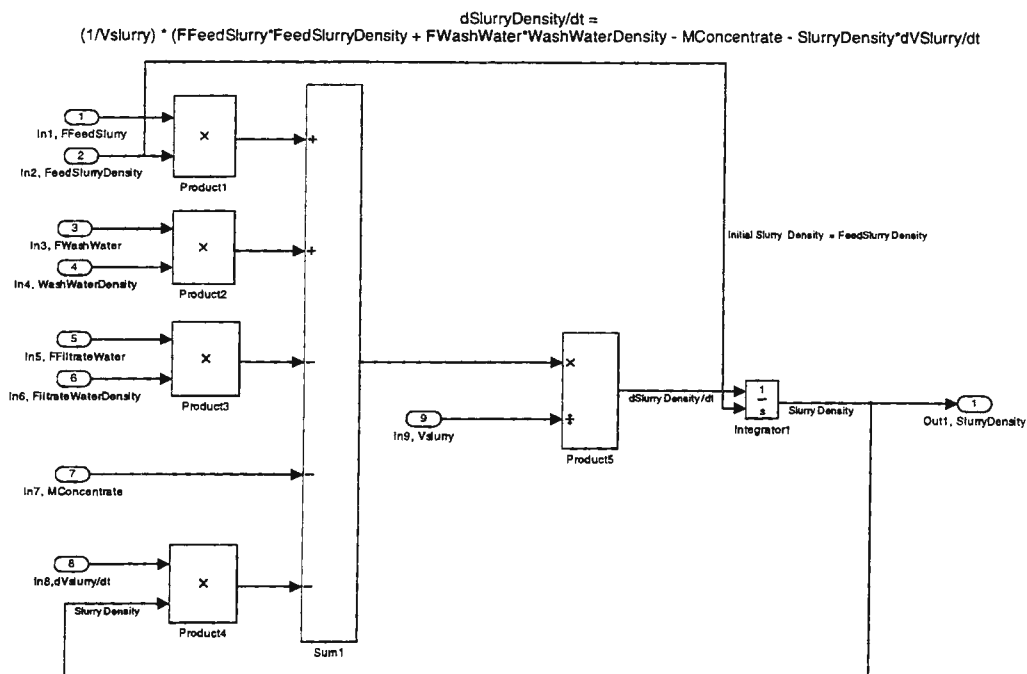
Figure 3. Simulink block diagram of the slurry density differential equation subsystem

4 Simulation results

4.1 Input data

Two variables – the disk round time and the scraper mounting gap – were varied in the simulation test runs, all the other operation variables being kept constant. The values of these constant parameters are given in table 1. Length of the operation period to be simulated was selected to be 12 hours. Altogether 66 simulation runs have been made with the variable ranges shown in tables 2 and 3. Simulation of one 12 hour operation period required about 11 minutes 36 seconds calculation time on a PC equipped with an Intel Pentium Pro 200 MHz CPU, 128 MB of RAM and Windows NT Workstation 4.0. The total calculation time required was thus about 12 hours 45 minutes.

Table 1. Values of the variables and parameters kept constant during the simulations

Parameter	Value
1) The ceramic filter dimension parameters:	
Number of the disks in the filter	5
Outer radius of the disk	0.951 m
Inner radius of the disk	0.635 m
2) Operation variables:	
Vessel pressure	280 kPa
Slurry level set point	0.2 m downwards from the shaft axis
Volume of the slurry at set point	2.676 m^3
3) Slurry parameters:	
Temperature of the slurry	20 ° C
Viscosity of water at 20 ° C	0.0010012 kg/m/s
Density of water at 20 ° C	998.204 kg/m^3
Density of the mineral particles	6500 kg/m^3
Feed slurry density	2500 kg/m^3 (70.97 weight-% of solids)
4) Cake parameters:	
Porosity of the cake	0.5
Solid mass fraction of the cake	0.95
Specific resistance of the cake	2E+10 kg/m
5) Ceramic plate parameters:	
Resistance of the ceramic plates	6E+11 1/m

Table 2. Value ranges of the variables varied in the basic simulations

Variable	Number of test values	Used test values
Disk round time	10	5, 10, 15, 20, 25, 30, 35, 40, 45, 50 [sec]
Scraper gap	5	0.0, 0.5, 1.0, 1.5, 2.0 [mm]
Time of the operation period	1	12 hours
Number of the simulation runs	10 x 5 x 1 = 50	

Table 3. Additional value ranges of the variables varied in the simulations for figures 10 and 11

Variable	Number of the test values	Used test values
Disk round time	8	55, 60, 65, 70, 75, 80, 85, 90 [sec]
Scraper gap	2	0.5, 1.0 [mm]
Time of the operation period	1	12 hours
Number of the simulation runs	8 x 2 x 1 = 16	

4.2 Dynamic changes of the simulation output variables

Basin slurry density and the other simulation output values reached their steady-state values during the first two hours of the 12-hour operation period. The time needed to reach the new steady-state values varied according to the magnitude of the feed slurry flow and therefore according to differences in water removal and produced concentrate capacity. Figures 4,5,6 and 7 illustrate the results of a simulation run in which the disk round time was 25 seconds and the scraper mounting gap 1.0 mm.
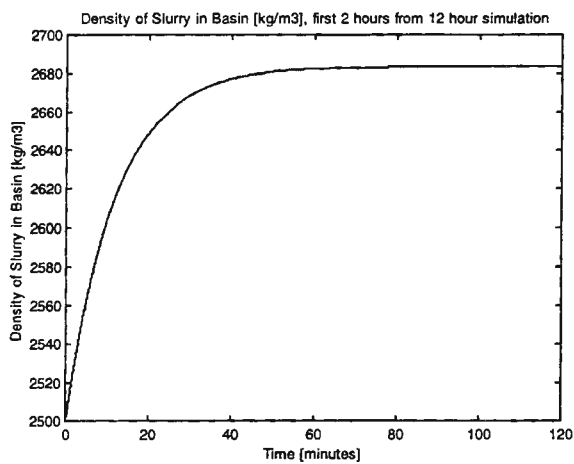
Figure 4. Basin slurry density [kg/m3], first two hours of 12-hour operation period
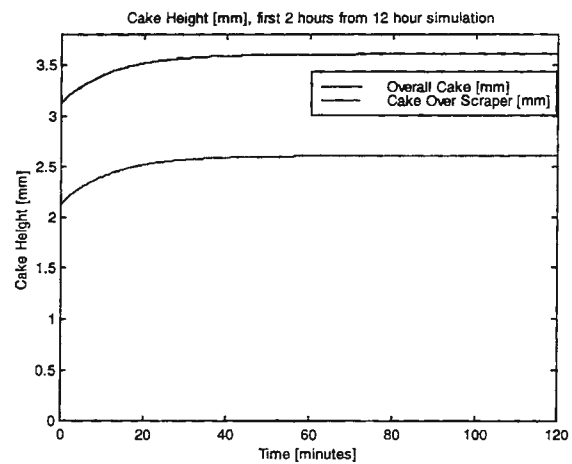
Figure 5. Cake height [mm], first two hours of 12-hour operation period
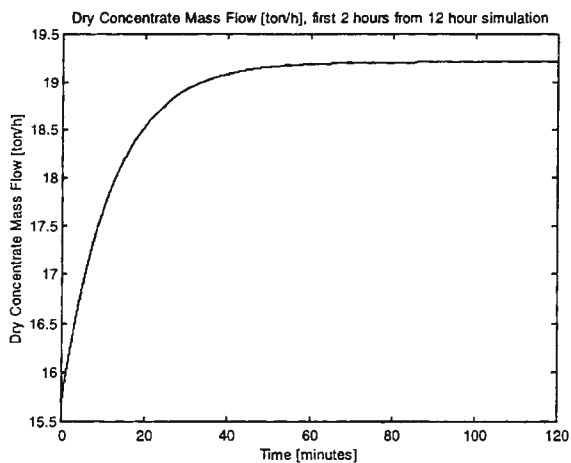
Figure 6. Dry concentrate mass flow [ton/h], first two hours of 12-hour operation period
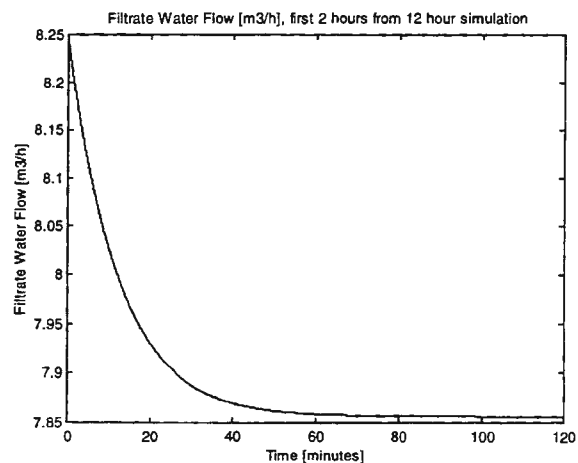
Figure 7. Filtrate water flow [m³/h], first two hours of 12-hour operation period

4.3 Cumulative concentrate capacity of 12-hour operation period

The cumulative concentrate capacity during a 12-hour operation period is shown in figures 8 and 9 as a function of scraper gap and disk round time. Figure 8 presents the results in 3-dimensional form and figure 9 in 2-dimensional form. If there is no scraper gap at all, increasing the rotation speed (same as decreasing the disk round time) increases the cumulative capacity monotonously. In the region of extremely high rotation speeds, there is probable some limit to the increase in capacity, because turbulence effects probably start to decrease the filtrate water flow. It is not probable, however that such a limit will be reached with the rotating speeds that are mechanically reasonable. With all scraper gap values higher than zero, increasing the rotation speed increases the cumulative capacity up until maximum point is reached, after which the cumulative capacity starts to decrease. The rotation speed value for the maximum cumulative capacity depends on the scraper gap length and also on many other process variables, like feed slurry density, disk and cake resistance and cake moisture (which are kept constant in the simulations shown in these figures).
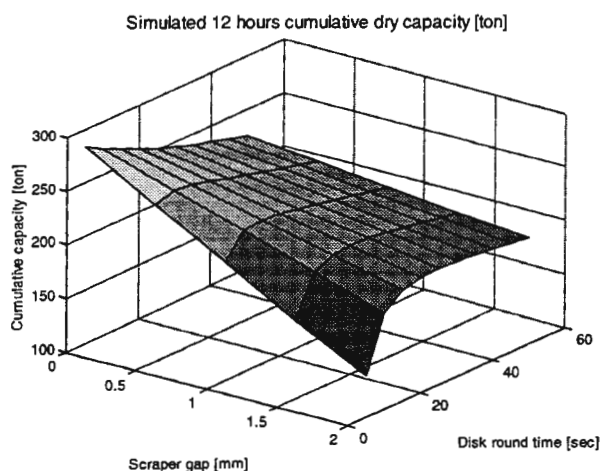
Figure 8. Cumulative dry concentrate capacity of 12-hour operation period as function of scraper gap and disk round time
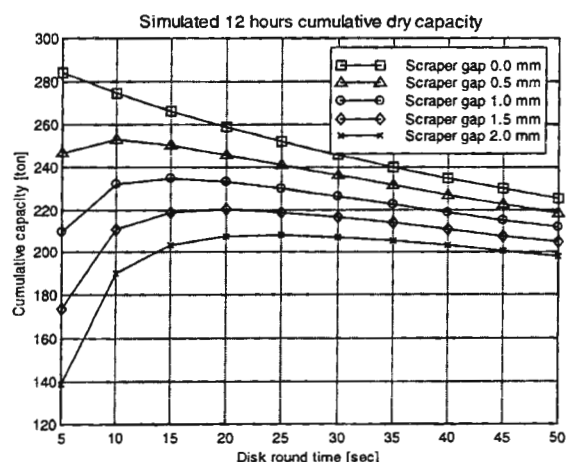
Figure 9. Same data as in figure 8 presented in 2-dimensional form

4.4 Momentary concentrate capacity compared to the average concentrate capacity of a 12-hour operation period

Figures 10 and 11 show the momentary and average (cumulative capacity divided by operation period length) dry concentrate capacities as a function of disk round time with scraper gap values of 0.5 mm and 1.0 mm. The momentary capacities have been calculated using feed slurry density (2500 kg/m^3) as the basin slurry density value.

As can be seen from figures 10 and 11, the maximum cumulative (or average) capacity of an operation period is reached at clearly faster rotating speeds (or smaller disk round times) than the maximum momentary capacity. This means that selecting optimal rotation speed based on short period plant tests can give rather misleading results.
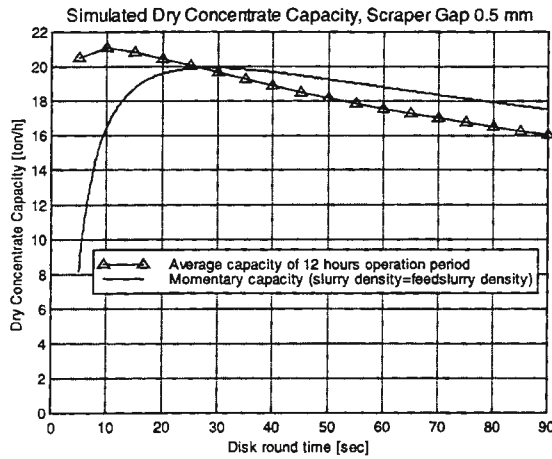
Figure 10. Momentary capacity compared to average capacity of 12-hour operation period, scraper gap 0.5 mm
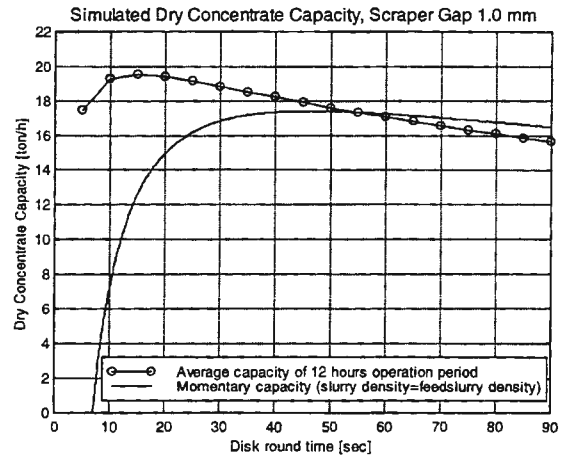
Figure 11. Momentary capacity compared to average capacity of 12-hour operation period, scraper gap 1.0 mm

5 Conclusions

5.1 Effects of scraper mounting gap and disk rotating speed on the cumulative capacity of the ceramic rotating disk filter

Table 4 lists three different mechanisms or reasons for the change in cumulative dry concentrate capacity when the values of the scraper gap or disk rotating speed are changed. If the operation period is long enough compared to the basin slurry density time constant as was the case in all the simulations presented in this article, then reason 1 (lower momentary capacity in the beginning of an operation period) has only a very small effect on the cumulative capacity of an operation period. Here operation period means a period after which the basin is drained e.g. for automatic washing of the filter. Reason 2 in table 4, cake resistance (to filtrate water flow), which is proportional to the cake height, is the most important factor explaining the changes in the cumulative capacity when the scraper gap or disk rotating speed is changed. Reason 3 starts to be important when the rotation speed is extremely high and the scraper gap is greater than zero.

Table 4. Different reasons for the change in cumulative dry concentrate capacity when the values of the scraper gap or disk rotating speed are changed

Mechanism or reason for the change in the cumulative capacity / Change in input variable of simulation	Increasing scraper gap	Increasing disk rotating speed
1 In the beginning of an operation period the momentary capacity is lower until new higher slurry density, cake height and concentrate capacity steady-state values are reached.	Decreases cumulative capacity	Decreases cumulative capacity
2 Higher cake means increased cake resistance, which results in decreased filtration rate (when pressure difference is constant), i.e. decreased water-removing capacity.	Decreases cumulative capacity	*Increases* cumulative capacity
3 If disk rotating speed is very high (and scraper gap > 0 and cake resistance > 0 and cake moisture > 0), then even with the higher slurry density steady-state value reached, the recycled part of the cake (cake under scraper) is high compared to the concentrate part of cake. This recycled cake also contains moisture, which is thus not passing out of the filter as it would if the scraper gap was zero (or cake moisture was zero). When the recycled part of cake is higher, then total removed water from the filter is lower, i.e. the water removing capacity is decreased.	Decreases cumulative capacity	Decreases cumulative capacity
Total effect of all 3 mechanisms or reasons	Decreases cumulative capacity	*Increases* cumulative capacity until maximum is reached

387

5.2 Possibilities to improve the simulation model

To determine the effects of dynamic slurry density changes on the cumulative capacity, the effects of ceramic plate fouling during an operation period have not been studied in these simulations, but disk resistance has been kept at a constant average value. For this reason, the dynamic simulation results deviate somewhat from typical plant data. The next step in improving the model would be to include the fouling effects in the simulation model by using an empirical model or typical collected time series data. Another constant parameter in the model is cake moisture. Adding a cake moisture model, even a very rough empirical one, would also increase the usefulness of the simulation model.

Acknowledgements

Notation

$C_{slurry} = \dfrac{\text{mass of solids in slurry } [kg]}{\text{volume of water in slurry } [m^3]}$

$F_{CONCENTRATE}$ = volumetric flow of concentrate $[m^3/h]$

$F_{FILTRATE\,WATER}$ = filtrate water flow $[m^3/h]$

$F_{SLURRY\,FEED}$ = feed slurry flow $[m^3/h]$

$F_{WASH\,WATER}$ = wash water flow $[m^3/h]$

h_{cake} = height of the cake $[m \text{ or } mm]$

$h_{scraper}$ = height of scraper gap $[m \text{ or } mm]$

L = level of slurry from axis (downwards = negative) $[m]$

$\dot{m}_{cake.dry}$ = mass flow of dry cake

$\dot{m}_{concentrate.dry}$ = mass flow of dry concentrate

$\dot{m}_{concentrate..wet}$ = mass flow of wet concentrate

N_{DISKS} = Number of disks in the ceramic filter

Δp = pressure difference over the cake and disk $[Pa \text{ or } kPa]$

r = radius $[mm \text{ or } m]$

r_{outer} = outer radius of disk $[mm \text{ or } m]$

r_{inner} = inner radius of disk $[mm \text{ or } m]$

R_{disk} = resistance of disk $[1/m]$

V_{SLURRY} = Slurry volume in the basin of the ceramic filter

X_{cake} = mass fraction solids in cake

α = spesific resistance of the cake $[m/kg]$

B_{cake} = permeability of the cake $[m^2]$

ε = porosity of cake

ρ_{slurry} = density of slurry $[kg/m^3]$

ρ_{solid} = density of mineral solids $[kg/m^3]$

ρ_{water} = density of water $[kg/m^3]$

μ = viscosity of filtrate water $[kg/m/s]$

ω = angular speed of rotating disk $[rad/s]$

References

[1] Ekberg, B., Woitkkowitz, S., Cake-forming wet filtration without gas throughput with the new ceramic Ecosuc disc filters, *Aufbereitungs-Technik*, **28** (1988), Nr. 4, 193-196.

[2] Ekberg, B., Haartti, J., Capillary Filtration – The Harnessing of Natural Forces to Serve the Process Industry, *XVIII International Mineral Processing Congress, 23-28 May 1993, Sydney, Australia*, 1303-1308.

[3] Kanninen, V., Capacity Model of Disk Filters, *8th IFAC Symposium on automation in Mining, Mineral and Metal Processing, 29-31 August 1995, Sun City, South Africa*.

[4] Lotzien, R., Jung, R.G., Janitschek, R., Filtration without air throughput of filter material, *Aufbereitungs-Technik*, **28** (1988), Nr. 11, 628-636.

TRANSIENT KINETIC MODELING OF ADSORPTION

Jaana Kanervo[1], Leif Backman[2], Sirkka-Liisa Jämsä-Jounela[3], Outi Krause[4]

[1]Helsinki University of Technology, P.O.Box 6100, FIN-02015 HUT, jojanen@cc.hut.fi
[2]Helsinki University of Technology, P.O.Box 6100, FIN-02015 HUT, backman@polte.hut.fi
[3]Helsinki University of Technology, P.O.Box 6100, FIN-02015 HUT, sirkka-l@cc.hut.fi
[4]Helsinki University of Technology, P.O.Box 6100, FIN-02015 HUT, krause@polte.hut.fi

Introduction

Transient response methods and their applicability in heterogeneously catalyzed reaction kinetics are studied in this research [1]. In transient kinetic studies a dynamic change is introduced into a reactor system, and its response in some reaction quantity is measured. A dynamic change can be introduced in flow rate, pressure, temperature or inlet concentration. Usually in heterogeneously catalyzed reactions concentration signals - steps or pulses - are imposed on the inlet stream under isothermal conditions. Experiments usually employ flow reactors. The measured quantity is outlet concentration. Transient responses can be analyzed in order to obtain more kinetic information than is possible using traditional steady-state experiments. [2]

This paper begins with the model formulation and the solution scheme used in the estimation of the model parameters. This is followed by examples of applications. The dynamics of hydrogen and methylcyclohexane adsorption are modeled using concentration pulse experiments. These adsorption phenomena are considered because the final goal, the gas phase hydrogenation of toluene, is assumed to contain these elementary steps.

Model formulation

Modeling is started by postulating a hypothetical reaction mechanism by dividing the reaction into elementary steps containing adsorption, surface reaction and desorption steps. The rates of the elementary reactions are given by the law of mass action. In heterogeneously catalyzed reactions at least one reactant is assumed to chemisorb on the catalyst. The dynamic mass balances for the test reactor are formed with respect to each species involved in reaction.

The reaction kinetic expression is embedded in the total mass balance. If we consider a packed bed reactor and assume that the catalyst bed is isothermal and the gas velocity is constant in time and place the model equation for the gas phase species is [3]:

$$\frac{\partial c}{\partial t} = -v\frac{\partial c}{\partial z} + r_v \frac{1-\varepsilon_b}{\varepsilon_b} \tag{1}$$

where c is concentration in the bulk phase (mol m^{-3}), t is time (s), z is the axial distance along the catalyst bed (m), r_v is the volumetric rate of reaction (mol m$^{-3}_{cat.bed}$ s^{-1}) and ε_b is the void fraction of the catalyst bed. This mass balance excludes other mass transport phenomena than convection. The mass balance for adsorbed, non-mobile species becomes [3]:

$$\frac{\partial c_{y*}}{\partial t} = r_v(y*) \tag{2}$$

where c_{y*} is the concentration of adsorbed y (mol m$^{-3}_{cat\,bed}$).

Usually, a model consists of a system of several partial differential equations. The number of dynamic equations is the sum of all gas phase and adsorbed species. In order to gain a unique numerical solution, additional conditions must be defined. The number of required initial conditions corresponds to the number of all model equations. The number of necessary boundary conditions is equal to the number of different gas phase species. When fitting a model to a set of data all the additional conditions must be defined according to the experimental conditions.

Solution scheme

As described earlier, the model consists of the transient mass balances of each species in a reactor. The mathematical models include the kinetics of the hypothetical elementary steps. In transient experiments the model describes the behavior of the concentration of one or more components in time and space. Such a partial differential equation model can be solved by a suitable pde-solver, but the determination of unknown parameters still remains. There are several methods to determine the unknown parameters and to fit the model to experimental data.

The non-linear least squares method consists of minimizing the difference between the data predicted by the model (C^{cal}) and the experimental data (C^{obs}). This minimization criteria, the sum of squares of residuals is given by (3):

$$SSR = \sum_{n=1}^{N} \sum_{m=1}^{M} (C_{n,m}^{obs} - C_{n,m}^{cal})^2 \qquad (3)$$

where n is the component number; m is the observation number, N is the total number of observable components, M is the total number of observations, C^{obs} is the observed outlet concentration and C^{cal} is the outlet concentration given by the model.

In heterogeneously catalyzed systems the concentrations of adsorbed species cannot be measured therefore a few of the system variables are not observable. Thus the model fitting is based on the behavior of the gas phase species. Figure 1 illustrates the flow chart of the solution scheme. The minimization procedure is iterative. The minimization routine starts with an initial guess for the parameters given by the user. Every time the routine needs to evaluate the object function (3) for a given new estimate, the pde-solver is called and the model is evaluated for this estimate. When evaluating the model, necessary input variables are set equivalent to those used in the experiments. The sum of squares of residuals is then calculated from the model data and the experimental data. The minimization procedure continues and generates new parameter estimates until the convergence criteria are reached. [4]
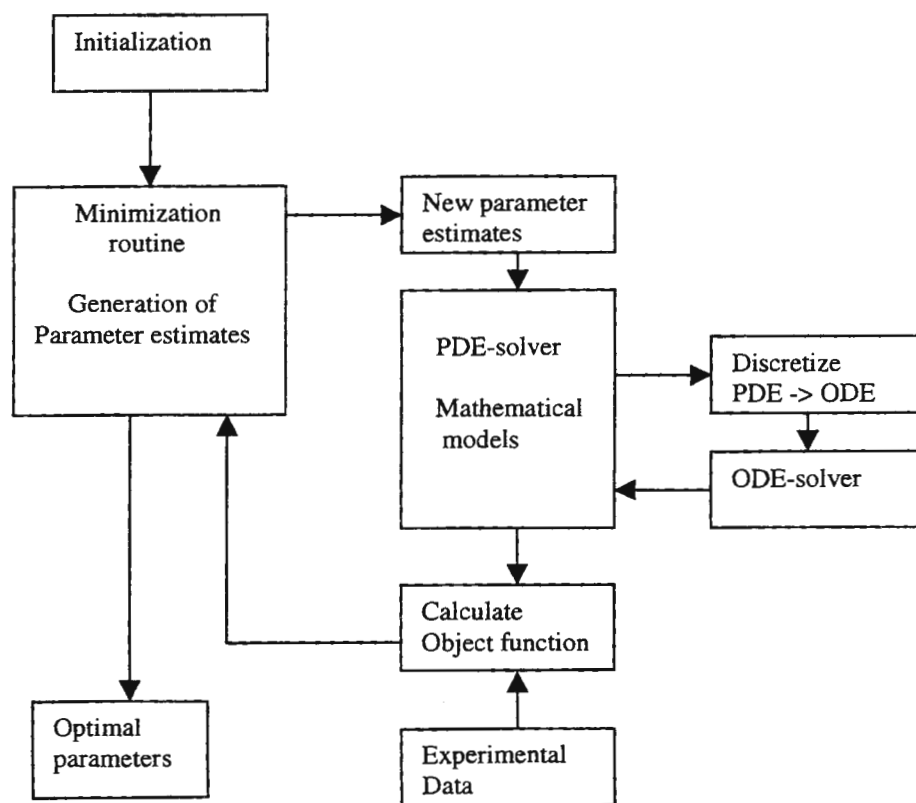
Figure 1. Flow chart of the solution scheme. [4]

All computations were implemented in the Matlab environment. The minimization task is unlimited nonlinear minimization with respect to the reaction rate parameters. The most time consuming part of calculating the object function is the solving of the pde-system, which is done on every iteration step. The optimization procedure fminu was selected among the Matlab optimization toolbox functions and its function is based on the quasi-Newton algorithm. The pde-solver was chosen from commercial NAG-fortran library. This NAG-routine, d03pef was made runnable in Matlab by utilizing the mex-file mechanism. Some minor m-files were written in order to complete the whole solution scheme.

Experiments and kinetic studies

The final goal is to analyze the mechanism of the hydrogenation of toluene. The reaction mechanism was assumed to consist of the adsorption of both reactants, toluene and hydrogen, a series of elementary steps on the catalyst surface and the desorption of the product, methylcyclohexane. All of these elementary steps could be assumed to be of equilibrium nature. Thus the mathematical models derived from these hypothetical mechanisms contained 11-13 model parameters and only a part of the system variables were observable. The direct applying of transient techniques was not expected to give unique, physically meaningful values for the reaction parameters. Consequently, as many elementary reactions as possible were to be analyzed separately by transient kinetic techniques. This paper covers the adsorption studies of hydrogen and methylcyclo-hexane.

Adsorption of hydrogen

If we assume dissociative, Langmuir-type adsorption for hydrogen on the nickel catalyst:

$$H_2 + 2^* \leftrightarrow 2 H^*,$$

the model for a packed bed reactor is given by (4):

$$\frac{\partial C_{H_2}}{\partial t} = -v\frac{\partial C_{H_2}}{\partial z} + (k_D C_{H^*}^2 - k_A C_{H_2} C_v^2)\frac{1-\varepsilon_b}{\varepsilon_b}$$

$$\frac{\partial C_{H^*}}{\partial t} = 2\left(k_A C_{H_2} C_v^2 - k_D C_{H^*}^2\right)$$

(4)

where C_v is the concentration of vacant active sites on the catalyst, k_D is the desorption rate constant and k_A is the adsorption rate constant. It is convenient to modify the system of equations (4) into a form where concentrations are expressed as fractions of total concentrations. For the gas phase we apply the ideal gas law $C_{H2} = x_{H2}*p_{tot}/RT$. Surface concentration is written as fractional coverage of $C_{H^*}= N_{tot} \theta_{H^*}$, where $N_{tot} = C_v + C_{H^*} \Leftrightarrow \theta_v = 1 - \theta_{H^*}$. The modified system becomes:

$$\frac{\partial x_{H_2}}{\partial t} = -v\frac{\partial x_{H_2}}{\partial z} + (k_D N_{tot}^2 \theta_{H^*}^2 \cdot \frac{RT}{p_{tot}} - k_A x_{H_2} N_{tot}^2 (1-\theta_{H^*})^2)\frac{1-\varepsilon_b}{\varepsilon_b}$$

$$\frac{\partial \theta_{H^*}}{\partial t} = 2\left(k_A N_{tot} x_{H_2} (1-\theta_{H^*})^2 \frac{p_{tot}}{RT} - k_D N_{tot} \theta_{H^*}^2\right)$$

(5)

For a unique solution the required boundary and initial conditions are:

$$x_{H_2} = x_{H_2}(0,t) = f(t) \quad x_{H_2}(z,0) = 0 \quad \theta_{H^*}(z,0) = 0$$

(6)

391

The conditions above correspond to the case of imposing a concentration pulse of hydrogen on the carrier gas when no hydrogen species exist initially in the reactor. The function f(t) is the time dependency of the hydrogen fraction at the reactor inlet.

The model system (5) has been simplified by an assumption that there are only one type of active sites. Temperature programmed desorption studies have demonstrated the surface heterogeneities of the nickel catalysts: hydrogen exists in different adsorption states (I-III), state I being responsible for the main part of the adsorption [5].

The experiments were carried out in the Laboratory of Industrial Chemistry at the Helsinki University of Technology using a pulse microcatalytic reactor. The outlet concentrations were analyzed by a mass spectrometer. Figure 2 presents graphically an example of input pulses and pulse responses. The pulses were injected into the reactor every 180 second. The initial fractional coverage of hydrogen was found to be non-zero at the beginning of later pulses due to this cycle time. The estimations were however done by using the data of later pulses, because dynamic phenomena of the very first pulses were found to be inexpressible with the model (5). The initial condition for the fractional coverage of hydrogen was set as a new free parameter in optimization.

The parameter estimation was performed by fitting the output of the model (5) to two sequential experimental pulse responses according to the solution scheme described earlier. The parameter estimates for the adsorption rate constant, the desorption rate constant and the number of active sites are given in table 1 for three different temperatures. Figure 3 illustrates the experimental data points and the output of the fitted model.
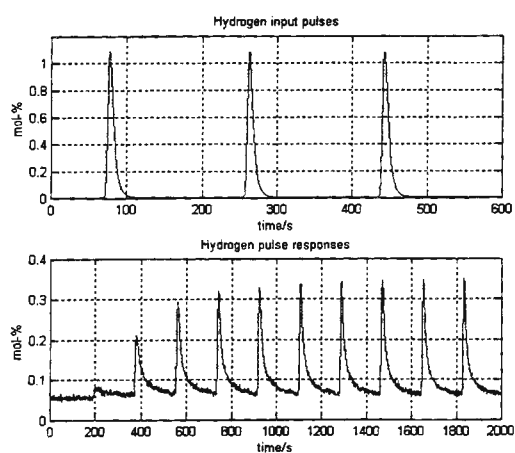
Figure 2. Hydrogen input and output data.
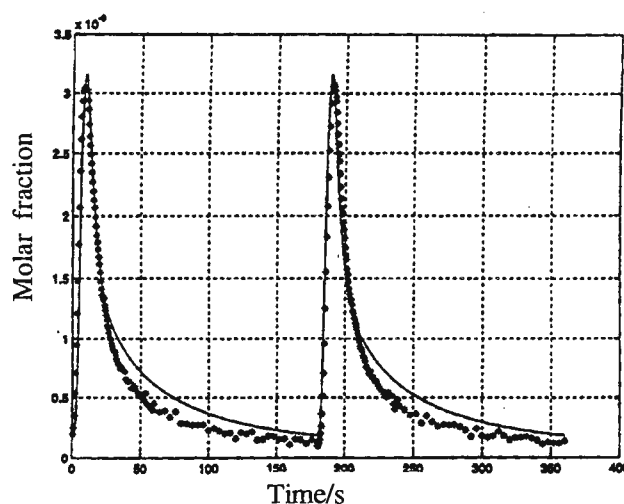
Figure 3. Model and experimental data points in hydrogen adsorption.

Table 1. Model parameters for the adsorption of hydrogen.

| T	$k_a/m^6mol^{-2}s^{-1}$	$k_d/m^3mol^{-1}s^{-1}$	$N_{tot}/mol/m^3_{cat}$	θ_0	SSR
150 °C	$2.09 * 10^{-4}$	$2.14 * 10^{-5}$	650	0.19	$6.6*10^{-6}$
170 °C	$2.27 * 10^{-4}$	$3.21 * 10^{-5}$	711	0.15	$4.8*10^{-6}$
190 °C	$3.30 * 10^{-4}$	$3.79 * 10^{-5}$	675	0.15	$3.7*10^{-6}$

According to the results, the model is in good agreement with the experimental data. All the values of the rate constants increase with temperature. The rate constants do not fulfill the Arrhenius law satisfactorily, but on the other hand the temperature scale is quite narrow.

Adsorption of methylcyclohexane

In the methylcyclohexane (MCH) adsorption experiments a constant hydrogen flow was used in addition to the MCH pulses in order to prevent the dehydrogenation of MCH. The qualitative preliminary study on MCH adsorption implied that hydrogen atoms and MCH molecules might adsorb competitively on same type of active sites. The model of adsorption in this case was to include dynamics of both adsorption phenomena. One issue under consideration was, how many active sites a MCH molecule uses on catalyst compared with a hydrogen atom.

The mechanisms for the adsorption of methylcyclohexane, if the adsorption is assumed to be competitive with the hydrogen adsorption and the desorption of methylcyclohexane gives X vacant sites, are:

$$MCH + X^* \overset{k_{A2}}{\underset{k_{D2}}{\leftrightarrow}} MCHX^*$$

$$H_2 + 2^* \overset{k_{A1}}{\underset{k_{D1}}{\leftrightarrow}} 2H^*$$
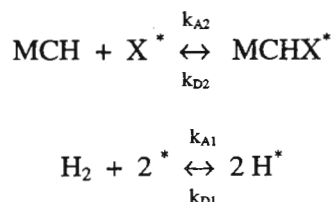
The system of equations for a packed bed reactor when adsorption kinetics are based on the mechanism above is given by (7):

$$\frac{\partial x_{H_2}}{\partial t} = -v\frac{\partial x_{H_2}}{\partial z} + (k_{D1}N_{tot}^2\theta_{H^*}^2 \cdot \frac{RT}{p_{tot}} - k_{A1}N_{tot}^2 x_{H_2}(1-\theta_{H^*} - X\theta_{MCHX^*})^2)\frac{1-\varepsilon_b}{\varepsilon_b}$$

$$\frac{\partial \theta_{H^*}}{\partial t} = 2\left(k_{A1}N_{tot}x_{H_2}(1-\theta_{H^*} - X\theta_{MCHX^*})^2\frac{p_{tot}}{RT} - k_{D1}N_{tot}\theta_{H^*}^2\right)$$

$$\frac{\partial x_{MCH}}{\partial t} = -v\frac{\partial x_{MCH}}{\partial z} + (k_{D2}N_{tot}\theta_{MCHX^*} \cdot \frac{RT}{p_{tot}} - k_{A2}N_{tot}^X x_{MCH}(1-\theta_{H^*} - X\theta_{MCHX^*})^X)\frac{1-\varepsilon_b}{\varepsilon_b}$$

$$\frac{\partial \theta_{MCHX^*}}{\partial t} = k_{A2}x_{MCH}N_{tot}^{X-1}(1-\theta_{H^*} - X\theta_{MCHX^*})^X\frac{p_{tot}}{RT} - k_{D2}\theta_{MCHX^*}$$

(7)

The required initial and boundary conditions are:

$$x_{MCH} = x_{MCH}(0,t) = f(t)$$

$$x_{MCH}(z,0) = 0$$

$$\theta_{MCH^*}(z,0) = 0$$

$$x_{H_2} = x_{H_2}(0,t) = g(t) = constant$$

(8)

$$x_{H_2}(z,0) = constant$$

$$\theta_{H^*}(z,0) = \theta_{eq}(z)$$

The conditions above correspond to the case of imposing a concentration pulse of MCH on the hydrogen containing carrier gas when no MCH species exist initially in the reactor. The function f(t) is the time dependency of the MCH fraction at the reactor inlet. The boundary condition for hydrogen is constant because the fraction of hydrogen in the carrier gas remains constant at the reactor inlet. The initial condition for adsorbed hydrogen can be calculated by using the model for hydrogen adsorption.

The estimations were done for the X values of 2, 3 and 4. Figure 4 illustrates an experimental response pulse and the fitted model with the X value of 3. The estimated parameter values for this model were $k_{A2}=2.7*10^{-5}$

$(m^3/mol)^3 s^{-1}$ ja $k_{D2}=2.3*10^{-2}$ s^{-1}. The model was fitted only to the time behavior of the gas phase MCH. The quality of the hydrogen response signal did not allow to use the hydrogen time behavior in the object function.

The physical significance of the calculated parameters may be questionable, but the performed estimations showed that models may consist of larger nonlinear pde-systems.
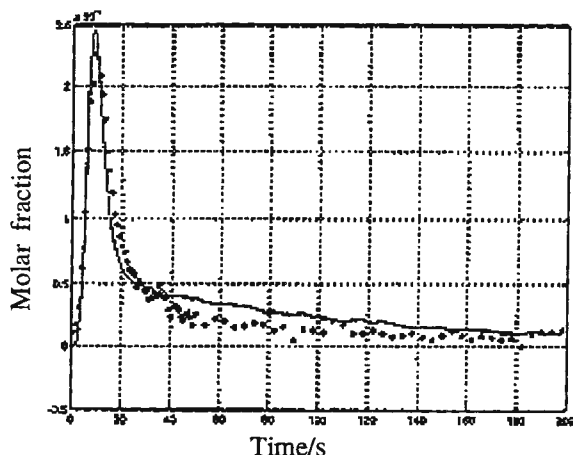

Figure 4. MCH adsorption model and experimental data.

Conclusions

The solution scheme worked in the minor transient kinetic modeling tasks. The ultimate goal to analyze the reaction mechanism of the hydrogenation of toluene is reachable provided that all the adsorption phenomena are satisfactorily modeled and the quality of the measurements places no serious obstacles. The applicability of dynamic methods in kinetic studies is mainly limited by the performance of the analyzer.

Transient kinetic analysis is suitable for kinetic modeling based on actual reaction mechanisms. Compared with traditional methods, the methods based on dynamic responses require more complicated computational analysis, but offer also better opportunities for mechanistic studies with the potential to discriminate between rival model hypotheses.

References

1. Kanervo, J., Hydrausreaktion transienttikinetiikka, Master's Thesis, Helsinki University of Technology, Department of Chemical Technology, Espoo 1998.

2. Kobayashi, H., Kobayashi, M., Transient Response Method in Heterogeneous Catalysis, Catal.Rev.-Sci.Eng., 10 (1974) s. 139-176.

3. Dekker, F.,H.,M., Transient Kinetics in Heterogeneous Catalysis, Ph.D. Thesis, University of Amsterdam, 1995, s. 1-50.

4. van der Linde, S. C., Nijhuis, T. A., Dekker, F. H. M., Kapteijn, F., Mouljin, J. A., Mathematical Treatment of Transient Kinetic Data: Combination of parameter estimation with solving the related partial differential equations, Appl. Catal. A, 151 (1997) s. 27-57.

5. Smeds, S., Salmi, T., Lindfors, L.P., Krause, O., Chemisorption and TPD Studies of Hydrogen on supported Ni Catalysts, Appl. Catal. A, 144 (1996) s. 177.

Simulation of two phase flow in a water model ladle with central gas injection

Jyrki Pitkälä[1*], Jiliang Xia[1], Jussi Vaarno[1], Ari Jokilaakso[2]

[1] Helsinki University of Technology, P O Box 6200, FIN-02015 TKK
[2] Outokumpu Engineering Contractors Oy, P O Box 862, FIN-02201 Espoo
* e-mail: Jyrki.Pitkala@hut.fi

Abstract

Simulation of two phase flow in a water model ladle with central gas injection has been carried out, using the CFX and PHOENICS codes. A two phase Eulerian model, along with the standard $k - \varepsilon$ turbulence model, is used. Predictions are evaluated against Castillejos & Brimacombe's[1] experimental data. Results show that the codes can reasonably predict the flow in the ladle, and predictions are generally in a good agreement with experimental data.

Introduction

In recent years, extensive studies have been carried out for the flow performance involved, for example, in ladle processing with submerged gas injection into molten metals. There are in profusion physical and mathematical models published in the field of ladle metallurgy[2-5]. There is also a large number of published experimental measurements made with laboratory scale water models of ladles[1,5,6] which provide a feasible basis for the verification of the mathematical models. Especially, CFD techniques have now been widely used as an implemented design tool in metallurgical industry, because CFD can provide detailed information of flow and mixing, and is relatively cheaper to be used to optimise operation systems. However, it is essential to validate numerical predictions. This may be done through the following scrutiny: (a) comparing with reliable experimental data; (b) comparing with benchmark solutions; (c) comparing predictions from different CFD codes; and (d) a combination of above methods.

There are quite a few commercial CFD codes which have been used for simulating fluid flow and mixing occurring in metallurgical processing operations. The present paper is to use the CFX and PHOENICS codes to model the two phase flow in a water model ladle with central gas injection, and compare the predictions with the experimental data.

Mathematical model

The problem considered is shown in figure 1. Air is injected via a nozzle placed on the centre of the bottom of the cylindrical vessel. Because of the symmetry, calculations are carried out in two-dimension. The air bubbles leave the domain through the top boundary which is the free surface in the water model. This free surface is assumed to be flat and stationary. Incompressible, isothermal and steady state flow is considered. The bubbles are assumed to have an average diameter of 6,0 mm. The standard drag curve[7,8] is used for the interphase friction force. In this model turbulence is considered for the liquid-phase but not for the gas-phase. The standard k-ε turbulence model[7,8] is used and an Eulerian two phase model is assumed. The pressure in both phases is assumed to be the same within the computational cell. Field equations for each phase are weighted with the volume fraction of that phase. The following transport equations are solved:

- continuity equations for gas and liquid phase

- time average liquid momentum conservation equation for two components in cylindrical co-ordinates

- time average gas momentum conservation equation for two components in cylindrical co-ordinates

Geometry and solution procedure

Data from one of the water-model ladle experiments of Castillejos & Brimacombe[11] is used in the present work. Air is blown through a circular hole with a diameter of 6.35 mm in the bottom of the cylindrical vessel which is 50 cm in diameter and the height of liquid is 40 cm. Gas flow rate is 1257 Ncm^3/min.

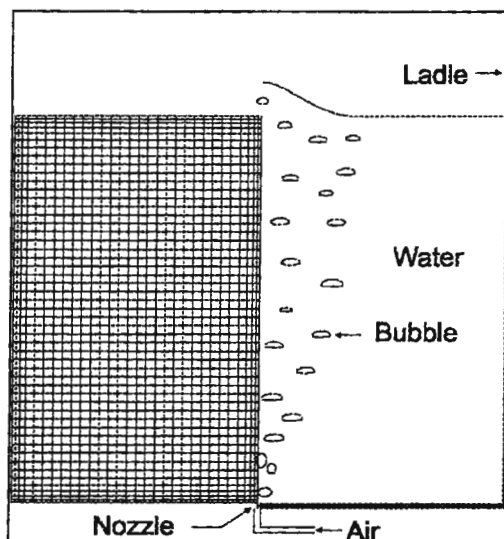*Figure 1. The computational mesh and the geometry of the studied ladle model.*

Free surface is assumed to be flat, and of a frictionless wall, through which the gas phase is allowed to flow out. In CFX, gas enters the domain as a mass source, and leaves using a sink. These have been done by modifying FORTRAN-subroutines. We set a sink term based on the vertical velocity of the bubbles at the active cells next to the degassing boundary.

The commercial CFX and PHOENICS codes have been utilized. A non-uniform 30 x 50 mesh (Figure 1) is used, which is found to produce grid-independent solutions according to trial calculations. Multi-phase calculations have difficulties in achieving good convergence. We noticed that a stable solution could be obtained by carefully changing relaxation factors or false time steps.

Results and discussion

Figures 2 and 3 show the predicted bubble rising velocities and gas volume fractions at the central axis of the vessel, respectively, and the corresponding experimental measurements are also plotted. As can be seen, CFX predicts the velocity profile amazingly well away from nozzle region whereas gas fraction profile predicted by PHOENICS is closer to measurements. It is necessary to remind that we have tried to use similar definitions in both codes. We have noticed that a more satisfactory accuracy could be achieved if a different set of model coefficients for each code is used.

Conclusion

Results suggest that the two codes can reasonably well simulate the flow in the model ladle for engineering purposes, and the predictions are generally in a good agreement with experimental data. However, differences do exist in predicting the gas volume fraction and velocity profiles in the gas-liquid plume. A more satisfactory accuracy could be achieved if a different set of model coefficients and additional physical models are used.
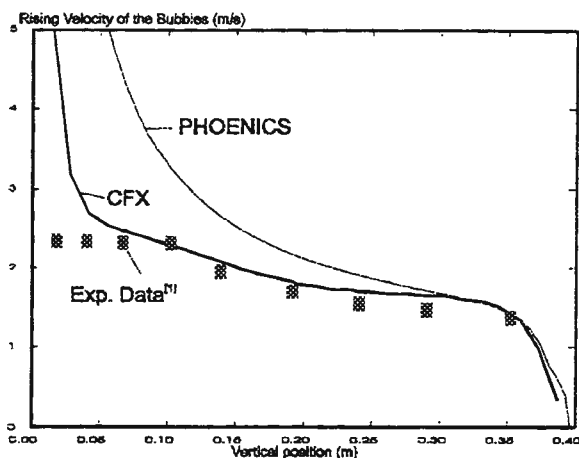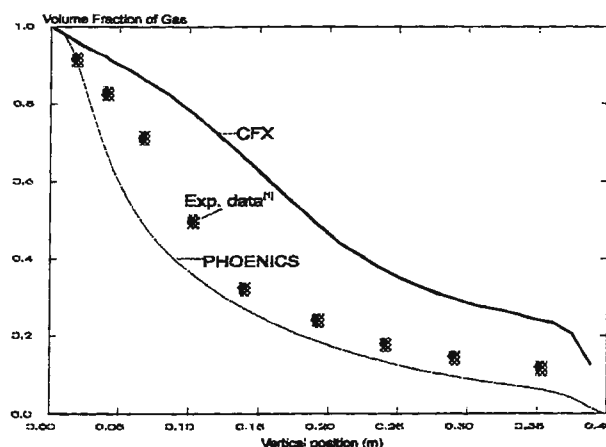
Figure 2. Bubble rising velocities

Figure 3. Gas fraction profiles

References

1   Castillejos, A. H. & Brimacombe, J. K. Measurement of Physical Characteristics of Bubbles in Gas-Liquid Plumes: Part II. Local Properties of Turbulent Air-Water Plumes in Vertically Injected Jets. Metallurgical Transactions B volume 18B, 1987 pp. 659-671.

2   Mazumdar, D. & Guthrie, R. I. L. A Comparison of Three Mathematical Modeling Procedures for Simulating Fluid Flow Phenomena in Bubble-Stirred Ladles, Metallurgical Transactions B volume 25B, 1993 pp. 308-312.

3   Schwarz, M. P. Simulation of Gas Injection Into Liquid Melts. Applied Mathematical Modelling volume 20, 1996 pp. 41-51.

4   Jönsson, P. & Jonsson, L. A Model of Gas-Stirred Ladle. In: Scaninject VII part I 7th International Conference on Refinig Processes June 7-8, 1995, Mefos, Luleå, pp. 385-409.

5   Sheng, Y. Y. & Irons, G. A. Measurements of the Internal Structure of Gas-Liquid Plumes. Metallurgical Transactions B, volume 23B 1992 pp. 308-312.

6   Türkoglu, H. & Farouk, B., Mixing Time and Liquid Circulation Rate in Steelmaking Ladle with Vertical Gas Injection. ISIJ International 1991 vol. 31, pp. 1371-1380.

7   CFDS-CFX User Guide, Version 4.1, Computational Fluid Dynamics Services, Harwell Laboratory, Oxfordshire, UK, October 1995.

8   POLIS, the PHOENICS On-Line Information System, CHAM Ltd., London, version 2.2.

Reducing Communication Overhead in Distributed Simulation Using Statistical Approach

Ivan Lališ, Peter Meňhart, Jiří Šafařík

Slovak University of Technology, 812 19 Bratislava, Slovakia

e-mail: lalis@dcs.elf.stuba.sk

http://www.dcs.elf.stuba.sk/~distsys/

Abstract

This paper introduces an idea of replacing a stream of simulation messages produced by logical processes in distributed discrete-event simulation with a reduced number of messages that statistically describe this stream. The paper presents this approach and analyses its suitability for two types of simulation models. It uses two examples to compare the results of this method with a sequential solution.

1 Introduction

One of commonly used paradigms in the field of distributed simulation is a paradigm of logical processes. It identifies physical processes (PPs) within a real discrete-event problem and interactions among them. It models PP using a logical process (LP). An interaction (exchange of information M) in a real time T between PP_i and PP_j is modelled by sending a simulation message $\langle M, T \rangle$ between LP_i and LP_j.

During a simulation run, logical processes execute local messages (events), send and receive external simulation messages. They may act asynchronously, but they must cope with the problem of causality errors. Two main groups of methods that deal with causality problems are known. Both require additional tasks to be accomplished. Typically, *conservative* methods calculate (or must be supported with) a look-ahead information and use costly mechanisms such as null messages to distribute it. *Optimistic* methods [2] use checkpoints, roll-backs and anti-messages to be able to detect a causality error and to recover from it.

Both solutions bring computational and communication overhead that may slow down logical processes and even degrade the whole computation in some cases. Although there are general approaches that try to minimise the communication overhead [3], they are usually not as successful when applied in the area of simulation. The presented solution tries to decrease the overhead, breaking one of the fundamental predicaments of the LP paradigm.

2 Principles of the Statistic Method

A simulation model consists of objects that interact with the other objects by sending and executing events. Let the model be stochastic (it contains random variables). A sequence of events scheduled by one object for the other forms a stochastic process. This process is usually not perceivable, because it takes place "inside" the model.

But when the same model is to be executed in a distributed environment, such process becomes visible. The simulation model is divided into several sub-models, each mapped to a different processor. In this case all communication activities among objects across sub-model boundaries are realised by sending time-stamped simulation messages[1]. If there were an observer monitoring a communication, he/she could perceive a stream of simulation messages (one particular realisation of a random process). The transmission of all these data along with synchronisation requirements cause the substantial efficiency problems of distributed simulation.

The idea of the method proposed in this paper is as follows. A communication between objects A and B is a stochastic process. For many models object B need not to observe the same realisation of this process as caused by the object A. It is only needed that B observes some realisation of **the same stochastic process**. A stochastic process can be described using statistical methods. The idea therefore is to

---

[1]although messages for different objects share the same channel between two processors, they can still be distinguished

1. **monitor** all efforts of the object A for inter-process communication
2. **measure** it using statistical methods
3. **transmit** a reduced amount of statistical data
4. **re-create** the data for the object B using the delivered data
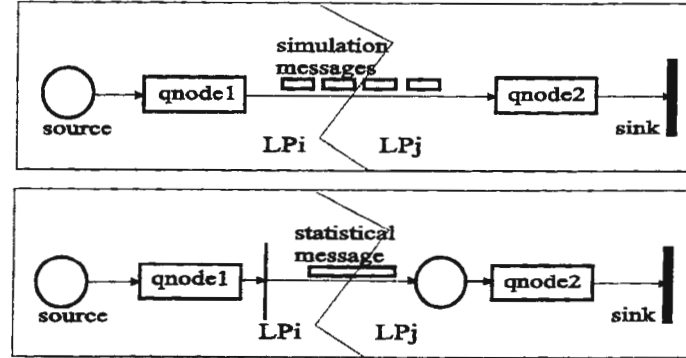

Figure 1: Basic idea of the statistic method

To transparently use this approach, one couple of special objects is inserted between each two objects communicating across sub-model boundaries (see Figure 1). A special sink object (statistical sink, *statsink*) monitors data coming from the object A and tries to capture (statistically) substantial properties. Then it sends them to a co-operating source object (statistical source, *statsource*) in a special statistical message. In turn statsource tries to "fool" the object B with the data. It can be said that a process of communication between objects A and B is modelled by a statsink-statsource couple.

3 Advantages and Disadvantages

And what is it good for? Let the simulation model represent a traffic system. If a goal is to find out how much time it takes for a particular car C to get from a spot S to a spot T then the approach fails, because it guesses the time. But if a mean value for all cars that undertook this journey is required the method can be used.

Another example comes from biophysics. In [7] a cluster of molecules is simulated using the same Markovian model for each molecule. Interactions among them are based on statistical values that each molecule contributes to. In such case the simulation messages contain statistical information and therefore may be replaced with statistical messages themselves.

The greatest advantage of the presented method is that it completely removes "traditional" data-dependencies among logical processes. All time-critical connections among LPs are replaced by time-tolerant. There is no more any possibility of a causality error. It means that all LPs can run at full speed, only in "proper moments" statistical information must be transmitted[2].

There are several limitations. As can be seen, the method tries to fake the real communication among LPs with its approximation using the statistic data obtained from the previous iteration. The method is therefore not suitable for deterministic models, where results to be observed are based on deterministic model behaviour, e.g., when a logical circuit is being simulated, an exact result is expected to check the proper function of the circuit.

Even for stochastic models, streams of messages must have appropriate properties in order to describe them statistically. E.g., when two processes exchange only a small number of messages, the statistical approach fails to describe this "stream".

"Proper moments" are the next problem. It may be clear that a transmission frequency of statistical information influences an accuracy of achieved results. It can be intuitively said that in order to achieve the same precision of statistical results of various models, a various number of messages must be exchanged. The method must cope with several situations:

---

[2]this approach is similar to synchronous approaches when a barrier synchronisation of all LPs is required. Nevertheless, in our approach the requirements on synchronisation are not so restrictive. A statistical data exchanged between two objects should not cover a future or a very deep past period of the receiver. Moreover the number of synchronisation events is much smaller and so is the communication overhead

1. parameters of all stochastic processes are known (no communication is required)

2. steady state parameters of stochastic processes are sought (probably to be used during the next simulation run to find out all parameters of the model's steady state)

3. dynamics of stochastic processes' parameters is also important

To sum it up, the method requires the following properties of the simulation model in order to be successfully applied:

stochastic model, because the method treats all the attributes of the message stream as stochastic and uses statistic methods to re-create it

appropriate number of messages per time unit to reach required precision using statistic calculations

fluent and relatively slow changes of the random process parameters, because they will be described statistically and any abrupt changes will be smoothen

4 Collection of Statistics

The stream of messages perceived by a special sink object (*statsink*) is a holy grail that a corresponding source object (*statsource*) tries to achieve. Nevertheless, statsource is unable to reckon the ideal solution because statsink does not provide enough information. All statsource can do, is to use all available information in order to approximate the stream best.

Various policies on exchange of information within "statsink-statsource" couple may be undertaken. In general no predicaments can be made on a stream of messages that is to be described. That is why a sophisticated policy must deal with changes both of stream's distribution and its parameters in time.

The simplest way to accomplish this task is to let the *statsink* object to build a histogram of message inter-departure times[3]. *Statsource* object uses histogram(s) to recreate a stream of messages. This way, statistical parameters of the stream may be preserved, provided there is a sufficient number of histogram values.

An alternative approach may to try to match the measured values with one of known distributions, executing statistical tests on the collected values.

If the measured parameters were stable, it would be sufficient to use all available data for calculations. In this case more data means a higher precision. Nonetheless, if the parameters of the distribution — or perhaps, the distribution itself, vary, the *statsink* object should "forget" old values, otherwise it would "smoothen" the dynamics of the stream. On the other hand the *statsource* object may use older histograms to predict the stream behaviour.

The histogram method tries to perceive as much information as possible — it monitors one random process realisation. It must cope with the problem of precision (statistics requires to obtain enough data) and with an evolution of model's parameters in time (the smaller the number of samples, the better dynamics of monitored parameters).

Although this method already brings a significant relaxation of commonly accepted restrictions imposed on a distributed simulation — simulation processors (LPs) know nothing about a real development of their neighbours — a process of relaxation may take several steps further.

In some situations a type of distribution of random variables (or functions of random variables) is known or can be estimated with a particular type of distribution. Therefore it should be sufficient to reckon estimations of this substituted distribution parameters (usually mean value, deviation, etc.) and not the distribution itself. Reduced calculations further decrease both computational and communication overhead.

Next sections pay attention to both histogram and known-distribution method.

4.1 Histogram Method

A histogram method is a kind of Monte-Carlo method for an estimation of a random variable density function . An error of the method can be estimated using Tschebyschev inequality:

$$P\left(|X - E(X)| \geq \lambda\sqrt{D(X)}\right) \leq \frac{1}{\lambda^2} \tag{1}$$

---

[3]the same histogram should be built for each relevant attribute of the message

where $D(X)$ is a variance of X, $E(X)$ is a mean value of X and $\lambda > 1$. Let Z be a random variable and $v_i \in R$ for $i = 0, \ldots, n$ and $v_i < v_j$ for each $i < j$. Let A_i be an event that the value of variable Z, $z \in \langle v_i, v_{i+1} \rangle$. Let $V_{i,j}$ be a random variable that reaches the value of 1 if in j-th trial A_i occurred and 0 if not. The number of occurrences m_i of event A_i in n trials is the value of the random variable M_i:

$$M_i = \sum_{j=1}^{n} V_{i,j}$$

Let the trials be independent. To estimate probability p_i of an event A_i (1) is applied (see [5] for details). It can be shown that the "real" $p_i \in \langle \hat{p}_i - \varepsilon, \hat{p}_i + \varepsilon \rangle$ with probability $1 - \delta$ and higher for n:

$$n > \frac{p_i(1 - p_i)}{\varepsilon^2 \delta}$$

The resulting n is a little bit pessimistic, e.g., using $p = 0.5$, $\varepsilon = 0.02$ and $\delta = 0.05$ results in $n > 12500$ trials. The estimation using Tschebyschev inequality can be very often further improved using central limit theorem. In this case a number of trials drops approximately ten times to more reasonable values.

A Monte-Carlo method is applied to estimate probabilities $p_0, p_1, \ldots, p_{n-1}$ for a given random variable Z and interval values $v_0, \ldots, v_n$. The probabilities along with interval values are used to generate the value from an interval $\langle v_0, v_n \rangle$ with given probabilities.

The histogram can be even utilised to restore the probability density function f (and so the type of distribution) if a method described in [1] is applied. It is based on a replacement of a single value in the sample with its kernel function. A kernel density estimate $\hat{f}$ is a combination of these kernel functions.

4.2 Known-Distribution Method

The method can be applied on queuing networks, if input streams of transactions are described by Poisson process and serving times of servers are of exponential distribution. For a network with N nodes, the system of equations must hold in a steady state:

$$\lambda_i = \gamma_i + \sum_{j=1}^{N} \lambda_j r_{ji}, \qquad i = 1, 2, \ldots, N \tag{2}$$

where γ_i is an intensity of incoming transactions from environment to the i-th node, λ_j is an overall intensity of transactions incoming to the j-th node, and r_{ij} is a transaction transition probability from the i-th to the j-th node.

In order to reach a steady state of the whole system an inequality $\lambda_i < n_i \mu_i$, $i = 1, 2, \ldots, N$ must hold. Given such system N. R. Jackson showed that each node in the network behaved as $M/M/n_i$ system with Poisson process of incoming transactions with parameter λ_i (see [5]).

The synchronisation strategy can thus take an advantage of the known distribution of the Δt intervals between two messages — exponential distribution with a parameter λ. It can easily estimate the λ parameter, for $E(X) = \frac{1}{\lambda}$.

In order to prove a suitability of the proposed simplification for queuing networks, it should be shown that the statistical solution of the problem converges to an analytical solution. In next paragraphs, such proof is given. The equations (2) can be re-written using matrices:

$$\mathbf{Ax} + \mathbf{b} = \emptyset \tag{3}$$

The solution of this system (provided that A^{-1} exists) is

$$\mathbf{x} = -\mathbf{A}^{-1}\mathbf{b} \tag{4}$$

The iterative formula for calculating $\mathbf{x}$ can be found the following way:

$$
\begin{aligned}
Ax + b &= \emptyset \\
(A + I)x - x + b &= \emptyset \\
x &= (A + I)x + b \\
x_n &= (A + I)x_{n-1} + b
\end{aligned}
$$

Let $x_0 = b$. Then the non-iterative formula can be derived for x_n:

$$\begin{aligned}
x_0 &= b \\
x_1 &= (A + 2I)b \\
x_2 &= [(A + I)(A + 2I) + I]\, b \\
&\cdots \\
x_n &= \left[(A + I)^{n-1}(A + 2I) + \sum_{i=0}^{n-2}(A + I)^i \right] b
\end{aligned}$$

Now it should be shown that

$$\lim_{n \to \infty} x_n = \lim_{n \to \infty} \left[(A + I)^{n-1}(A + 2I) + \sum_{i=0}^{n-2}(A + I)^i \right] b = -A^{-1}b$$

The proof will be given in two steps

1. $\lim_{n \to \infty} (A + I)^n = \emptyset$
   $(A + I)$ expanded with one row and column that introduces an absorbing state of the system (it collects all transactions leaving the system of N nodes) is a stochastic matrix of Markov chain with one absorbing state. It can be shown that a limit of such matrix results in a matrix with all values nulls, except of those $p_{i,absorbing} = 1$. The limit values of the former matrix are not influenced by the expansion, while $p_{absorbing,i} = 0$, for i not absorbing by definition and all multiplications "beyond" the former matrix boundaries result in zero values ($p_{i,absorbing} \cdot p_{absorbing,j} = 0$). Therefore $\lim_{n \to \infty} (A + I)^n = \emptyset$

2. $\lim_{n \to \infty} \sum_{i=0}^{n-2}(A + I)^i = -A^{-1}$
   It is clear that
   $$(I - M)(I + M + M^2 + M^3 + \ldots + M^m) = I - M^{m+1}$$

Therefore (provided $(I - M)^{-1}$ exists)

$$\sum_{i=0}^{m} M^i = (I - M)^{-1}(I - M^{m+1})$$

Using Step 1 of this proof, $\lim_{n \to \infty} \sum_{i=0}^{n-2}(A + I)^i$ can be calculated as

$$\lim_{n \to \infty} \sum_{i=0}^{n-2}(A + I)^i = (I - (A + I))^{-1}(I - \emptyset) = -A^{-1}$$

5   Alternative Sources of Statistics

Throughout this paper "statsink-statsource" couples are being mentioned. Nonetheless, the only connection between them is a data sent from sink to source in a format known to both sides. That is why the statsink part of the couple can be replaced with other object that sends the data in the same format:

simplified model — if only coarse results are required, some parts of the model (especially those not involved in model cycles) can be replaced with *statistics driven communicators* (replacement of statsink object) fed by simulation results of these parts

hybrid simulation — if the simulation runs in a real time ($\Delta simulation\ time \doteq \Delta real\ time$) even a real system may be included into the simulation system, provided measurement devices deliver required data, e.g., a radar measures the velocity of cars and provides mean value, variance, median, etc. Such a combination may be also used for a prediction based on real-world results [6]

6   Implementation Issues

The proposed method was implemented within the YeAST simulation toolset as a new synchronisation strategy for LPs (for details on YeAST see [4]).

The modifications of the model required for statistical simulation (see Figure 1) are transparent for the designer of the simulation experiments. He/she creates a hierarchical structure of the model, assigns

```
statsim(SimProc ANY DATA(END ntimes)
    CONNECT(src:0 > qnode1:0
            qnode1:0 > qnode2:0
            qnode2:0 > snk:0)
    source(src SP1 DATA(EXP 1))
    qnode(qnode1 SP1 DATA(NOLIMIT NEGEXP 0.6))
    qnode(qnode2 SP2 DATA(NOLIMIT NEGEXP 0.6))
    sink(snk SP2)
)
```

Figure 2: Structure of the model in Figure 1

the parts of it to existing processors, and declares that the statistical method is to be used. It is also possible to set the number of statistic messages to be sent within each *sink-source* couple (see Figure 2). During the translation phase, the presence of a connection between SP1 and SP2 (simulation processor) is detected and the model description is automatically modified (see Figure 3). A couple of *statsink* and *statsource* objects is added with names that reflect the name and the port number of the object that causes the connection, e.g., qnode1_0_snk stands for a statistic sink for the qnode1 object, connected to port #0, qnode1_0_snk_src is a counterpart to qnode1_0_snk object.

```
statsim(SimProc ANY DATA(END ntimes)
        CONNECT(src:0 > qnode1:0
            qnode1:0 > qnode1_0_snk:0
            qnode1_0_snk_src:0 > qnode2:0
            qnode2:0 > snk:0)
    source(src SP1 DATA(EXP 1))
    qnode(qnode1 SP1 DATA(NOLIMIT NEGEXP 0.6))
    statsink(qnode1_0_snk SP1)
    statsource(qnode1_0_snk_src SP2)
    qnode(qnode2 SP2 DATA(NOLIMIT NEGEXP 0.6))
    sink(snk SP2)
)
```

Figure 3: Structure of the modified model

During the simulation run, each sink of *sink-source* couple is prompted *ntimes* times to send the statistical information to its counterpart. The simulation run is blocked until all messages are re-transmitted. The non-blocking solution is also possible, but due to different computational load may lead to incorrect results.

7 Examples and Results

In order to justify the described method, two examples were solved, each using the proposed method and a sequential approach. The accuracy of results as well as the speed-up were checked.

In both cases $N = 16$ and a "network" of four T800 transputers was used. Both examples were mapped onto a circle topology. One, two and four transputers were used, assigning to each of them a segment of 16, 8 and 4 queuing nodes.

7.1 Example #1: Pipe of N Queuing Nodes

The first example scales the problem in Figure 1 to N queuing nodes and n simulation processors[4]. The results of speedup and precision of a value of utilisation of the N-th server ($100.\frac{|\rho_{N,seq} - \rho_{N,statist}|}{\rho_{N,seq}}$)are depicted in Figure 4.

As can be seen, the speedup is almost linear with a quite good value of 3.5 for four processors. Precision of results increases with increasing number of messages. 100% difference for two messages and

---

[4]this simple example is often used to check a speedup of distributed simulation, because it contains a high level of parallelism
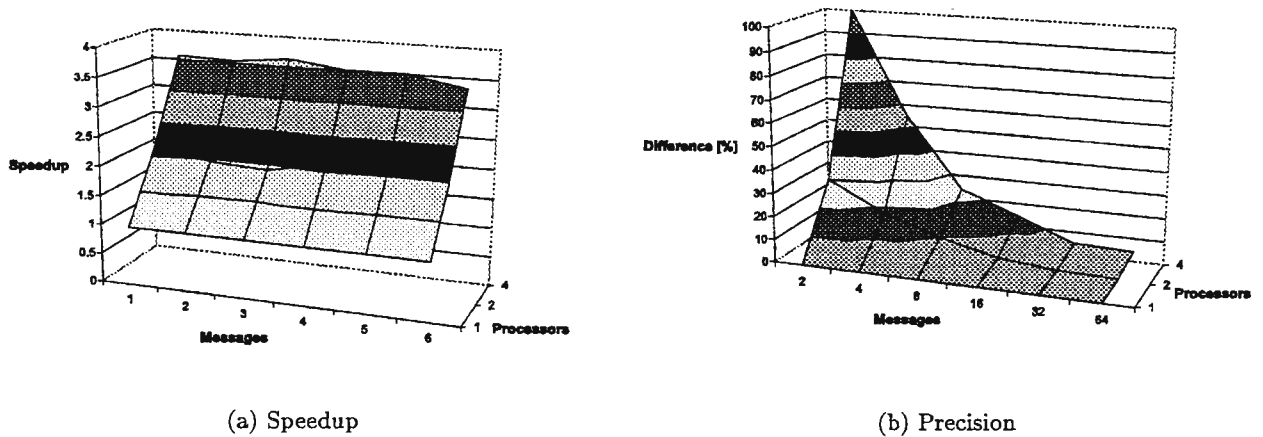
(a) Speedup   (b) Precision

Figure 4: Pipeline of 16 Nodes

four processors implies utilisation $\rho_{N,statist} = 0$, i.e., N-th server processed no transactions at all. An exchange of at least $n - 1$ statistical messages during a simulation run is needed to make the n-th processor work.

7.2   Example #2: Queuing Network with a Cycle

The network is the same pipeline of N queuing nodes, but in this case, the transactions may return with a probability p to the first node or leave the system with a probability $1 - p$.

Due to a presence of the cycle in the model, a *conservative* approach cannot be used without a lookahead information. This cycle may also increase a probability of a rollback, when an *optimistic* approach is chosen. Nonetheless, a statistical approach can transparently deal with this problem, only an appro-

(a) Speedup   (b) Precision
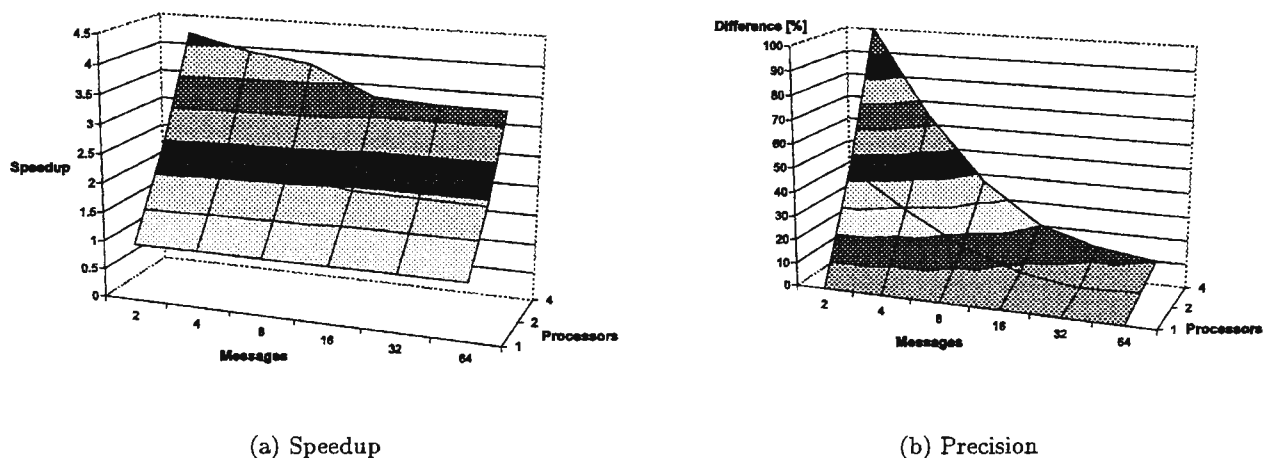
Figure 5: Cycle of 16 Nodes

priate interval between two statistical messages should be chosen, in order to achieve a desired accuracy of results (see Figure 5). Another way to obtain better results is to re-simulate the model using message stream parameters as observed in a steady state (see section 3).

8   Conclusions and Future Work

The proposed method was approved to be suitable for obtaining statistical results, although no claim on the precision of the results was made. It would be therefore interesting to try to estimate a required

number of messages for a given number of processors and precision. The precision of the method may be also improved by reducing the first, tuning phase, when processors are not working with proper data via a seed of pre-calculated values.

Acknowledgment

References

[1] L. Budínská, J. Štefan, and N. D. Tran. Comment on kernel estimates probability density. In *Advanced Simulation of Systems ASS'96*, pages 95–101, Zabreh, September 1996.

[2] P. Hanáček. Problems of implementation of virtual time for parallel simulation. In *Advanced Simulation of Systems*, pages 298–303, Ostrava, 1996.

[3] L. Hluchý, M. Dobrucký, and J. Astaloš. Hybrid approach to task allocation in distributed systems. In *Lecture Notes in Computer Science 1277*, pages 210–216. Springer, 1997.

[4] I. Lališ and P. Meňhart. Object oriented toolset for sequential and distributed simulation. In *European Simulation Multiconference ESM'95*, pages 604–608, Prague, June 1995. SCS.

[5] Š. Neuschl, J. Blatný, J. Šafařík, and J. Zendulka. *Modelovanie a simulácia*. Alfa Bratislava, 1988.

[6] E. T. Peytchev and A. Bargiela. Paralel simulation of city traffic flows using Padsim (Probabilistic ADaptive SImulation model). In *European Simulation Multiconference ESM'95*, pages 330–334, Prague, June 1995. SCS.

[7] A. Zahradníková and I. Zahradník. A minimal gating model for the cardiac calcium release channel. *Biophysical Journal*, 71:2996–3012, December 1996.

THE ATM VIRTUAL CHANNELS SIMULATION FOR A PACKET NETWORK SIMULATOR

Julije Ozegovic[1], Mario Mornar[2]

1. Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split,
HR-21000 SPLIT, R. Boskovica bb, Croatia, julije.ozegovic@fesb.hr

2. Croatian Post and Telecommunications, Telecommunications Center Split,
HR-21000 SPLIT, Sinjska 4, Croatia, mario.mornar@efst.hr

ABSTRACT: The step by step introduction of ATM technology will in the next decade result in heterogeneous networks of Ethernet LANs interconnected through the ATM WAN segments. The protocol and flow control architectures of heterogeneous networks are analyzed. End to end flow control with traffic shaping at the edge of the ATM network is assumed. In this work, the simulation of heterogeneous networks using the event driven simulation systems is based on a simple ATM SVC model which consist of traffic shaper and a conventional communication channel. The experimental implementation of the model is described, and the preliminary simulation results are presented.

KEY WORDS: event driven simulation, ATM, Internet, end to end flow control

1. INTRODUCTION

The broadband integrated services networks will probably be completed through the realization of the global public ATM network, which is to be used for the Ethernet LAN interconnections [2]. In the heterogeneous networks of such structure the conditions for the data traffic growth are to be achieved, where the flow control algorithms take the main role in the process of integration.

Experimental ATM and heterogeneous networks have been built in recent years to investigate the flow control dynamics and predict their operational characteristics. However, simulation of such networks is an unavoidable step in the performance analysis, which provides the possibility to control the every aspect of the experiment. Financial effects of the simulation experiments are significant as well.

Among several classes of the ATM and packet networks simulators, the event driven simulator systems are common because of their suitability to incorporate the sophisticated flow control algorithms [1]. More than that, those algorithms are implemented using the same code as in the real implementations, contributing to the experimental results reliability and to the quality of the software solutions tested.

Depending of the experiment's topics, event driven network simulators are build to simulate packet or ATM networks. Merging the two is not a trivial issue, because resulting simulation structure is complex, dependent on a large number of simulation parameters, and results are interpreted with significant difficulties.

Our approach to the problem is to implement the ATM network channels as an extension to the packet network simulator, based on the simple model of data traffic shaping on the edge of the ATM part of the heterogeneous packet network.

2. ATM AND HETEROGENEOUS NETWORKS FLOW CONTROL ARCHITECTURE

The ATM (Asynchronous Transfer Mode) technology has been developed in last ten years to provide the media for various traffic integration (voice, video, multimedia, real-time, data). Its perspective is based on the compatibility with the SDH/SONET optical transmission system. However, ATM technology lacks the application layer and will depend on other specifications. It is likely that the Internet application layer, and consequently transport and network layer specifications, will be temporarily adopted due to the existing Internet customers population.

Step by step realization of the global public ATM network will postpone expected changes of the customers installations. Taking into account the existing base of Ethernet (and other LAN) installations, it can be foreseen that the next decade will be dominated by the Ethernet LANs interconnected through ATM networks, possibly evolving to the global public ATM network. Such networks we call "heterogeneous". The protocol architecture of the heterogeneous network, based on the Internet paradigm, is shown on the Figure 1.
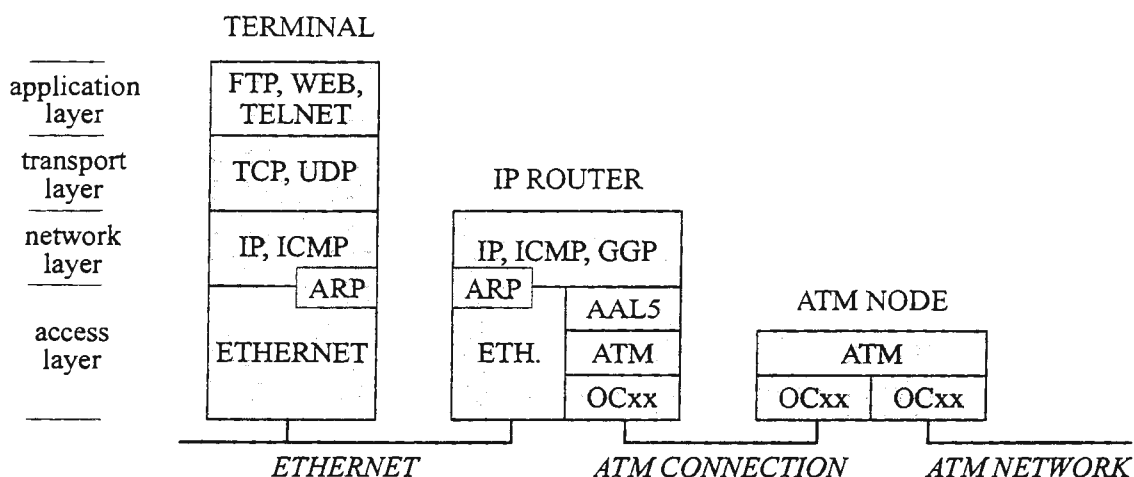
Figure 1 - The protocol architecture of the heterogeneous network

In such a network, using existing network (IP) and transport (TCP) protocols to maintain compatibility with the non-ATM parts of the global Internet, it is not possible to use the ATM flow control mechanisms to tune the sending rate of the terminal transmitter. The same difficulties occur in the simple case with a terminal using the Internet protocols directly connected to the ATM network. At the moment, only IP to ATM address transformation is solved, while the optimal ATM SVC (Switched Virtual Channel) parameter signalization is far from the final specification. However, using SVCs makes it possible for IP network to use the optimal routing of the underlying ATM network.

The ATM traffic management specification [5] provides four classes of service, the CBR (Constant Bit Rate), real and non real time VBR (Variable Bit Rate), ABR (Available Bit Rate) and UBR (Unspecified Bit Rate). CBR and VBR use forward flow control, appropriate for high capacity traffic flows when significant network delay prohibits the lost packet retransmission. Users negotiate the SVC parameters with the network, and later shape their traffic in accordance to these parameters. The network controls the user traffic conformance, marking excessive cells lower priority or dropping them altogether.

The ABR and UBR classes are provided for data communication users, interfacing through the AAL5 layer. ABR guarantees some QoS (Quality of Service) providing backward (feedback) flow control. The available rate is signaled to the AAL5 process, where traffic shaping takes place. Conformance tests are performed at the network side of the interface. UBR class of service provides no flow control. Low priority cells are buffered at the ATM nodes, while the excessive delays and possible cell losses can be used as the implicit congestion indication to the transport layer flow control algorithms.

The only possible flow control architecture at the current stage of protocol development is the end to end (transport layer) flow control, Figure 2. This approach is also the only appropriate one for the UBR class of service.
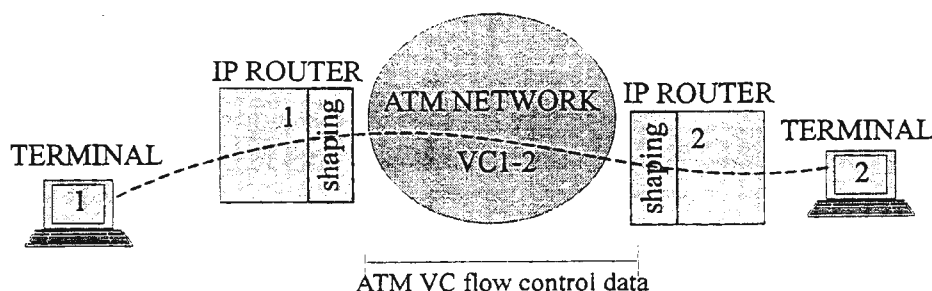

Figure 2 - The flow control architecture of the heterogeneous network

The interest of the user is to minimize cell losses, i.e. to shape his traffic to the SVC parameters negotiated. This function is performed at the IP-ATM interface (AAL), often far away from the terminal transmitter process. Other simulation studies have shown that the consequence of the shaping process is the cell buffering at the edge of the ATM network [4]. The main role of traffic shaping, to protect other ATM users from misbehaving ones, is achieved.

3. THE ATM VIRTUAL CHANNEL MODEL

The existence of the traffic shaping process at the edge of the ATM network makes it possible to simulate ATM SVCs in a simple way. Instead of simulating a whole, complex ATM network, one can simulate the dynamics of the traffic shaper on the edge of the ATM network. It can be assumed that the shaped traffic will be served with negotiated QoS, meaning with constant delay and low cell loss probability. Another advantage of this approach is that a complex model of ATM network with numerous simulation parameters (of the background traffic) is substituted with a simple model which needs managing only few simulation parameters. The ATM SVC is simulated using a traffic shaper and a conventional, constant delay duplex channel T_{pi}, C_i, Figure 3.
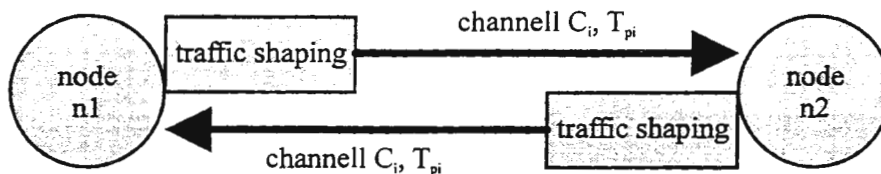
Figure 3 - The ATM SVC simulation model

The ATM traffic management specification defines the conformance testing procedure based on the leaky bucket model. To satisfy conformance tests, users can shape their traffic using the same leaky bucket algorithm. This approach is appropriate for the event driven simulation systems.

The leaky bucket shaper for the packet network consists of a cell buffer, where packets are segmented to cells, and a PCR (Peak Cell Rate) sender synchronized with the credit availability. Credits are generated with the appropriate process at SCR (Sustained Cell Rate) and accumulated at the credits buffer. The credit buffer depth is tuned to the MBS (Maximum Burst Size) value.

The CBR class of service emulates switched channel with constant cell rate and nearly constant delay appropriate for voice communications. The CBR channel simulation is achieved setting SCR=PCR and MBS=1, i.e. peak and sustained cell rates are the same and maximum burst size is one cell (no cell bursts), Figure 4.
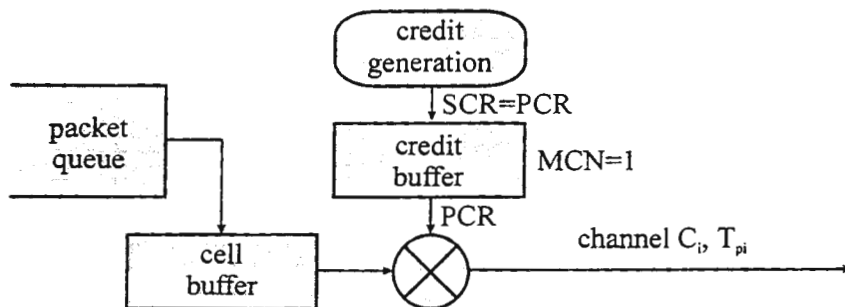

Figure 4 - The CBR simulation model

The VBR class of service is devoted to users which send their cells at variable rate (compressed voice and video, multimedia), which makes it possible to statistically multiplex their data flows. The network utilization is higher than with CBR service, and consequently more economical to the user. The VBR channel simulation is characterized with separate PCR and SCR settings together with the MBS greater than 1, meaning that cell bursts are allowable, Figure 5.
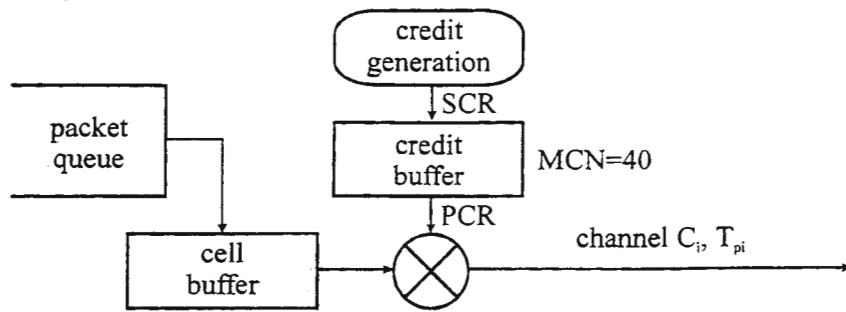
Figure 5 - The VBR simulation model

The residual network capacity is offered to data communications users after the CAC (Customer Admission Control) process has made necessary resource reservations for CBR and VBR channels. The ABR class of service is used for data communications with closed flow control feedback loop, to inform users about the average residual capacity available. The traffic management specification specifies explicit congestion notification mechanism and explicit rate signalization mechanism. Our model is based on the congestion indication mechanism. The sending process sends special forward RM (Resource Management) cells every 32nd data cell or at least every 100 ms. Forward RM cells collect the congestion information in the CI (Congestion Indication) and NI (No Increase) bits. The RM cell is sent back at the destination (backward cell). After reception of the backward RM cell, sender process either decreases its ACR (Actual Cell Rate, CI=1), increases ACR (CI=0, NI=0), or simply keeps its ACR constant (CI=0, NI=1). The ACR increase and decrease rates are set at the channel establishment phase (default to 1/16).

The actual ABR channel simulation is achieved performing the modulation of the credit generator with a random process, bounded to (0.1 - 1)PCR. The random process changes its state every 100 ms. The effect of random process is synchronized to the credit generator with RM cell mechanism according to the traffic management specification, Figure 6.
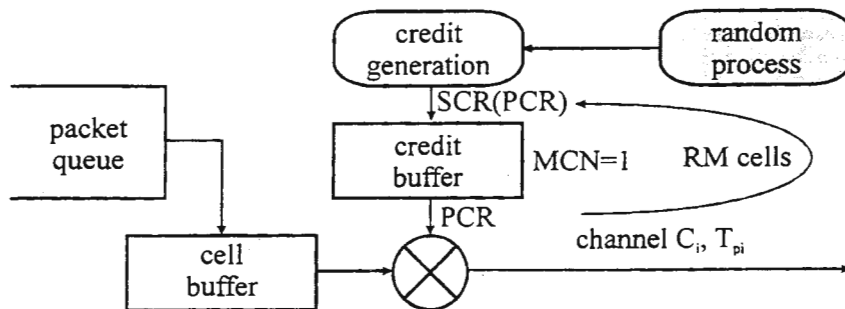
Figure 6 - The ABR simulation model

The UBR channels are used for data communications without any QoS guarantees. No flow control nor conformance testing are provided. The cells are set to low priority level and dropped if excess node buffer capacity is occupied. The cell buffering has the consequence of prolonged cell delays. In our UBR simulation model the credit generation is modulated with random process on every cell to induce cell delays. The cell loss process with loss probability proportional to the cell delay is included, Figure 7.
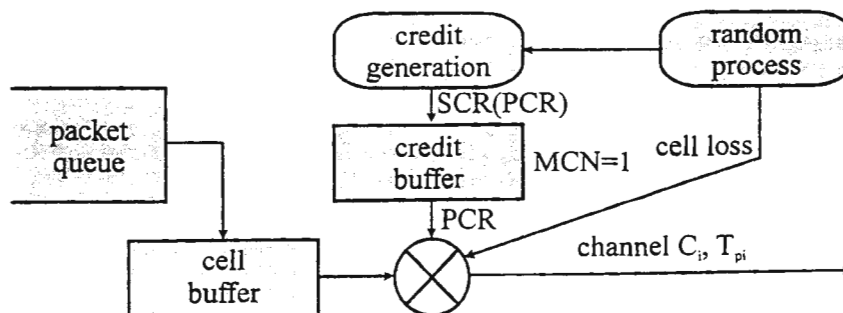

Figure 7 - The UBR simulation model

4. EXPERIMENTAL IMPLEMENTATION

The ATM channel simulation models are experimentally embedded in the Lawrence Berkeley Laboratory Network Simulator NS-1.1, an event driven packet network simulator designed for the research of the end to end flow control algorithms [6]. The object structure of the simulator is presented on the Figure 8.
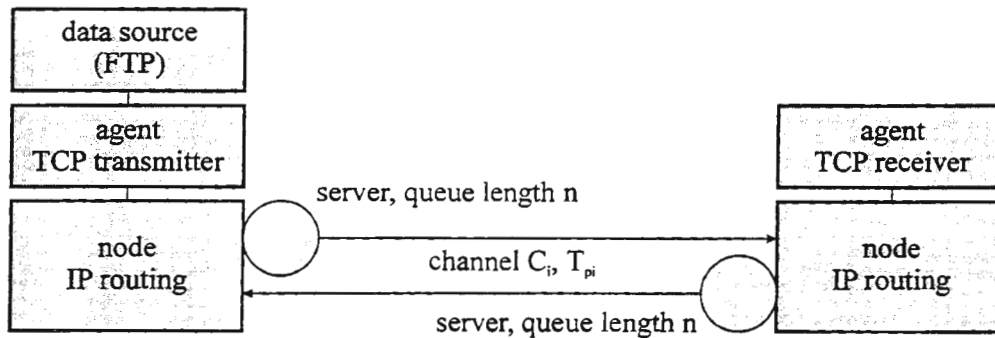

Figure 8 - The LBL NS-1.1 simulator object structure

Central process of the simulator is the event scheduler working in the simulation time. Simulation topologies are built from node objects, connected through link objects (communication channels). Link objects perform server algorithms (FIFO, RED, class based queuing). TCP agents, which include flow control algorithms, reside above the appropriate nodes, and logical connections between transmitters and receivers are established. IP routing is accomplished automatically. Several data sources are available to simulate user behavior. Receiver agents just drop received "data".

ATM channels are embedded in the link object of the simulator, spreading its functionality with the four classes of ATM service. The simplified formal specification of the simulation algorithm using the pseudo language is shown in Figure 9.

```
object link
function handle (event from scheduler)
    if (event=send_interrupt)
        send(), exit;
    if (event=credit_interrupt)
        new_credit(), exit;
    if (event=random_credit_interrupt)
        random_credit_update(), exit;
    if (event=RM_minimum_imterrupt)
        send_RM_cell(), exit;
    // default (event=RM_receive_interrupt)
    update_ABR_rate(), exit;
function send()
    if (atm=NO)
        conventional_channel_send(), exit;
    if (credits=0)
        exit;
    if (cell_count=0)
        if (new_packet=0)
            exit;
        cell_count (new_packet);
    if (atm=UBR)
        random_time=f (random_function, PCR);
    credits--;
    schedule send_interrupt (PCR)
```

```
    if (atm=UBR)
        cell_lost(random_function, PCR);
    if (atm=ABR)
        RM_cell_count--;
        if (RM_cell_count=0)
            send_RM_cell();
    if (--cell_count=0) // last cell of packet
        if (atm=UBR & cell_lost=1)
            drop_packet();
        else
            ATM_packet_send();
function new_credit()
    credits++(max_credit_number);
    if (atm=ABR or atm=UBR)
        schedule credit_interrupt (random_credit);
    else (atm=CBR or ATM=VBR)
        schedule credit_interrupt (SCR);
function random_credit_update()
    random_credit=f (random_function, PCR);
    schedule random_credit_interrupt (100 ms);
function send_RM_cell()
    RM_cell_count=32;
    schedule RM_receive_interrupt (2*delay);
    cancel RM_minimum_interrupt;
    schedule RM_minimum_interrupt (100 ms);
```

Figure 9 - The formal specification of the ATM channel simulation

The experimental results of the TCP-Reno flow control algorithm behavior on a simple one-channel network are presented on the Figure 10.
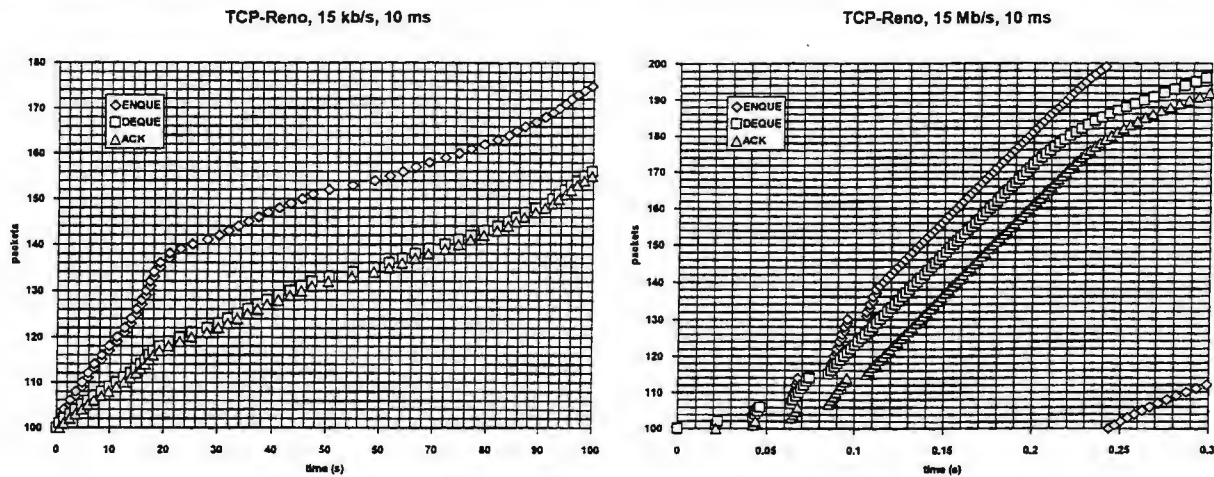

Figure 10 - The TCP-Reno behavior over the ABR channel

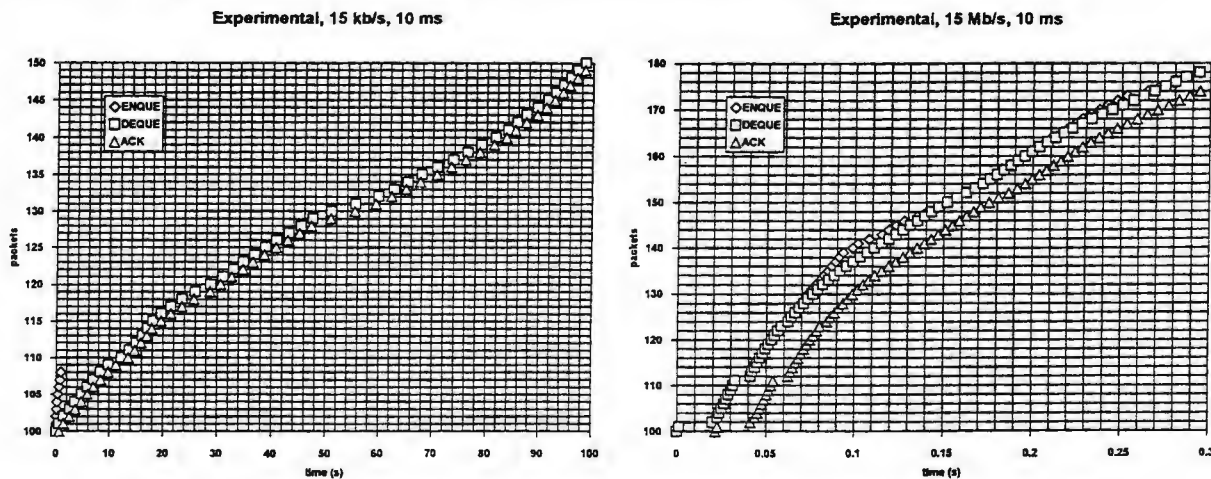The behavior of an experimental flow control algorithm [3] is presented on the Figure 11.

Figure 11 - The experimental flow control algorithm behavior over the ABR channel

It is evident that the experimental flow control algorithm estimates the available channel capacity.

5. CONCLUSIONS

The packet network simulator is extended with ATM SVCs based on the simple model consisting of the traffic shaper and the conventional channel. All relevant flow control algorithms of the ABR class of service are simulated. The experimental results show that the simulation model used is appropriate for at least unrefined simulations. The model makes it possible to compare flow control algorithms behavior, such as TCP-Reno and experimental ones used to obtain simulation results presented.

REFERENCES:
[1] Frost, V.S.: "MODELING AND SIMULATION IN NETWORK MANAGEMENT", McGraw-Hill, 1993.
[2] Ozegovic, J.: "FLOW CONTROL ARCHITECTURE IN HETEROGENEOUS PACKET NETWORKS", Proceedings SOFTCOM'97, Split 1997.
[3] Ozegovic, J.: "OPTIMAL FLOW CONTROL ALGORITHMS IN HETEROGENEOUS NETWORKS", Ph.D. Thesis, University of Split, Split 1997.
[4] Patel, B. V, Bisdikian C. C.: "End-Station Performance under Leaky Bucket Traffic Shaping", IEEE Network Magazine vol. 10, no. 5, pp. 40-47, Sep. 1996.
[5] xxx "ATM TRAFFIC MANAGEMENT SPECIFICATION VERSION 4.0", ATM-Forum 1996.
[6] xxx "NS MANUAL PAGES", LBL Berkeley 1993.

A COMPARISON OF THE MODELS FOR THE PATH LOSS PREDICTION IN SUBURBAN ENVIRONMENT

Igor Zanchi 1, Zoran Blažević 2, Ivan Marinović 3

1, 2, 3 University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Split, Croatia
Ruđera Boškovića bb, 21000 Split, Croatia, fax 385 21 563877
1 izanchi@fesb.hr 2 zblaz@fesb.hr 3 ld@fesb.hr

Abstract -The goal of the paper is to compare chosen models for path loss prediction and, at the same time, apply them on a practical example of suburban area mobile communications. Furthermore, we'd like to see in what extent these methods satisfy us in a case where all of the criteria required by examined models aren't satisfied by environment, and also where exact area geometry data is unknown. For that purpose, theoretical and measured data are compared.

1. INTRODUCTION

When dealing with design of any communication system, the main need is to know the path loss between transmitter and receiver. In the case of radio networks, since mathematical modeling of the propagation of radio waves is very complicated, especially in urban or suburban environments, several empirical, semiempirical and even physical models have been developed to suit the needs of system designers. The propagation of radio waves in urban or suburban environments is strongly influenced by the size, type and density of building, as well as natural objects. So, various experimenters use number of characteristics and methods to classify land usage types and environment [1].

Before we enter in description of compared prediction models it is worth to note neither of the method is applicable in all situations and for all environments. The accuracy of particular model depends on the fit between parameters required by the model and those available for given environment as well as theirs accuracy. In general, we are examining a small area (e.g. street) and trying to predict the mean signal strength for it. Of course, because there is no model applicable for all environments, it is recommended to compare some data obtained by selected model with experimental data or to be very careful when choosing a method. Short descriptions of models examined and compared in this article and their basis are as follows.

2. DESCRIPTIONS OF MODELS

As mentioned above, there are three approaches in designing of the path loss models: empirical based on experimental or measured data as well known Hata model based on measurements made by Okumura or Lee's model, semiempirical as COST 231 Walfish – Ikegami model or physical as Ikegami – Yoshida – Takeuchi – Umehira model [1], [2], [3], [4].

Physical models are mainly based on the geometrical optics assuming a simple two-ray model. Why is that so may be viewed from Fig. 1. At street level buildings shadow the fields emanating from an elevated fixed antenna. Majority of the propagation paths cannot lie through the buildings and cannot be associated with propagation between the buildings or with multiple diffracted components [3]. Thus, we can conclude that field reaching street level results from diffraction and reflection fields from buildings in the vicinity of the mobile. If we take a look at mobile situation at ideal city structure, as depicted in Fig. 2, and assume plane waves, than available power $P(x)$ received by an isotropic antenna is given by [4]:

$$P(x) = \frac{1}{2}\sum_{i=1}^{N}E_i^2 + \sum_{i=1}^{N}\sum_{k=i+1}^{N}E_iE_k \cos\left[2\pi\frac{\cos\phi_i - \cos\phi_k}{\lambda}x + (\theta_i - \theta_k)\right], \tag{1}$$

where E is field strength in V/m, θ is phase and λ is wavelength. A mean power received by mobile moving over a distance l is as follows:
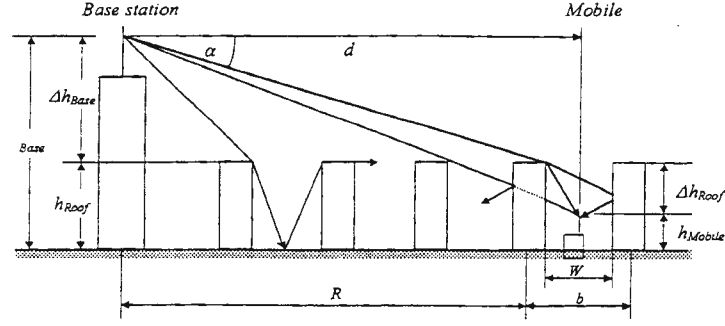
Figure 1. Various ray paths for UHF propagation in presence of buildings.

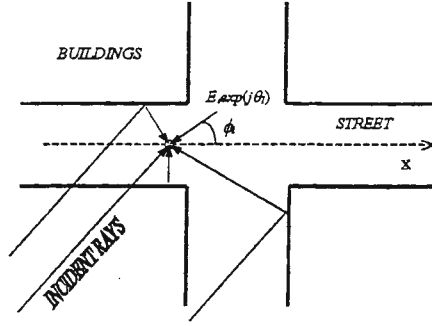**Figure 2.** A multipath propagation model

$$\overline{P}(X) = \frac{1}{l}\int_0^l P(x)dx = \frac{1}{2}\frac{1}{l}\int_0^l \sum_{i=1}^N E_i^2 dx +$$
$$+\frac{1}{l}\int_0^l \sum_{i=1}^N \sum_{k=i+1}^N E_i E_k \cos\left[2\pi \frac{\cos\phi_i - \cos\phi_k}{\lambda}x + (\theta_i - \theta_k)\right]dx \tag{2}$$

The second term of equation (2), which is oscillatory due to interference of each two multipath waves, tends to disappear when l is large enough compared with the periods of fluctuation. Thus, if that condition is fulfilled, only first term remains:

$$\overline{P}(X) = \frac{1}{2}\sum_{i=1}^N \overline{E_i^2} = \sum_{i=1}^N \overline{P}_i , \tag{3}$$

where $\overline{P}_i$ is a mean power of ith multipath wave over examined distance l. So, mean received signal power is given by the power sum of the arriving waves. If we denote power sums of accounted and disregarded waves by $\sum P_a$ and $\sum P_d$, respectively, then the error is given by:

$$\Delta = 10\log\frac{\sum P_a + \sum P_d}{\sum P_a} = 10\log(1+\delta) \text{ [dB]}, \tag{4}$$

where $\delta = \sum P_d/\sum P_a$. For example, if disregarded power is less than accounted the error is within 3 dB.

2.1 IKEGAMI - YOSHIDA - TAKEUCHI - UMEHIRA MODEL

Field strengths of the two principal rays E_1 and E_2, e.g. diffracted and reflected component respectively, are given by the following approximate equations with some assumptions [4]:

$$E_1 \approx 0.225 \Big/ \sqrt{2}\, E_0 \sqrt{\lambda w} \Big/ (h_{Roof} - h_{Mobile})\sqrt{\sin\phi} \tag{5}$$

413

$$E_1 \approx 0.225 \Big/ \sqrt{2} E_0 \sqrt{\lambda(2W - w)} \Big/ L_r (h_{Roof} - h_{Mobile})\sqrt{\sin \phi} , \qquad (6)$$

where E_0 is free space field strength, λ is wavelength, w is distance between mobile and building, and L_r is reflection loss defined by an amplitude ratio of incident and reflected waves. Reflection loss takes values of the order of 4 – 10 dB on average according to the experiments of VHF and UHF bands [4]. Also, it can be shown mean field strength is almost constant across a street [4] thus may approximately be presented by that at the center of the street. So, path loss in dB may be calculated as in formulae:

$$L = L_0 - (5.8 + 10\log(1 + \tfrac{3}{L_r})^2 + 10\log W - 20\log(h_{Roof} - h_{Mobile}) +$$

$$- 10\log(\sin \phi) - 10\log f) \qquad (7)$$

where L_0 is free space (FS) loss calculated from well known formulae:

$$L_0[dB] = 32,4[dB] + 20\log f + 20\log d \qquad (f \text{ in MHz}, d \text{ in km}) \qquad (8)$$

2.2 COST 231 WALFISH – IKEGAMI MODEL

COST 231 Walfish –Ikegami model is applicable to large, small and micro cells. It is composed of three terms:

$$L_b = \begin{vmatrix} L_0 + L_{rts} + L_{msd} \\ L_0, \quad L_{rts} + L_{msd} \le 0 \end{vmatrix} \qquad (9)$$

where L_{rts} is rooftop-to-street diffraction and scatter loss, L_{msd} is multiscreen loss, and L_0 is FS loss. The rooftop-to-street diffraction loss is:

$$L_{rts} = \begin{vmatrix} -16,9 - 10\log W + 10\log f + 20\log \Delta h_{Mobile} + L_{ori}, \quad h_{Roof} > h_{Mobile} \\ 0, \qquad L_{rts} < 0 \end{vmatrix} \qquad (10)$$

where

$$L_{ori} = \begin{vmatrix} -10 + 0,345\phi, & 0 \le \phi \le 35^0 \\ 2,5 + 0,075(\phi - 35^0), & 35^0 \le \phi \le 55^0 \\ 4,0 - 0,114(\phi - 55^0), & 55^0 \le \phi \le 90^0 \end{vmatrix} \qquad (11)$$

$$\Delta h_{Mobile} = h_{Roof} - h_{Mobile}, \qquad (12)$$

$$\Delta h_{Base} = h_{Base} - h_{Roof} \qquad . \qquad (13)$$

The multiscreen diffraction loss is:

$$L_{msd} = \begin{vmatrix} L_{bsh} + k_a + k_d \log d + k_f \log f - 9\log b \\ 0, \qquad L_{msd} < 0 \end{vmatrix} \qquad . \qquad (14)$$

The term k_a represents the increase of the path loss for base station antennas below the rooftops of adjacent buildings, the terms k_d and k_f control the dependence of the multiscreen diffraction loss versus distance and radio frequency, respectively, and L_{bsh} is a path loss component which is a function of the Δh_{Base} [2].

2.3 LEE'S MODEL

Lee's model is an example of empirical model and is intended to use at 900 MHz. It's applicable in two modes: an area-to-area mode and a point to point mode. In the area-to-area mode, the prediction is based on three parameters: median transmission loss at range of 1 km (L_0), slope of the path-loss curve (γ) [1], and adjustment factor (F_0). Thus, the path loss formulae has a form:

$$L_{50} = L_0 + \gamma \log d + F_0 \tag{15}$$

As mentioned above, the values of L_0 and γ were derived from experimental data and are given as defaults [1]. An adjustment factor is given by:

$$F_0 = F_1 F_2 F_3 F_4 F_5 \tag{16}$$

where $F_1 = (^{ACTUAL\ BASE\ STATION\ ANTENNA\ HEIGHT\ (m)}/30.5)^2$, $F_2 = {}^{ACTUAL\ TRANSMITTER\ POWER\ (W)}/10$,
$F_3 = {}^{ACTUAL\ GAIN\ OF\ BASE\ STATION\ ANTENNA\ (dB)}/4$, $F_4 = (^{ACTUAL\ MOBILE\ ANTENNA\ HEIGHT\ (m)}/3)^2$, for $h_R \neq 3\ m$
$F_5 = (f_c/f)^n$, $n = 2 - 3$, $f_c = 900\ MHz$

To predict propagation loss in the point-to-point mode, one may use following formulae:

$$L = L_{50} + 20\log(H_e/30) \tag{17}$$

where H_e is effective base station height.

3. COMPARISONS WITH FIELD EXPERIMENT

In order to try the effectiveness of described model and as we have taken in an account that whole realm of wireless communications doesn't always lies in big flat urban centers, the experiment was executed in small town of Supetar (island Brač, Croatia) where terrain isn't so flat and building heights aren't so uniformly distributed as depicted in Fig. 1. The parameters of the antennas are given in table 1. In order to compare theoretical and experimental results one representative route is chosen at range about a half kilometer from base station and is 160 m long. Frequency is 907,8 MHz. For simple map data input a simple MS Qbasic program is used and for plotting and calculation the Matlab scripts. Because of the complexity of the computer simulation, several assumptions have been made as follows.

1. There is no street canyon propagation in a given example.
2. If mobile antenna is visible from the base station, propagation loss is only a subject of FS loss.
3. There is always a reflected wave component.
4. In one profile, the distribution of building heights is uniform, with first building height adjacent to a mobile greater than mobile antenna height as reference.
5. In our example, some of the predefined values [2] have been exploited because of lack of exact building data.

The comparisons of experimental and theoretical data are given in figure 3.

4. CONCLUSIONS

As we can see from graphs depicted in Fig. 3, our ad hoc example gave us different results. Although one example may not be enough to extract a precise conclusion about usefulness of shown methods it is sufficient to compare them for our practical example. One of the important parameter we are going to employ is Root Mean Square (RMS) error. It can be calculated as:

$$RMS_{error} = \sqrt{\frac{(Y_p - Y_m)^2}{N}} \tag{18}$$

where Y_p denotes predicted values, Y_m measured values while N stands for number of values considered. RMS errors for given examples are shown in table 3. So, where are those differences arises from? By careful examination of given models and results one may notice following facts.

1. Lee's method does not relay on actual building geometry. It relays solely on experimental data for given environments. For a given example, it provides bad tracking of measured data but the smallest RMS error.
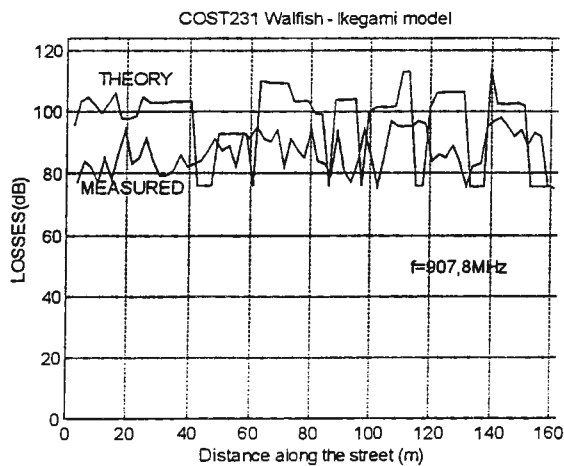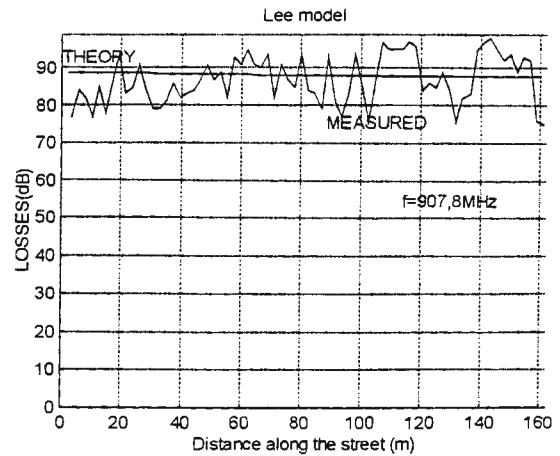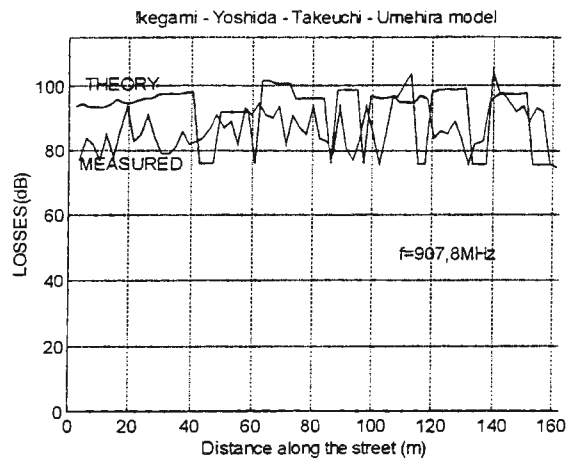
Fig. 3a Ikegami-Yoshida-Takeuchi-Umehira model results

Fig. 3b Lee's model results

Fig. 3c COST 231 Walfish-Ikegami model results

Table1. The parameters of used antennas

Antenna	Gain (dBi)	Power (W)	Pattern	Height (m)
Base	8	10	OMNI	16
Mobile	2,1	-	OMNI	1,8

Table 2. RMS errors for given examples

Method from figure	3a	3b	3c
RMS error (dB)	12,43	6,62	16,96

2. Physical based Ikegami – Yoshida –Takeutchi - Umehira model provides the best tracking, even not the least RMS error.
3. COST 231 method is a model with good resolution, which takes in account many factors, so lack of some precise data may cause additional error. In our example, this method gave us overestimated losses, which arises from insufficiently correct building geometry data.

Despite all imperfections of described methods, they represent a power tool for system designers and engineers. But, when deciding about a method, one must know general fact about a number of methods, characteristic of examined area and limitations of the considered method in order to exploit one among them that the best suits the need of a given task.

REFERENCES

[1] V. K. Garg, J. E. Wilkes: "Wireless and Personal Communications System", Prentice Hall PTR, USA, 1996.
[2] J. Doble: "Introduction to Radio Propagation for Fixed and Mobile Communications", Artech House, Boston – London 1996.
[3] J. Walfisch, H. L. Bertoni: "A Theoretical Model of UHF Propagation in Urban Environments", IEEE Trans. on APP, Vol. 36, No. 12, Dec. 1988.
[4] F. Ikegami, S. Yoshida, T. Takeuchi, M. Umehira: " Propagation Factors Controlling Mean Field Strength on Urban Streets", IEEE Transactions on Antennas and Propagation, Vol. AP-32, No. 8, August 1984.

ASSESSMENT OF METHOD SELECTION FOR CALCULATION OF MULTIPLE DIFFRACTION LOSSES BY APPLICATION OF "KNIFE EDGE" MODEL

Igor Zanchi 1, Ivan Marinović 2, Zoran Blažević 3

1, 2, 3 University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Split, Croatia
Ruđera Boškovića bb, 21000 Split, Croatia, fax. 385 21 563877
1 izanchi@fesb.hr 2 1d@fesb.hr 3 zblaz@fesb.hr

Abstract - Multiple diffraction losses according to Knife Edge model are defined by multiple Fresnel integral which, except in the case of two obstacles, may be solved by numerical method acc. to Vogler, or by widely accepted approximate methods. Taking into consideration diffraction losses calculated by numerical integration (Vogler), as reference ones, it is evident that approximate Daygout's model gives the best results. However, with certain configuration of diffraction obstacles on radio path, result according to Deygout may be considerably improved by introduction of Giovaneli's correction. The intention of this work is to give an improvement assessment of results obtained by application of Giovaneli's correction as the function of geometric disposition of diffraction obstacles.

1. INTRODUCTION

The simplest assessment model of diffraction losses in radio communication is based upon application of Frensel diffraction theory. The obstacle is evaluated only by its height (hill or object), and the model is known as "Knife Edge" (KE). In the cases of the presence of multiple obstacles, approximate methods are commonly used. The best results among approximate methods are obtained by Deygout's method with Giovaneli's correction. In order to define variables, which determine the amounts of corrections that may be obtained by Giovaneli's model, we are going to employ Figure 1. There are two emphasized obstacles between transmitter and receiver denoted only by theirs height h_1 and h_2 in respect to transmitter-receiver junction line.

Let us assume for a beginning there is only one obstacle with height h_1 and another is removed. Diffraction losses according to KE model are determined by:

$$a(dB) = -20\log\left(\left|\frac{\mathbf{E}}{\mathbf{E}_0}\right|\right) = a(d_1, d - d_1, h_1) = a(v_1)$$

$$v_1 = h_1\sqrt{\frac{2}{\lambda}\frac{d}{d_1(d - d_1)}}$$

where E stands for electric field at the receiving point in presence of a diffraction obstacle, $\mathbf{E}_0$ field at the same point in free space (FS) environment, and their ratio is determined by Frensel's integral:

$$\frac{\mathbf{E}}{\mathbf{E}_0} = \frac{1 + j}{2}\int_v^\infty e^{-j\frac{\pi}{2}v^2}dv$$

In the presence of multiple diffraction obstacles on the radio path, diffraction losses are determined by multiple Frensel's integral, which can be solved analytically only in the case of two obstacles. Thus in practice, there are two ways to deal with this problem. According to one of them, it is possible to reach a

solution using Vogler's [1] numerical procedure. Another way used in practice employs several approximate methods [4], in which the contribution to overall losses of particular diffraction obstacle is evaluated on different ways.

In order to evaluate particular approximate method, obtained diffraction loss values were compared with ones obtained by numerical calculations according to Vogler taking them as reference. Namely, according to many examples of comparison between calculated diffraction losses according to KE model, Vogler's method provided the best similarity between calculated and measured values.

The model for prediction of multiple diffraction losses according to Deygout gives the results the closest to Vogler's. Simplicity of this model is making it suitable for practical use. However, in some cases, diffraction path loss results according to Deygout are overestimated. This deficiency is alleviated or almost totally avoided by implementation of Giovaneli's correction [3].

2. GIOVANELI'S MODEL CONTRIBUTION ANALISYS

Total diffraction loss caused by presence of both obstacles (Fig. 1) according to Deygout's model may be obtained as a sum of particular obstacle contribution, according to following interpretation:

$$a_D = a(v_1) + a(v_2')$$ (1)

where $$v_1 = h_1 \sqrt{\frac{2}{\lambda} \frac{d}{d_1(d - d_1)}}$$ (2)

and $$v_2' = h_2' \sqrt{\frac{2}{\lambda} \frac{d - d_1}{d_0(d - d_2)}}$$ (3)

According to this, h_1 is height of dominant obstacle on the radio path, and obstacle with higher parameter p_i is dominant:

$$p_i = \frac{h_i}{r_i}$$

where h stands for ith obstacle height, and r for radius of the first Frensel's ellipsoid on the same obstacle spot. According to Giovaneli, diffraction losses on same radio path are determined by:

$$a_G = a(v_1') + a(v_2')$$ (4)

with $$v_1' = h_1' \sqrt{\frac{2}{\lambda} \frac{d}{d_1(d - d_1)}}$$ (5)

It is evidently that a_G and a_D shall be equal to each other with $h_2 \leq 0$, because Giovaneli's model is Deygout's when h(R') ≤ h(R), according to marks on Figure 1.
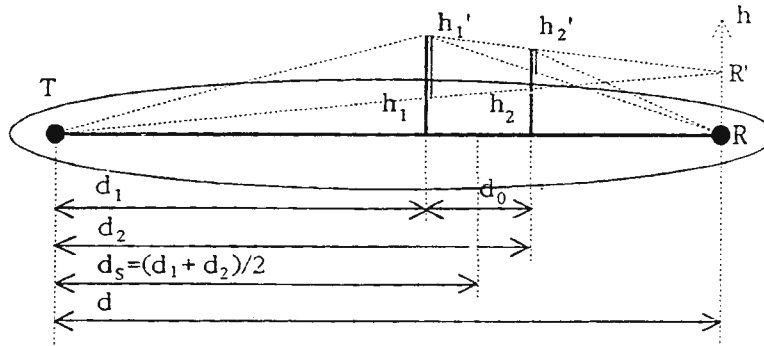

Figure 1. *Radio-path with two obstacles*

Losses a_G are less than a_D for value $a(v_1') - a(v_1)$, which enables an implementation of the correction:

$$Corr\ (\%) = \left| \frac{a_G - a_D}{a_D} \right| \cdot 100 \qquad (6)$$

In order to express value $Corr$ in dependence of relative dimensions, new values are defined:
$w_1 = \dfrac{d_0}{d}$, $w_2 = \dfrac{d_s}{d}$, $w_3 = \dfrac{h_2}{h_1}$, as well as parameter $q = \dfrac{h_1}{r_M}$, where $r_M = \sqrt{\lambda d}\,/\,2$ stands for maximum radius of first Frensel ellipsoid, and h_1 denotes dominant obstacle. Values v_1', v_2' and q may be represented as:

$$v_1' = v_1 \left[1 - \left(w_2 - \frac{w_1}{2} \right) \frac{w_1 - (1 - w_3)\left(1 - w_2 + \frac{w_1}{2} \right)}{w_1} \right] \qquad (7)$$

$$v_2' = v_1 \left[w_3 \left(1 - w_2 + \frac{w_1}{2} \right) - \left(1 - w_2 - \frac{w_1}{2} \right) \right] \sqrt{\frac{w_2 - \frac{w_1}{2}}{w_1 \left(1 - w_2 - \frac{w_1}{2} \right)}} \qquad (8)$$

$$q = v_1 \sqrt{2 \left(w_2 - \frac{w_1}{2} \right) \left(1 - w_2 + \frac{w_1}{2} \right)} \qquad (9)$$

Applying a new notation, parameters p_i are defined by ratio:

$$\frac{p_1}{p_2} = \frac{1}{w_3} \sqrt{\frac{\left(w_2 + \frac{1}{2} w_1 \right)\left(1 - w_2 - \frac{1}{2} w_1 \right)}{\left(w_2 - \frac{1}{2} w_1 \right)\left(1 - w_2 + \frac{1}{2} w_1 \right)}} \qquad (10)$$

When $p_1/p_2 > 1$, the first obstacle is dominant.
If the second obstacle is dominant, employing following mapping:

$$w_2 \Rightarrow 1 - w_2$$
$$w_3 \Rightarrow \frac{1}{w_3} \qquad (11)$$

problem is driven again to a problem with dominant first obstacle. Also, it is interesting to analyze again when $Corr > 0$ (Fig. 2),
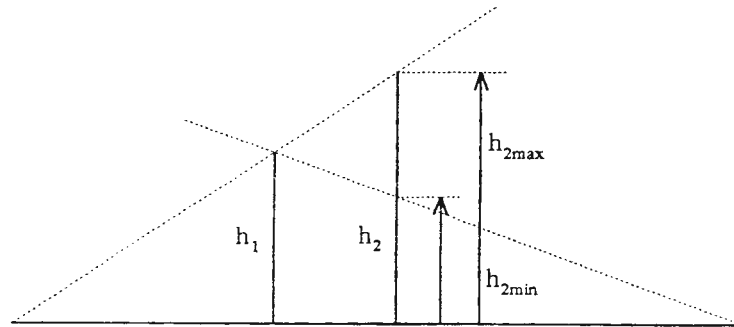

Figure 2. Illustration of geometry for Corr>0

Corr ≠ 0 if $h_{2min} < h_2 < h_{2max}$, that is:

$$\frac{1 - w_2 - \frac{1}{2}w_1}{1 - w_2 + \frac{1}{2}w_1} < w_3 < \frac{w_2 + \frac{1}{2}w_1}{w_2 - \frac{1}{2}w_1} \tag{12}$$

Analyzing value *Corr* in relation to w_1, w_2 and w_3 one may conclude about contribution of Giovaneli's correction. Note that at every graph (Fig. 1, 2, 3) one of these values is independent, while other two are constants. Furthermore, for every quantity of a parameter *q* different quantity of *Corr* is obtained, thus it can be viewed as a parameter value. However, graphs were made in such way that for every quantity of an independent variable, value of a parameter *q* that results in maximum correction *Corr* is determined, by which the maximum error is defined also, so described analysis may be viewed as the worst case calculation.
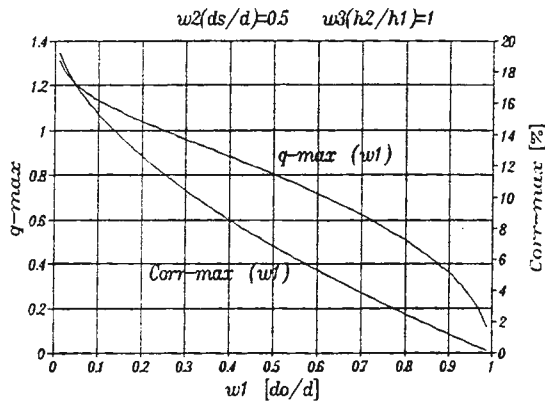
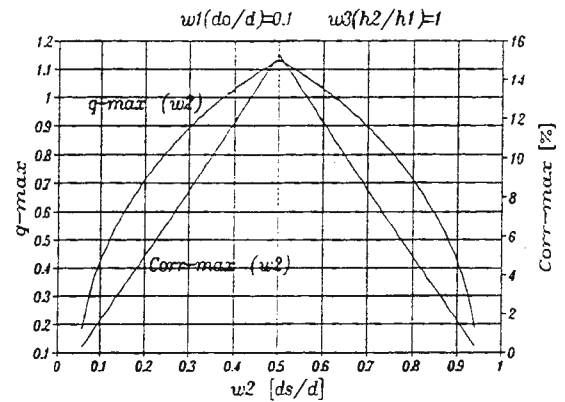**Fig. 3.** *Display of* qmax *and* Corrmax *relating to* w_1

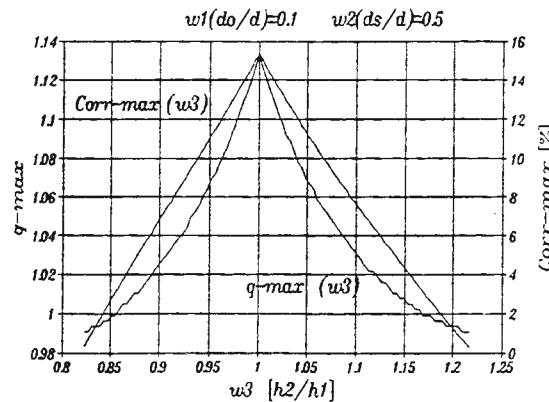**Fig. 4.** *Display of* qmax *and* Corrmax *relating to* w_2

Figure 5. *Display of* qmax *and* Corrmax *relating to* w_3

Figure 3 indicates dependence of *Corr* to w_1. Giovaneli's contribution to accuracy is higher when w_1 is smaller, that is when obstacles are closer to each other. Figure 4 shows dependence of *Corr* to w_2. The highest correction is when obstacles are placed at the middle of the radio path. Figure 5 shows that the highest correction is when obstacles are same height. Thus, for that case, three-dimensional graph showing how *Corr* is dependent to w_1 and w_2, with $w_3 = 1$ as parameter has been made (Fig 6). It can be seen that the highest possible correction is 20.5%, which is obtained when two obstacles with same height are meeting on the half way of the radio path. This result is approaching to quantity of losses obtained by one obstacle with same height at the middle of the radio path.
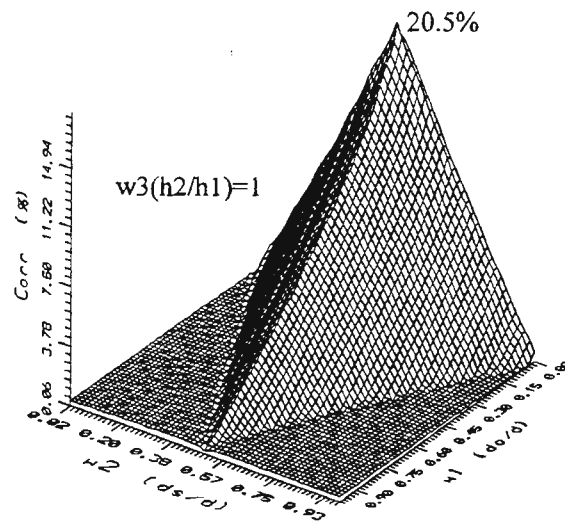
Figure 6. *3D display of Corr relating to w_1 and w_2*

All curves greatly depend of remaining two parameters, so it's hard to employ obtained numerical values. However, on the basis of curve trend, some conclusions can be made, so it has been done in next chapter.

3. CONCLUSION

According to presented analysis, it may be concluded that Giovaneli's procedure in particular configuration greatly contributes increasing accuracy of estimating diffraction losses according to Deygout. Specially, it's worth with very close obstacles settled at the middle of the radio path and with same heights. Also, it's very important to implement this procedure if q is close to q_{max} (by calculating with height of dominant obstacle). Similar thinking worth even when multiple obstacles are present. Correction is going to be higher with q closer to quantity of q_{max} for dominant obstacle, with dominant obstacle closer to the middle of the radio path, with remained obstacles relatively closer to dominant and with more similar heights. In those cases, it's advisable to employ Giovaneli's procedure.

REFERENCES

[1] L. E. Wogler: "An Attenuation Function for Multiple Knife-Edge Diffraction", Radio Science, Vol. 17, No. 6 pp. 1541-1546., Nov.-Dec. 1982.

[2] J. Deygout: "Multiple Knife-Edge Diffraction of Microwaves", IEEE Transactions on Antennas and Propagation, Vol. AP-14, No.4, pp. 480-489., July 1996.

[3] C. L. Giovaneli: "An Analysis of Simplified Solutions for Multiple Knife-Edge Diffraction", IEEE Transactions on Antennas and Propagation, Vol. AP-32, Mar.1984., p.297.

[4] CCIR Report 715-2 (1978.-1982.-1986.): "Propagation by Diffraction", (Study Programme 1A/5)

SimLab: Enterprise simulation in virtual reality

Riitta J. Smeds[1]

[1]Helsinki University of Technology, P.O. Box 9500, FIN-02015 HUT, riitta.smeds@hut.fi

Simulations for increased enterprise innovation and learning

Enterprise innovations originate from management practice, not from management theory. New management principles and organizational forms are typically first experimented and implemented in forerunner companies, and receive their theoretical explanation ex post through empirical research. [1, 2, 3, 4, 5]

In today's turbulent competitive environment, enterprises have to continuously develop their structures and strategies. Real life experiments to create and adopt more competitive business processes and strategies are conducted in companies all the time. New management principles like Just-in-time, Lean Production, Activity Based Costing, Value Management, Agile Manufacturing, or organizational forms like Process Organization, Team Organization, Network Organization, Virtual Organization etc. have been created in innovative companies at an accelerating speed, and thereafter eagerly benchmarked by the followers - as well as explained and theorized by researchers.

However, trial and error is a slow and ineffective way of innovation and learning, and errors in real life can be fatal. For product innovations, scale models and even virtual prototypes are used for testing out different product concepts before expensive realization. Analogically, for "enterprise innovations", *simulation models* are needed that allow "enterprise prototyping", the experimentation of an enterprise's organization, its operational processes, strategies, management principles and potential performance, before final design and implementation. *Simulations on enterprise models* are needed to connect practice and theory, doing and thinking, into an efficient spiral of innovations and learning in companies. [5, 6, 7]

Classification of enterprise simulation models

Enterprise simulation models can be classified according to the *nature of the system to be simulated*, and the *tools* used the simulation (Figure 1). For example material flows and layouts of complex production systems, as well as economic systems, are often modeled as *technical subsystems* of the enterprises. They can be simulated with computer supported mathematical models. Production scheduling simulation packages for companies, as well as software business games for education are examples of technical enterprise simulation models. Simulations on the technical models are run on the computer, and the outputs can thereafter be used as input in the human decision processes or for training.

But enterprises do not only consist of technical systems, rather they are complex socio-technical systems. Thus the models of enterprise processes should also incorporate the social dimension, the *interaction of human beings*. This can be achieved "manually" in brainstorming and games, but also applying *virtual reality to support the social simulation of groups* (Figure 1). In this *hybrid simulation*, the group can experience, experiment, discuss and develop a joint understanding of the dynamic functioning of enterprise systems, with the help of virtual reality models.

The third dimension of the classification comes from the fact, that the simulation models can be either generic or tailored. *Generic simulation models* simulate the behavior of fictive enterprises. They can be used for educational purposes, to create awareness and understanding about the general dynamics of enterprises, about new management principles and concepts. A *tailored simulation model* supports the experimentation and innovation of a specific system of a certain company.

Enterprise simulation games

In the above classification of simulation models, enterprise simulation games are *social simulations*. The most important elements in the simulation models are the *players of the game*, since the dynamic operation of the process is simulated through their *action and communication*.

	Technical systems	Social systems
Manual	Concretization Conceptualization Analytical models Scale models	Brainstorming Role games Board games Social simulation games
Computer supported	Graphical models Mathematical models Dynamic models	*Hybrid:* *Pedagogical animations,* *Virtual reality* *& Social simulation*

Figure 1. ·Classification of enterprise simulation models [8]

An enterprise simulation game is based on a simplified and accelerated model of a selected *enterprise process*. This model is often physically constructed e.g. on a game board or built into a simulation room. The game model and rules should be in relevant aspects as isomorphic with the real process as possible, because they direct and restrict the understanding created by the simulation. The simulation games are often facilitated by one or several *facilitators*.

Generic enterprise games are often based on a *manual model*. For example a production system game can be built of a game board, physical material, written rules and documents. The game is played according to the rules by human players occupying fictive roles [e.g. 9, 10]. Some of these generic games are available as commercialized products, e.g. ADVANTIG [11], and the Lego Truck Game [12]. - To a growing extent, the economic calculations and the technical subsystem simulations during the games are conducted on *computer-based mathematical models* [e.g. 13, 14].

Computer-based *business games* are generic games, frequently used in university education. The simulation is restricted to the economic performance and competition of fictive companies, based on decisions of the players that are fed into mathematical models. Thus the business games are basically technical simulations. Through the inclusion of *team work and seminars* into the decision making and reporting, the social dimension can however to some extent be incorporated into the game. [15]

Tailored simulation games are based on company specific models, and on company data. Typically, a process flow chart of the activities, information and material flows is first created. Thereafter this rough model is *manually constructed*, e.g. by furnishing a seminar room as a "scale model" of the process, with adequate paraphernalia and process materials. In the game itself, the players fill in the relevant details of the process by explaining their operations and the coordination and cooperation rules. Thus the composition of the game team is especially important for the validity of the tailored simulation model. The team should mirror the organization of the process as closely as possible; ideally all employees and managers engaged in the real process should participate also in the simulation.

The players act in their own work roles. They *act and "talk through"* the activities and information and material flows of the process, following the *route of a concrete case example*, e.g. a specific customer order, and using supporting case material. Observers, e.g. support staff from the company's development functions, representatives from other departments or organizations can participate in the game and in the discussions, depending on the focus of the game. If top managers participate in the game, also the strategic and policy aspects of process operations can be included in the simulation [e.g. 5].

Through the human interaction and discussion in the game, the players' tacit knowledge about the process is externalized and shared, and a common understanding is created. Development ideas, problems and open questions concerning the process are awakened and written down for further use in smaller process development teams. The teams refine the ideas into a new process design to be tested in a prototyping manner in the subsequent simulation games. [4, 16, 17, 18, 19, 20, 21, 22].

Tailored simulation games are efficient methods for process experimentation and innovation [e.g. 4, 5, 6]. Therefore the choice of the target process and the objectives for the process "prototyping" should be based on strategic analyses. Sometimes, the first simulation game becomes itself an important method to clarify and define the strategic objectives for the process re-engineering project [5, 6]. Although the models of tailored games are enterprise specific, the methods and procedures for building the models and for running the games can to some extent be standardized [23, 17]. There exist also tailored games, where the technical subsystem simulations are run on a *computer model*. This enables an accelerated simulation e.g. for capacity planning in the production process [24].

Virtual reality to enhance the effectiveness of enterprise simulation games

The *social simulation* in both generic and tailored enterprise games is achieved through human *interaction, discussion and experimentation* during the game. This human interaction is supported by manual or computerized tools, which both have their limitations. In the *manual* games, the alternatives that can be jointly experimented are restricted by the limitations of the physical place and material, and also time, since the game runs have to be performed manually. Using conventional *computerized tools* in some parts of the simulation can cause limitations, since the dynamics of the game is restricted by the mathematical model and hidden into computer program, and thus cannot be "seen" and acted upon by the players.

Virtual reality gives new technological possibilities to overcome these limitations. Through the use of the latest achievements in multimedia, computer graphics and visual programming, a *visual interactive model* can be created for the group simulations and experimentations of both generic and tailored enterprise systems. This is the development idea of the *Enterprise Simulation Laboratory SimLab* [8].

The Enterprise Simulation Laboratory aims to support the interactive human experimentation through the creative combination of existing *enterprise simulation games, dynamic mathematical models, and virtual reality*. The virtual reality for simulations will be created through the *virtual wall* concept [25]. On one wall of the laboratory, a huge three-dimensional picture of the simulated system is projected from a computer display. This picture can be flexibly modified applying computer graphics and e.g. video picture. Every player can simultaneously see the wall, which enables a shared simulation experience.

The virtual wall gives almost unlimited possibilities to *visually model the enterprise "space"*, from the simple furnishing of the simulated room (e.g. applying animation) to the visual presentation of data, and even to the telepresence of the players in the virtual space (immersive virtual reality):

1) Conceptual simulation data can be projected to the wall, for every player to see: e.g. the layouts and process charts, performance measures, economic analyses of the processes; also outputs from mathematical simulations of the technical subsystems, if applied during the simulation game.
2) The physical simulation environment can be visualized in a two- or three-dimensional form: e.g. the future layout of the new company, the view of the workshop with its machines, the material flows and even the movements of virtual actors in the workshop. - The game facilitator can change these images flexibly and quickly on the screen, even during the simulation.
3) The players can interact directly with the virtual wall model, e.g. change the layout of the factory and the location of the machines in the workshop.
4) The players can immerse themselves into the virtual reality, e.g. into the planned future factory, and experience the space and the functioning of the factory from "within".
5) Even players at different geographical locations can participate in the same simulation, if the virtual wall picture is transmitted via interactive broadband telecommunication networks simultaneously to several locations. The players can interact applying CSCW technologies and video, and even meet as virtual actors in the "enterprise space". The Enterprise Simulation Laboratory is thus not necessarily tied to one location, but can exist in a telematic network.

The two first applications can be realized with existing multimedia software, hardware, and simulation game methods, but the latter three require multidisciplinary research and development effort to create software and equipment for an appropriate user interface for the virtual "enterprise space".

The SimLab development project

The Enterprise Simulation Laboratory SimLab is being developed in a three year project 1997-2000 at Helsinki University of Technology, TAI Research Centre, as a multidisciplinary effort of three laboratories: Industrial Economics, Industrial Psychology, and Telecommunication Software and Multimedia (TLM).

The project proceeds in a "concurrent engineering mode" with three cutting edge Finnish industrial companies representing machine industry, telecommunications and medical industry:

1) Pilot company A needs a generic, modular simulation game to train its whole personnel in understanding the operations and strategies of a typical industrial company, and the connections between operational processes and economic performance.

2) Pilot company B is growing rapidly, and is moving to a new factory. It needs tailor-made technical and social simulation support to re-engineer its layout, production lines, material flows, and later its global logistic chains. The company is also in the process of implementing a new operations control system, and has to train its employees to the new processes.

3) Pilot company C is re-organizing its highly complex, knowledge intensive R&D process of medical drugs, and needs interactive social simulation games to be able to dramatically decrease the lead time of new product development.

The project proceeds in a prototyping manner. For each pilot company, simulation model prototypes are developed and tested with company test groups. The evaluations of the prototypes' *usability*, *learning effects*, effects on process *innovations*, as well as the *technical requirements* concerning the software and graphics are important feedback for the development of the subsequent versions of the models, and of the whole Enterprise Simulation Laboratory concept.

Special emphasis is put from the start of the development project on the validity of the models and the learning effects of the simulations. The experiences that are created in the virtual enterprise space have to equal in relevant aspects the real enterprise processes, otherwise the players will learn the wrong lesson. Another crucial aspect is the usability of the virtual reality applications: the players' experiences and the group interaction during the simulations should not be hampered by complex user interfaces.

SimLab in future use

After the development project, SimLab will continue it functioning as a multidisciplinary research and teaching laboratory at Helsinki University of Technology, and as an enterprise clinic. SimLab and its generic enterprise simulation games can be used for new, computer supported experimental learning in many Industrial Management courses. The combination of the virtual wall and advanced telematics creates new possibilities for students to engage in distant collaboration (CSCW) and learning in the "virtual university".

Practitioners from industry can be invited for educational generic simulations to SimLab. But secondly, and more importantly, they can use SimLab as an "enterprise clinic" for their specific problem solving and process development tasks. Development groups can attend simulations, which are tailored to the enterprises' needs. Even global enterprise processes can be experimented through the virtual meeting of players in the same "enterprise space".

Multidisciplinary research teams of SimLab will have excellent possibilities to study virtual reality and multimedia simulations of enterprise processes, as well as their user interfaces and learning effects. As "patients" of the SimLab clinic, the enterprises will provide rich empirical data for research on organizational decision making, innovation and experimental learning in a laboratory setting.

References

1 Chandler, A.D., JR. (1962). *Strategy and structure: Chapters in the history of the American industrial enterprise*, Cambridge, Mass.: The M.I.T. Press.

2 Helling, J. (1991) *World Champions*. Sellin & Parner Förlag AB, Stockholm (in Swedish).

3 Womack, J. and Jones, D. (1994) From Lean Production to Lean Enterprise, *Harvard Business Review*, March-April, 93-103

4 Smeds, R. (1994) Managing Change towards Lean Enterprises. *International Journal of Operations & Production Management*, 14 (3) pp. 66-82.

5 Smeds, R. (1996) *Management of Enterprise Evolution. Evolution Management Principles and Methods for Learning Organizations*. Doctoral Dissertation. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 80, The Finnish Academy of Technology, Helsinki.

6 Smeds, R. (1997) Organizational learning and innovation through tailored simulation games: Two process re-engineering case studies. *Knowledge and Process Management. The Journal of Corporate Transformation* 4 (1) 22-33.

7 Riis, J., Smeds, R., Johansen, J. and Mikkelsen, H. (1998) Games for organizational learning in production management. In: N. Okino, H. Tamura and S. Fujii (Eds.) *Advances in Production Management Systems. Perspectives and future challenges*. Chapman & Hall, London, 327-338.

8 Smeds, R. (1998) Enterprise simulation laboratory for simulation games in virtual reality. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 13-23.

9 Schierholt, K. and Brütsch, D, (1998) Implementing process-oriented supply-chains in a logistics game. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 118-130.

10 Alstrup, L. and Busk Kofoed, L. (1998) "There is more than one way" - a game about production and working environment. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 131-143.

11 Gertsen, F. (1995) ADVANTIG - ADVANced Technology Implementation Game. In: Jens Riis and Hans Mikkelsen (eds): *Simulation Games and Learning in Production Management*. Chapman & Hall, London, 156-162.

12 Johansen, J. and Mikkelsen, H. (1995) The Lego truck game, In: Jens Riis and Hans Mikkelsen (eds): *Simulation Games and Learning in Production Management*. Chapman & Hall, London, 127-133.

13 Zülch, G. ,Jonsson, U. and Rinn, A. (1998) INSIGHTS - Integrated simulation game for a comprehensive redesign of production systems. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 105-117.

14 Mulder, K.F., Ree, C.M. and Mulder, H.A.J. (1998) Remodel, a game for strategic issues in industrial R&D and production planning. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 144-154.

15 Smeds, R., Tarkiainen, M. and Virtanen, T. (1996) Educational business simulation on the Internet, in Y. Merkuryev and Rapp, B. (Eds.) Abstracts of the International Conference on Simulation, Gaming, Training and Business Process Re-engineering in Operations, The International Federation of Operational Research Societies Special Conference, September 19-21, Riga, Latvia, p. 17.

16 Smeds, R. and Haho, P (1995) Tailored order-to-delivery process game. In: Jens Riis and Hans Mikkelsen (Eds): *Simulation Games and Learning in Production Management*. Chapman & Hall, London, 145-155.

17 Haho, P. and Smeds, R. (1998) The Softmatch-method: Enterprise Transformation through Simulation Games. In: Peter Saunders and Benita Cox (Eds.): *The Simulation and Gaming Yearbook Volume 5*. Kogan Page, London, 48-63.

18 Forssén-Nyberg, M. and Hakamäki, J. (forthcoming) Development of the production using participative simulation games, *International Journal of Production Economics*

19 Ruohomäki, V. (1995) A simulation game for the development of administrative work processes, In D. Saunders (Ed.) *The Simulation and Gaming Yearbook, Volume 3, Games and Simulations for Business*, Kogan Page, London, 264-270.

20 Haho, P. (1998) Tailored simulation games for successful business process development. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 24-38.

21 Forssén-Nyberg, M. and Kutilainen, P. (1998) Participative simulation game as facilitator of organizational development process - two case studies. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 39-49.

22 Pankakoski, M. (1998) How to disseminate knowledge and skills from using a tailored simulation game. In: R. Smeds and J. O. Riis (Eds.) *Experimental Learning in Production Management*. Chapman & Hall, London, 50-59.

23 Piispanen, E., Ruohomäki, V., Pankakoski, M. and Teikari, V. (1996) The Work Flow Game - a new method for developing office work. In: D. Saunders, F. Percival and M. Vartiainen (Eds.): *The Simulation and Gaming Yearbook Volume 4*. Kogan Page, London, 85-95.

24 Savukoski, E., Plukka, S., Enestam, V., Savolainen, M. and Piltonen, P. (1995) The Enterprise Game - real process simulation, In: D. Saunders (Ed.) *The Simulation and Gaming Yearbook, Volume 3, Games and Simulations for Business*, Kogan Page, London, 254-263.

25 Takala, T. (1996) Building a virtual reality space at Helsinki University of Technology, Internal Memorandum, Laboratory of Telecommunication Software and Multimedia, unpublished (in Finnish)

J. Teich[1], H. Wallenius[2] and J. Wallenius[3]

WWW-Based Support Tools for Business Negotiation and Communication Simulations

[1]Department of Management, New Mexico State University, Las Cruces, New Mexico 88003 USA, email: jteich@nmsu.edu

[2]Department of Industrial Management, Helsinki University of Technology, FIN 02150 Espoo, Finland, email: hannele.wallenius@hut.fi

[3]Center for Innovative Education, Helsinki School of Economics, POB 1210, FIN-00101 Helsinki, Finland, email: walleniu@hkkk.fi

Abstract

We describe a system in development that could be used for the simulation of two party negotiations. The system would be available on the internet and would allow synchronous communication tools to be used, such as the internet phone and chat, combined with analytical tools that may identify Pareto preferred alternatives. Potential uses of such a system include the possibility of role-playing in life-like negotiation situations, as well as training in advanced negotiation tools.

Introduction

Firms, their subsidiaries, subcontractors and customers are often spread throughout the world, placing increasing demands on communications and negotiations among entities. Long distance travel by employees is demanding, time consuming and expensive. This pressure to more efficiently conduct world-wide business transactions has lead to a natural increase in the demand for telecommunications.

The internet and various internal company intranets have experienced phenomenal growth over the last 3 years. All predictions point to continued exponential growth in their use. The number of households alone in Western Europe using the internet for shopping is predicted to increase from current 1% to 4% by year 2000. At the same time, the number of households possessing internet connections is predicted to increase from 4% in 1997 to 16% in 2000 in Western Europe. The growth in the US will be even faster. The technology offers a variety of options, such as classic e-mail, file transfer and now the more recent use of www applications including home pages, chat, internet phone, audio and video, and web broadcast systems.

An important, yet relatively undeveloped component for electronic commerce is the negotiation of contracts and the resolution of disputes. For additional details, see Teich, Wallenius and Wallenius [1].

Supporting Negotiations

DeSanctis and Gallupe [2] have classified systems whose primary goal is to improve communications between participants as level one Group Decision Support System (GDSS). They define level two GDSS by the use of analytical models to support negotiators. Level three GDSS contains an expert systems component. Level 1, 2 or 3 systems would have to be computerized to be of practical value. The systems that we refer to have originally been developed for a PC/Mac environment, although in many cases the authors have an interest in developing net versions of their systems.

Negotiation Support Systems (NSS) are GDSSs specifically designed for conflict resolution in the negotiation process. Three main reasons why a good NSS has potential in aiding negotiations are:

1. To increase the likelihood that an agreement is reached when a zone of agreement exists (solutions that both parties would accept).

2. To guarantee the Pareto optimality of an agreement. (See Teich et al. [3]).

3. To decrease the direct and indirect costs of negotiations, such as costs caused by time delays (strikes, violence), and attorneys' fees, among others.

The need for a NSS originates from negotiators often settling for suboptimal compromise agreements rather than integrating their interests.

The research field of NSSs is currently in the growth stage. The idea of using computers to aid the process of negotiations began in the 1960's. The last decade has witnessed the development of many NSSs. Several NSSs have been designed and tested, but as of yet, their full potential has not been realized by practitioners.

Description of a Negotiation Site
The negotiation site under development is expected to consist of the following features:

1. The site's home page itself, with access to multimedia communication channels, such as chat, voice, white board, bulletin board, video, all designed for public use, private between party use, private within party use and private individual to individual use.

   The main page will provide access to all features of the site: required programs download, descriptions of different services and features, guide, demo for newcomers, login for experienced users, and a search engine. The site could include an optional feature to gather data about every action taken during the sessions to document the negotiation process and/or outcome.

2. A matchmaking feature designed to bring together negotiating partners, such as buyers and sellers. Such features could include a submission form to include one's own information to a user's database and a search engine to find partners.

3. Analytical plug-ins or Java applets, incorporating negotiation support for the users with the purpose of helping parties structure negotiations, aid in locating an initial settlement, aid in finding Pareto preferred settlements, and ensuring the Pareto optimality of the settlement.

4. An example problem for negotiators to demonstrate the use of the support tools and other features of the site combined with a simulation game to encourage users to practice their negotiation skills and be trained in analytical techniques.

5. Documentation of the site
   - a video and/or a web-based animated guide to teach users the fundamentals of the site in multiple languages
   - hyperlinks to related sites
   - other links to literature, guides, reviews, conferences, other services etc.

Uses of the Site
Potential uses of this site are described below. They include, among others, the possibility of simulating negotiations in virtual reality, role-playing in life-like negotiation situations, as well as training in advanced negotiation support tools.

Using the Site in Teaching Negotiation Skills
The negotiation site could be used in training people in the use of analytical NSS techniques and the art of negotiations in general, the use of computers in negotiations, preparing for negotiations, and in integrative bargaining.

Experimentation with the site
· using experimental subjects to test human capability with different media (information overload)

- using experimental subjects to test the analytical support tools over the net in combination with different media
- other behavioral studies with human subjects to test the lack of direct human contact between negotiators
- performing negotiations with the help of our site with real negotiators with real-world cases, but in a simulated environment

Real-World Applications
Examples of such applications include:
- multiple issue, multiple party business transactions of buy/sell nature
- international dispute resolution (business disputes, political disputes)
- internal company negotiations and communications
- brainstorming
- divorce mediation

References
[1] DeSanctis, Gerardine and Gallupe, R. Brent (1987) "A Foundation for the Study of Group Decision Support Systems", *Management Science* 33:589-609.

[2] Teich, Jeffrey; Wallenius, Hannele; and Wallenius, Jyrki (1994) "Advances in Negotiation Science", *Transactions on Operational Research* 6:55-94.

[3] Teich, Jeffrey; Wallenius, Hannele; Wallenius, Jyrki; and Zionts, Stanley (1996) "Identifying Pareto-optimal Settlements for Two-Party Resource Allocation Negotiations", *European Journal of Operational Research* 93:536-549.

MODELLING ORGANISATIONAL MAPS : A NEW CHALLENGE FOR SIMULATION

P.Burlat[1], J. P. Campagne[2], G. Neubert[3]

1 Ecole des Mines de Saint-Etienne, 158 Cours Fauriel, F 42023 Saint-Etienne, Burlat@emse.fr
2 Ecole des Mines de Saint-Etienne, 158 Cours Fauriel, F 42023 Saint-Etienne, Campagne@emse.fr
3 INSA de Lyon, 20 avenue A.Einstein, F 69621 Villeurbanne, gneubert@ifhpserv.insa-lyon.fr

Introduction

This article will deal with the use of simulation in manufacturing management. Our aim is to show that simulation tools must now integrate a cognitive dimension to represent relevantly agile manufacturing. Indeed, models of enterprise processes should incorporate the interaction of human beings because enterprises are complex socio-technical systems, and also because competitiveness is now based on competence improvement rather than on factor optimisation.

Of course, distributed artificial intelligence techniques such as multi-agents systems improve those models in that way. But we will show that it leads to a different approach of modelling and to a different use of simulation.

From reactive towards proactive systems

Market instability and rapid development of new technologies can both be considered as uncertainty factors that increase complexity inside manufacturing systems. To meet the customers' needs and to improve their industrial performances, factories have to be reactive.

To be reactive, i.e. to be able to answer quickly and efficiently to unexpected events, industrial organizations must implement reactive systems based on a decentralized management. But they must equally be able to fit their reactive system to environmental changes or to internal changes.

This implies a supervision of those reactive systems by « proactive » systems capable of anticipating, of adapting resources allocation, of improving programs and management rules, in order to maintain the controlled system in a nominal running state.

Such a proactive system will encircle reactive systems

- it anticipates by adapting resources, plans or management rules to changes in environment or internal changes of the supervised system,
- it enhances its capacities of analysis and the efficiency of its decisions by a learning process.

To do so, proactive systems will be based on three principles : a decentralised management system built around autonomous agents, a management of noticed and foreseeable disturbances, a dynamic adaptation of reactive rules, based on environment analysis and observations of the state of the supervised system.

We represent in the following diagrams the differences between proactive level and reactive level

A proactive and reactive system
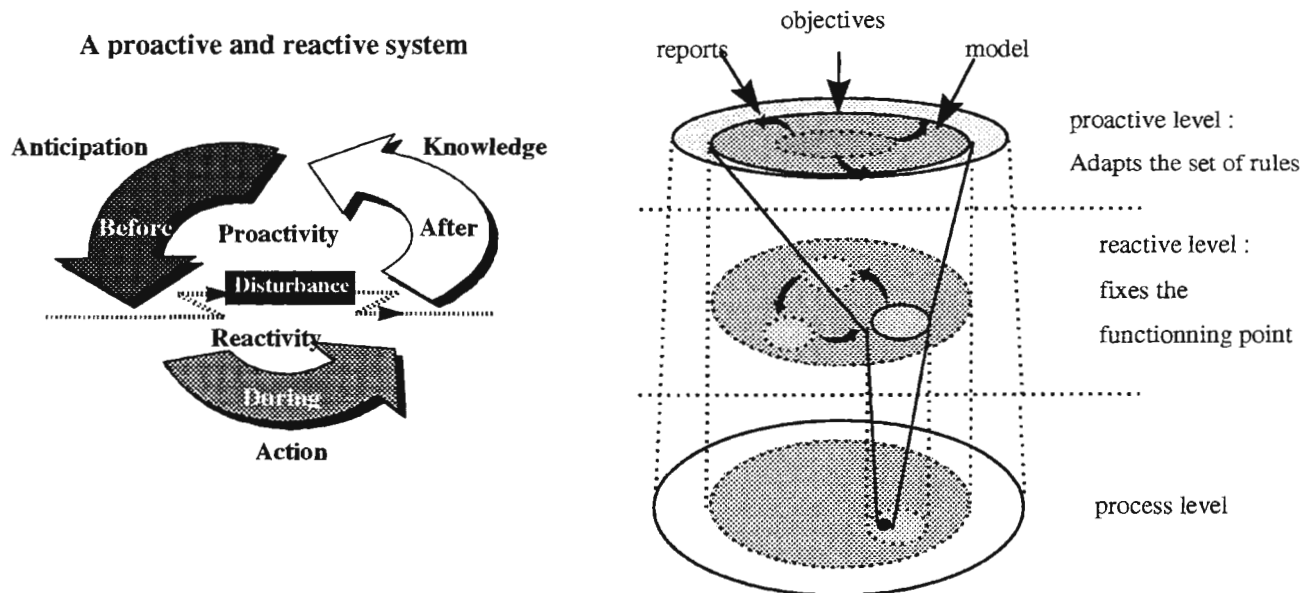
In this context, improving reactive behavior implies enhancing anticipation capacities, within a reactive loop process.

Of course, proactive systems shall be built on decentralised structures, in which decisions processes are distributed among autonomous agents.Each agent:

- fulfills a mission according to some objectives,
- uses some resources,
- combines reactive behavior and proactive behavior[5].

Those agents will make choices to reach their goals, according to their environment and to their knowledge. They will also increase their knowledge through learning loops. Developing such proactive systems is nowadays a way to ensure management's efficiency and to improve competitiveness.

Evaluation as a collective learning process

Thus, to meet the customers' needs, factories have to put a stress on anticipation and knowledge improvement. In this context, the internal efficiency of an organisation is determined by its own ability to design innovative solutions to deal with environment disturbances. Its real value relies on its capacity to implement collective learning processes.

After the "factors firm", which optimises physical efficiency with a static affectation of resources, comes the "competencies firm", which also takes into account qualitative criteria such as quality or variety, which bases its efficiency on innovation and creativity, and which develops its own manufacturing factors, through a constant learning process [2]. Such a dynamic process is based on procedural rationality [7], for incomplete information can only lead to a temporary solution that will be improved later with a new innovation step. In this context, possible choices are never entirely known ex ante, but are progressively built within the decision making process. We represent in the following table the main differences between factors firms and competencies firms:

431

FACTORS FIRM	COMPETENCIES FIRM
Functions	Processes
Optimisation	Satisfaction, improvement, innovation
Data processing	Cognitive processes
Allonomy	Autonomy
Information, instructions	Knowledge, learning
Substantive rationality	Procedural rationality
Negative noise	Positive noise
Perfect information	Incomplete information
Routines	Multi-criteria decisions

New aims for modelling and simulation

We know that traditional analytical management tools are no longer relevant when such a dynamic process is concerned. For instance, no optimum can be identified and found when systems are evolving permanently. But traditional simulation studies are often expressed in the form of an optimisation model. For instance, a simulation run will help a designer to determinate the right speed for conveyers which optimises the flow in a workshop.

That sort of studies are of course necessary. But they are not sufficient to characterise the performance of industrial organisations. As performance is no longer the report of an instantaneous efficiency but a dynamic process including innovation, we need to represent competencies and autonomy when modelling industrial systems. To do so, we may use distributed artificial intelligence techniques such as multi-agents systems and object oriented programming [6]. We also may use hybrid simulation, in which people are engaged as actors among a virtual reality model. Thus we are capable of modelling proactive organisations and of simulating for instance the effects of autonomy onto economical performances [2], or the effects of human relations onto competitiveness.

In doing so, we actually evolve from factors firm modelling towards competencies firm modelling. For example, within hybrid simulation routines will be turned into multi-criteria decisions making processes. Or else, a perfect information context in an optimisation model will evolve into a context of incomplete information due to the mere participation of human actors in the model.

Simulation as a collective tool

But this leads to a different approach of modelling and to a different use of simulation. We know that actions within the firm are not determined by objective reality - which is far too complex - but often by individual and collective representations of this reality. One of the aims of simulation is thus to supply people in the firm with collective models, with a view to shaping their collective behaviours and to giving them indicators capable of linking global orientations with local actions. For a proactive firm, modelling also means building representations that induce learning processes and entice innovation and creativity.

For instance, a simulator which includes cognitive agents will make it possible to assess the effect of evaluative indicators upon decision making process within the workshop, and finally upon the performances of the firm. After working with these simulators, people improve their organisational map [1] of the firm, thus improving their organisation without conflict [2].

In this context, modelling does not always mean building an accurate image of reality : a model will now be valid as soon as it enables people to communicate through a common representation of their organisation. Thus people within proactive organisations should use simulation both to improve the reactive level

(conception process) and to improve the proactive level (innovation process). But we think that firm modelling processing should now put a greater emphasis on innovation than on conception, in order to make people imagine innovative solutions to deal with environment disturbances.

References

[1] ARGYRIS C. and SCHÖN D. : *Organizational learning : a theory of action perspective*, Addison-Wesley Publishing Company, Massachussets, 1978.

[2] BURLAT P. : *Contribution to economical evaluation of industrial organisation : toward "competencies-firms" modelling*, Thesis, Univ. Lyon II, France, January 1996.

[3] BURLAT P., GIRARD M.A, OUZROUT Y. : "SIM'2 : a manufacturing management simulation game", ESM'97, Istanbul, June 1-4 1997, Turkey.

[4] NEUBERT G. and CAMPAGNE J. P : *New management systems : towards proactive factories*, IFIP TC5/WG5.7, APMS'96, Kyoto, November 1996.

[5] NEUBERT G. : *Contribution to disturbances management based on reactive and proactive control*, Thesis, INSA de Lyon, December 1997.

[6] OUZROUT Y. : *Modelling and simulating reactive production systems : a multi-agents approach*, Thesis, Univ. Lyon I, April 1996.

[7] SIMON H. : "From Substantive to Procedural Rationality" in *25 Years of Economic Theory*, edited by Kastelein, and others, Nijhoff, Leiden, 1976.

[8] VAN GIGCH J. P. : *System design modelling and meta modelling*, Plenum Press, 1991.

Measuring Factories Performance Using Data Envelopment Analysis

Livio Cricelli, Massimo Gastaldi, Nathan Levialdi
Dept. of Production, Systems and Computer Science
University of Rome - "Tor Vergata"
Via di Tor Vergata - 00133 Rome, Italy
E-mail: cricelli@disp.uniroma2.it

Abstract. The objective of this paper is to illustrate how a mathematical programming technique call Data Envelopment Analysis (DEA) can be used to measure the efficiency of factories of a multinational company, given the existence of multiple inputs and/or outputs in the transformation process with no *a priori* weightings. To show the DEA potentiality, an analysis on 12 factories in paper industry is presented. Besides identifying the inefficient units, a DEA assessment yields valuable by factories that can be used to determine a suitable course of action for improving the performance of the inefficient units.

Key Words: Efficiency, data development analysis, simulation in economy.

1. Introduction

One of the most relevant topic in production management is the development of integrated systems able to co-ordinate different technologies and applications, aiming at making the whole process of production efficient. In a competitive environment, industrial enterprises must continuously improve their productivity to sustain long term growth and profitability. To increase efficiency, without absorbing further resources, productivity measurement and analysis can therefore play an important role in the strategic planning and competitive analysis [1]. Productivity measures encompass estimations of technical change, technical, scale and allocative efficiencies. In standard microeconomic theory, a production process is efficient if it is not possible to augment any output without increasing any input or if it is not possible to decrease any input without augmenting any other input to obtain a prefixed output. Describing the technology of the production process with a production function, it allows to define the production frontier as the set of all technically efficient production combination. So measuring the distance between the production frontier and the production process, an estimate of the process efficiency can be obtained. The production frontier is not always known and, many times, only a set of observations corresponding to achieved output levels for given input levels is available. During the 1980's two different methodologies were used for constructing an empirical production frontier based on observed data:
- an econometric estimation of frontier;
- a version of linear programming called data envelopment analysis (DEA).

In the former, a parametric representation of the technology is assumed and the parameters are estimated utilising the observed data and introducing statistical noise, generally assumed to follow a normal distribution [2], [3]. The latter is a mathematical technique to develop a single measure of efficiency. In contrast with the econometric methods, DEA uses a non parametric approach and does not require any assumptions about the functional form of the frontier. This technique was designed by Charnes, Cooper and Rhodes [4] with the aim of measuring the relative efficiency of a set of Decision Making Units (DMUs) which are characterised by multiple, non commensurate inputs and outputs. The relative efficiency of each DMU is defined in a non-linear programming problem and is measured relative to all other DMUs with the only restriction that all DMUs are inside at efficient frontier.

DEA has been widely applied in recent years in operation research and management science. Applications of DEA include the evaluation of educational strategies, criminal superior court, school districts, national parks, hospitals [5], transports [6]. More recent applications regard the performance improvement of private and public organisations [7], [8]. and in particular, modern industries, such us microelectronics, telecommunications and manufacturing [9].

In this paper, using Data Envelopment Analysis an analysis on 12 factories of a multinational firm in paper industry is presented. Besides identifying the inefficient units, some guidelines can be used to determine a suitable course of action for improving the performance of the inefficient unit. The paper is organised as follows. In section 2 the main characteristics of DEA are presented. Section 3 applies DEA to a sample of factories of paper industry. Finally, Section 4 summarises the paper.

2. Data envelopment analysis

Data Envelopment Analysis, introduced by Charnes et al. [4], [10], is a powerful technique for measuring and improving the relative efficiency of a set of independent and homogeneous DMUs. In a process with a single input and a single output, efficiency is defined simply as:

$$Efficiency = \frac{output}{input}$$

More in generally, in a process there are several input and output and the efficiency can be defined as:

$$Efficiency = \frac{weighted\ sum\ of\ output}{weighted\ sum\ of\ input}$$

The problem of above definitions is in the choice of weights that is often difficult. The DEA solves this problem introducing a particular weight system for every single DMU. Charnes et al., in fact, propose that the maximum of efficiency for a unit k_0 can be calculated solving the following problem:

$$Max_{w,v}\ h_0 = \frac{\sum_{j=1}^{n} w_j y_{jk0}}{\sum_{i=1}^{m} v_i x_{ik0}}$$

subject to:

$$\frac{\sum_{j=1}^{n} w_j y_{jk}}{\sum_{i=1}^{m} v_i x_{ik}} \leq 1 \qquad k = 1,...,z$$

$$w_j, v_i \geq \varepsilon \qquad \forall j, i$$

where:

z = number of units;
m = number of inputs;
n = number of outputs;
w_j = weight given to output j;
v_i = weight given to input i;
ε = a small positive number.

This model maximizes the ratio of weighted output to weighted input of the k_0^{th} DMU, with the constraint that the same ratio for the other DMU's should not exceed unity which is the maximal efficiency. The efficiency of the unit k_0^{th} will either equal 1, in which case it is efficient relative to the other units, or will be less than 1, in which case unit is inefficient. The above equations are nonlinear, but Charnes et al. converted this problem in an ordinary fractional linear programming problem. DEA models can be divided

in input-oriented models and output-oriented models: the former favour the potential improvement of resource utilisation and the latter analyse the potential improvement of produced outputs. DEA models also solve linear programming problems for each technology satisfying both constant returns to scale (CRS) and variable returns to scale (VRS).

3. Evaluation of factories performance

This section presents the use of DEA models using a sample of 12 factories of a large multinational firm in paper industry. Software IDEAS (Integrated Data Envelopment Analysis System) is here utilised. IDEAS provides the users with the ability to perform Data Envelopment Analysis using a number of different models. Each of the various models has an implicit evaluation system that is used to individuate which of n factories determine an envelopment surface considering m inputs and s outputs. Factories that lie on the surface are deemed efficient in DEA terminology, while factories that do not lie on the surface are termed inefficient.

A typical factory in paper industry has two departments. In the first one a continue production process converts cellulose in large rolls of paper that weigh several quintals. In the second one the rolls are subjected to other processes: calendering, rolling, winding and principally cutting. In this department there are a batch production process with a large number of workers. Almost all factories of our sample have the two departments, but there are some of them that have only one department.

The DEA model is input-oriented and evaluates how efficient a factory is in producing paper. Given a set of n comparable factories, let k be the index of the factory whose relative efficiency is to be evaluated. In this model, the total tons of paper produced in a day is used to measure output produced (PROD). This production is also divided into the production of the two departments (PROD1 and PROD2). The following inputs are considered:
- total number of workers (WORK), divided in number of workers of the two departments (WORK1 and WORK2);
- quantity of cellulose utilised, measured in tons in a day (CELL);
- production potentiality of machines, measured in tons in a day (MACH) and divided in production potentiality of continue machines in the first department (MACH1) and cutting potentiality in the second department (MACH2).

In table 1, the data of our sample are shown. The observed inputs and outputs are the medium values of 1996 related to 12 factories of a multinational firms leader in paper industry with about 75% of Italian market and about 10% of European one.

The actual values of the weights are determined by the model using the following criterion: choose positive values of the weights so as to maximise the relative efficiency of the kth factory with the constraints that each of the n factories, using the same weights, has a relative efficiency measure that does not exceed unity. In this manner the optimal performance of the kth can be examined.

Fact.	WORK	WORK1	WORK2	CELL	MACH	MACH1	MACH2	PROD	PROD1	PROD2
A	500	210	290	228	915	600	315	754	514	240
B	510	205	305	180	1480	420	1060	902	402	500
C	350	140	210	116	945	220	725	360	200	160
D	465	180	285	110	775	360	415	450	310	140
E	490	195	295	216	1495	480	1015	780	470	310
F	450	450	-	578	1060	1060	-	980	980	-
G	430	430	-	340	620	620	-	571	571	-
H	230	230	-	51	168	168	-	151	151	-

I	110	-	110	-	415	-	415	78	-	78
L	290	290	-	88	400	400	-	380	380	-
M	330	180	150	224	725	445	280	440	350	90
N	195	95	100	145	640	200	440	245	175	70

Table 1: Observed inputs and outputs

The utilised model includes a single output and multiple inputs. Using a non-Archimedean model, the analysis has been carried out both for constant return of scale (CRS) and variable return of scale (VRS); to distinguish such analysis we call technical efficiency the results of DEA applied with constant return of scale and pure technical efficiency those obtained with variable return of scale. Considering the ratio between results obtained by the model with constant return of scale and those obtained with variable return of scale it is possible to determine a measure of scale efficiency.

Table 2 and Table 3 present the DEA model results obtained with CRS and VRS and ordered for decreasing level of efficiency. The second column shows the actual efficiency level (objective function - OBJ) while in the third column the adjusted level of efficiency obtainable with a equiproportional inputs reduction (respectively ITE; IPTE) is presented. The last columns report the % values of inefficiency for each considered input calculated as:

$$\Delta X_i\% = \frac{\Delta X_i}{X_{i0}} = \frac{(1 - ITE_i) \cdot X_{io} + E_i}{X_{io}}$$

Where X_{i0}, E_i and ITE_i are, respectively the observed value, the index of technical efficiency and the residual inefficiency for the ith input.

The more efficient factories are A, B, I, L; the linear combination of their activities represents a sort of projection point for all the other factories since we suppose in this phase the existence of constant return of scale. The average value of index ITE, for all factories, is 0.8966; this means that factories are characterised by a input surplus of about 10%. Building the Cross Efficiency Matrix, which conveys information on how unit's relative efficiency is rated by other units, it is possible to note that factory I is not rated with other units. This factories, in other words, is dissimilar to the rest of units and should not be considered in the analysis, since it is the only one without the department 1.

If our objective is to compare the efficiency of each factory to all the other factories, we have to use the model with variable return of scale where a convexity constraint (a convex activities combination of efficient factories) has been added. Since in this case the performance evaluation considers only the performance level of each factory, the efficient index (IPTE, Index of Pure Technical Efficiency) assumes values equal or higher than the corresponding values calculated with constant return of scale (ITE, Index of Technical Efficiency). This analysis shows that the number of factories considered efficient grows when Variable Return of Scale is inserted in the model. This result does not contradict the previous one since the DEA model with Variable Return of Scale computes for each factory the Index of pure technical efficiency with respect to the actual production level.

Fact.	OBJ	ITE	WORK %	CELL%	MACH %
A	1	1	0%	0%	0%
B	1	1	0%	0%	0%
I	1	1	0%	0%	0%
L	1	1	0%	0%	0%
G	0.86	0.99	1%	59%	1%
E	0.83	0.90	10%	28%	14%
M	0.78	0.83	17%	48%	17%

D	0.77	0.85	40%	15%	15%
F	0.73	0.91	9%	71%	9%
H	0.71	0.95	50%	31%	5%
C	0.61	0.62	42%	38%	38%
N	0.56	0.71	29%	66%	37%

TAB 2: DEA model results with CRS

Fact.	OBJ	IPTE	WORK%	CELL%	MACH%	ISE
A	1	1	0%	0%	0%	1
B	1	1	0%	0%	0%	1
G	1	1	0%	0%	0%	0.992
H	1	1	0%	0%	0%	0.946
I	1	1	0%	0%	0%	1
L	1	1	0%	0%	0%	1
N	1	1	0%	0%	0%	0.710
E	0.907	0.921	8%	20%	11%	0.976
M	0.889	0.923	8%	46%	8%	0.904
D	0.767	0.912	31%	9%	30%	0.936
C	0.755	0.798	20%	20%	56%	0.782
F	0.700	0.924	8%	75%	8%	0.984

TAB 3: DEA model results with VRS

This means that all the factories with ITE=1 assume IPTE=1. The estimated inefficiencies individuate inputs waste to eliminate in the short period before reorganising factory towards an high level of productivity. Now we can isolate the inefficient components linked to the production scale; the scale efficiency ISE showed in the last column of table 3 can be obtained from the results of CRS and VRS models by means the following ratio:

$$ISE_j = \frac{ITE_j}{IPTE_j} \quad j=1,\ldots\ldots,12$$

Figure 1 shows the comparison between indexes ITE and IPTE, with the factories ordered by growing value of production. High values of ISE are obtained for factories with a great quantity of produced paper demonstrating the positive impact of dimension on scale efficiency. In general, the greater factories are the more efficient, with the exception of factories I and H.

Figures 2 and 3 show the % values of inefficiency obtained, respectively, by CRS and VRS models, with the factories ordered by growing value of production. In general, the values of inefficiency for workers and machines are decreasing with production, while those for cellulose are increasing with production. In order to define the source of the inefficiencies an analysis of the single departments can be performed.
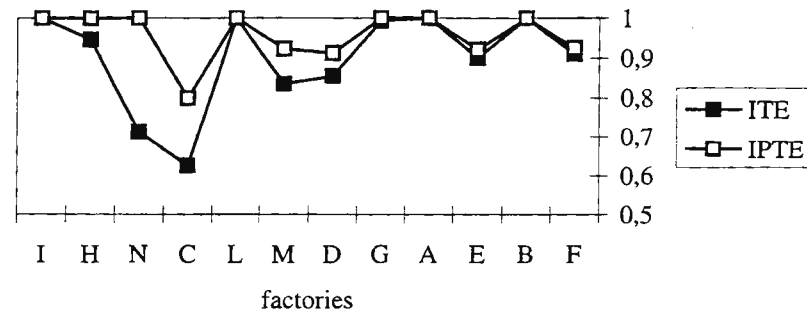
Fig. 1.: ITE and IPTE comparison
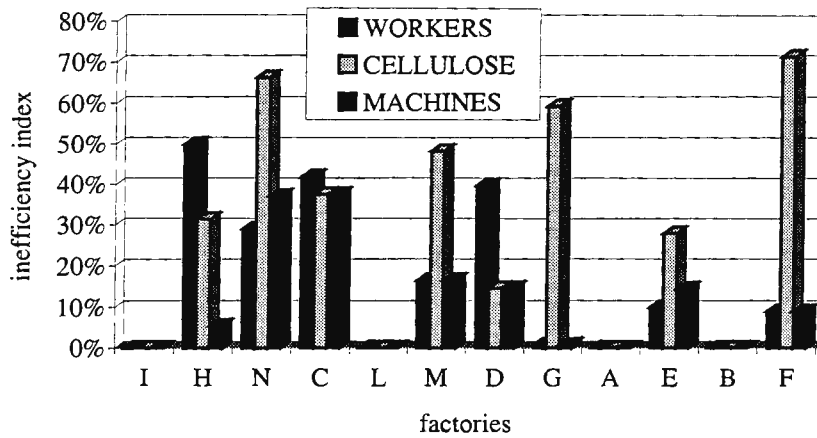
Fig. 2: Inefficient % obtained by CRS model
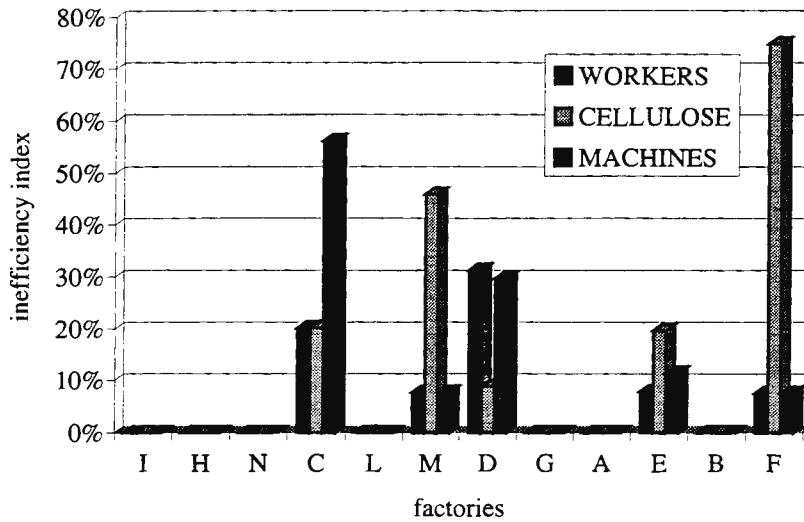
Fig. 3: Inefficient % obtained by VRS model

Comparing obtained results by the three CRS, Input-oriented models (department 1, department 2 and all factory) we can note that:

• factory A, having modern plants, always results efficient;
• in the department 2, factory B assumes the higher level of efficiency, while in department 1 it could

increase its efficiency by a small reduction (1%) of all inputs;

- factory E results, instead, strong damaged by its department 2, whereas, in department 1 it is the more efficient factory of the sample.

The main problems can be individuated in department 2, principally for an high excess of machines. This is due to the technological binds of the cutter machines that are the real bottleneck of the whole production process. A possible solution to improving performance is in the introduction of an order scheduling. Notice that the two factories (A and B) that use such methodology are already efficient. The obtained results for department 2 show an excess of workers in all inefficient factories, probably, due to the difficulties to manage the Job-Shop process in this department.

In department 1, we can note that machines are rather old and that they would be renewed. With referring to cellulose, the higher inefficiencies are in factories M and N. While in the first case it is due to the particular kind of produced paper (rich in cellulose), in the factory N there is a probable waste. Unfortunately we are not able to perform cost-efficiency analysis for lack of significant data about the real cost of several factories input.

4. Conclusions

In this paper the utilisation of DEA models to evaluate the efficiency of factories is discussed. Moreover an application of DEA methodology on a sample of factories in the paper industry is also presented. We show that DEA is able to let inefficiency component relatively at any single input and therefore to determine objectives to pursue in the short period (VRS) and in the long period (CRS).

References

[1] Farrell M.J., "The Measurement of Productive Efficiency", Journal of Royal Statistical Society", A 120, 253-281.

[2] Aigner D.J., Knox Lovell C.A. and Schmidt P. "Formulation and Estimation of Stochastic Frontier Production Function Models", Journal of Econometrics, 6, 21-37, 1977.

[3] Aigner D.J and Schmidt P. "Specification and Estimation of Frontier Production, Profit and Cost Function", Journal of Econometrics, 13, 1-138, 1980.

[4] Charnes A., Cooper W.W. and Rhodes E. "Measuring the efficiency of decision making units", European Journal of Operational Research, 2, 6, 429-444, 1978.

[5] Sengupta J. K. "Data envelopment analysis: a new tool for improving managerial efficiency", International Journal of System Science, 27, 12, 1205-1210, 1996.

[6] Chu X., Fielding G. J. and Lamar B. W. "Measuring Transit Performance using data envelopment analysis", Transportation Research, 26A, 3, 223-230, 1992.

[7] Charnes A., Cooper W.W. and Lewin A. Y. and Seiford L. M. (editors), Data Envelopment Analysis: Theory, Methodology and Applications, Kluwer Academic Press, Boston, 1996.

[8] Hatry H. P. "Performance measurement principles and techniques: an overview for local government", Public Productivity Review, 4, 312-339, 1980.

[9] Shafer S. M. and Bradford J. W. "Efficiency Measurement of Alternative Machine Component Grouping Solutions Via Data Envelopment Analysis", IEEE Trans. on Eng. Management, 42, 2, 159-165, 1995.

[10] Charnes A., Cooper W.W. and Rhodes E. "Program Evaluation and Managerial Efficiency: an Application of Data Envelopment Analysis to Program Follow Through", Management Science, 27, 6, 668-697, 1981.

Complete Modeling of Production for the Design of Planning and Control

Ph. Moscoso, H. Ulrich

Institute for Operations Research, Swiss Federal Institute of Technology Zurich (ETH), CH-8092 Zurich, Switzerland,
e-mail: moscoso@ifor.math.ethz.ch

Abstract

In a business era characterized by a dazzling rate of change, the improvement of production planning and control begins to be a main objective for manufacturing industries. Based on the analysis of a prototypical solution for a highly automated production planning and control supported by simulation which is realized in practice, some basic requirements for the design of planning and control systems are derived. Particularly we illustrate, how the socio-technical and dynamic nature of production systems implies important consequences for the development of corresponding planning and control systems. Decisions concerning function allocation between human and technical sub-system as well as the consideration of the entire industrial operation are shown to be crucial for the envisaged system design. With the analysis it became apparent that the lack of completeness of the developed planning and control system is a reflection of the technocratic perception of the production system on which the design was based. We realize that production models required for the design of planning and control systems (i.e. design models) cannot be identical to production models required for planning and control of production systems (i.e. regulatory models). This led to the conclusion, that there is a pressing need for system modeling techniques that support the process of system design. Without such adequate techniques, designed planning and control systems will not be sufficiently reliable, maintainable and flexible, will be difficult to comprehend, and their elements will not be re-usable. In addition, the fundamental relevance of providing a system model in the context of system design is investigated from a cognitive ergonomic point of view. Furthermore, depending on the chosen design procedure, very different levels of abstraction must be covered by the modeling concept. Considering these fundamental insights, a general modeling approach that supports complete design of planning and control systems, and which differentiates two generic modeling concepts is outlined. The presented modeling framework is finally complemented with a valuable instrument for the integrated evaluation of design strategies.

Keywords

production planning and control, system design model, regulatory model, simulation, production systems.

Introduction

In the face of today's business environment, which is becoming ever more competitive and global, only constantly evolving companies will remain viable. For manufacturing industries, this requires a permanent revision of production systems. In addition, increasing structural and dynamic complexity of production processes have led to the realization that an exact prediction of process states cannot be guaranteed [cf. Malik 1996]. Consequently, new planning and control systems based on new design strategies and concepts are needed where unsteady-states of production are considered as the norm, and even as a presupposition for organizational learning. Basically this means to manage a production system trying to optimize certain skills (e.g. flexibility), instead of certain states (e.g. utilization). In conclusion, to remain competitive, the improvement of planning and control systems must become a main objective equal in priority to product innovation [cf. Günther et al. 1997]. The first issue this paper wants to point out is, that a real design improvement of planning and control systems in an environment characterized by strengthening interdependency of relationships will only be possible if a fundamental change of mind from a static to a dynamic as well as from a technical to a socio-technical perception (i.e. complete perception) of production systems takes place. Acknowledging this, the second issue of this paper consists in the analysis of the potentials and possibilities of a simulation-supported modeling of the production system for the complete

design of the production planning and control system, and the corresponding development of a modeling framework for this purpose.

The paper starts with a brief description of a planning and control system realized in practice. Derived conclusions for the improvement of the system design are then pointed out. It follows a discussion of the relevance of simulation-supported production models for the design of planning and control systems. In the next two sections, requirements for the development of a complete model of production systems are analyzed. The gained insights are considered for the development of a modeling framework for complete system design. In the next section a valuable instrument for the evaluation of design strategies is presented, and integrated into the modeling framework. Conclusions are drawn in the last section.

Lessons from practice: a case study as starting point of research

The conclusion concerning the fundamental importance of an appropriate design of planning and control systems for production outlined above, is to a large extent the result of an interdisciplinary research project together with partners from industry. The starting point of this research project was the analysis of a highly automated work-cell for semi-conductor assembling with special regard to its integration into an existing manufacturing process. The work-cell is built up by several production facilities and the transport within the work-cell is performed by a robot. Despite the high degree of automation, at least one operator is needed for each work-cell to perform manual tasks like resetting and quality control.

A special focus of the analysis was put on the planning and control process of the work-cell. To provide support for the derived scheduling problems showed up to be a difficult task. Depending on the user of the work-cell different optimization criteria and several different restrictions may be formulated. In addition, solutions calculated with classical scheduling approaches considering only the static problem soon become obsolete in this dynamic environment. Planning and control have to be flexible allowing the integration of new equipment types to the work-cell and supporting the integration of new process steps. The time to search a solution has to be short and the information to deal with could be fuzzy (e.g. 'this lot has high priority'). Confronted with all this difficulties, the work-cell company asked our institute to develop an intelligent and adaptive planning and control system for the work-cell. For this purpose, the use of an agent-based planning and control system, where scheduling problems are distributed to virtual agents was envisaged [cf. Wooldrige *et al.* 1995]. In the developed system, each of these agents supports a clearly defined task and has its own specific problem solving strategies. Agents have information about the problem area they are responsible for (e.g. a group of equipment) and they can request services from other agents. Events from the ongoing process are forwarded in real time to the agents. The agents compare the actual state of the process with the planning data taking appropriate actions if necessary.

Because of the high dynamic complexity of the interrelated process-steps, the use of a simulation model by the agents to determine the future consequences of their decisions, allowed important improvements of the planning and control [cf. Lüthi *et al.* 1996]. The simulation model was developed under a strictly logistic view using our discrete event simulator HIDES (Highly Interactive Discrete Event Simulator, [cf. Graber *et al.* 1993]). HIDES is a general purpose simulation framework which is designed for event-oriented simulation and implemented in an object-oriented environment. It allows the building of hierarchical models with different levels of abstraction. HIDES-models comprise a physical and a logical level. In the physical level, the static structure of the system and the dynamic sequence of the physical events are represented. The relationships are modeled as directed graphs with two types of nodes, stations and queues. Resources are introduced as abstract elements that can be allocated dynamically containing information about their availability and their behavior. The logical level comprises the control of the system, together with the information flow which is necessary for this purpose. In order to embed the developed agents in our simulation platform, an extended level (i.e. 'management-level') was specified. This extension forms the basis for developing the coordination process among the agents, and therefore can be considered as an extension of the described logical level. The agent-level allows to illustrate a management concept which formally regulates the competencies and responsibilities of the agents.

But the introduction of the first generation of work-cells showed two major problems with regard to the integration of the work-cell with its internal planning and control system into an existing manufacturing process. It became apparent to us, that one key error in the followed design approach was to ignore the socio-technical character of the production system, or more precisely, only technology was considered as enabler of dynamic process planning and control. The other key error was to neglect, that the work-cell only supports a few manufacturing steps within the entire industrial operation and therefore previous and following process steps must exert a more deliberate influence on the design of the planning and control system. Summarizing, the design of the planning and control system, including its integrated simulation model, was very technocratic, and considered only the process steps supported directly by the work-cell.

From a more theoretical perspective, the mayor relevance of this case study for us was to comprehend, that to understand and assimilate the complex nature of production systems is of fundamental importance for the appropriate design of planning and control systems. We had to accept, that with our technocratic design concept we had ignored one of the basic theorems in cybernetics, i.e. the Conant-Ashby theorem which states, that every controller must contain an adequately rich model of what should be regulated [Conant *et al.* 1970]. The lack of completeness of the developed planning and control system, basically was the reflection of a technocratic perception of production systems. In consequence, the exposed error of disregarding the complexity of the production systems during the development of the planning and control system seems to be of basic nature and therefore deserves a deeper investigation. We have to analyze more carefully which are the requisites for the development of a complete model of production systems, in order to be able to formulate a general modeling approach that allows to improve the design of planning and control systems. Due to the exposed complexity of the manufacturing process, only a complete system model could allow to clarify the effects of design strategies (e.g. estimate their efficiency in advance), and to analyze different questions related to the proper operation of the planning and control system. But first of all, considering the exposed difficulties and risks of modeling production systems, the question of the relevance of developing a complete system model that could support the design process, deserves a more careful analysis.

The potentials of models for system design

The addressed problem of designing planning and control systems is more than classical problem solving. In most cases there is no well-defined problem given, which alone must be solved. Instead, initially the key problem itself must be defined or recognized and different solutions must be considered. If several variants of solution principles are to be developed, then the most promising of them should be recognized in order to finally master the posed problem. However, this procedure is not only rule-governed deduction, but also creative induction. The designer should compare the characteristics of solution variants, recombine characteristics, introduce new characteristics or reject irrelevant ones. Here the design procedures of human beings strictly differ from those of computer programs. Human, in most cases, will not start from the details and proceed to a complete solution. Rather they will start from vague impression of a complete solution and proceed to details. Solution ideas usually are fixed as an abstracting sketch of the solution principle and are specified afterwards within feedback-circles of thinking, sketching and critical inspection of the refined sketch. Due to the limitations of human mental capacity (i.e. the 'working memory'), this human design procedure is very open to external support as simulation models. Without support relieving working memory load, designers are forced to be limited to a section of features only. They will be restricted not because they could imagine further feature patterns, but since they cannot hold all the patterns within consciousness [cf. Hacker 1997].

Given that the designer developed a search space in mind, the designer must then operate on its representations, because the final solution will depend on whether the relevant solution possibilities were represented or stored at all. The mental representations are the search space, and no solution will be better than the search space allows. Consequently, a simulation model of the production system would be very helpful for the design of a planning and control system. In this sense, a system model (i.e. specially simulation-supported system models) could fulfill all of the main functions of prototypes during a design process [cf. Hacker 1997]:

- a means of problem analysis,
- an aid to create new ideas,
- an instrument of evaluation,
- a means of storage relieving working memory, and
- a communication support.

Therefore the development of a general modeling concept for system design seems to be very promising from a cognitive ergonomic point of view, and a more detailed analysis of the demands to a system model for complete system design must take place.

The complex nature of production systems and derived consequences

The concept of designing processes is coming to occupy an increasingly central position in industrial practice because of an improved perception of the difficulties related to the dynamic complexity of production systems. The relevance of an adequate dealing with time calls for a deeper investigation of processes abandoning pure structural descriptions to understand them in their dynamic context. In fact, associated with the growth of a process oriented perception of production systems has been the development of a wide variety of modeling concepts using simulation for the design of production systems [cf. Gallasch 1996]. However, these new concepts are often insufficient for supporting complete system design. The set of variables represented by a model as well as its interactions, is conditioned by both the modeler and the modeling technique. Comparing different design solutions (e.g. by computer simulation), only those variables represented by the model are considered in deriving design scenarios. The effect of variables that are not represented by the model (e.g. the human operator's motivation) are neglected. Most modeling concepts focus on technical aspects of manufacturing processes, representing only 'hard technical' (i.e. quantifiable) variables as production time and throughput. 'Soft variables' as skills and potentials of operators as well as the influences of a given organization, are not considered at all. Consequently, the developed model would be the same for an assembling factory in Asia as for one in Europe, regardless of the different social and organizational characteristics they could present.

But we learned from our case study, that a design based on such a technocratic modeling concept will lead to a highly intricate planning and control system, which leaves little room for autonomous decision-making of human operators. This, in addition to insufficient training or information for personnel involved in and affected by planning and control, normally will revert to an informal planning and control system, with foremen keeping their own additional databases in order to be able to keep due dates despite numerous delays and changes [cf. Wiendahl 1997]. In consequence, to improve the design of the planning and control system, we need a model of the production system which takes into account in an integral manner technical, organizational and human considerations [cf. Ulich 1994]. Only then we will be able to face the challenge of taking human and organizational characteristics into consideration within the design of the planning and control system.

The challenge of modeling a production system more completely for system design purposes, doesn't seem to be very promising at first sight. How shall influences like motivation of human beings be modeled in a simulation? It is true that computers are numerical and precise. However, it is also true that motivation is a very real phenomenon. Indeed, precisely such things as motivation provide production systems with their essential character. Besides the development of an interactive simulator which is only practicable for the design of single human-machine systems, one of two basic options could be chosen for modeling production systems. Either a numerically precise model that certainly will miss some essential aspects of system behavior is formulated, or a certain kind of precision is given up, in order to include variables playing a critical role in generating the system behavior.

For the generation and posterior dissemination of a production model we are therefore confronted with a basic dilemma. We must keep a balance between the required complexity of the production model needed for the proper understanding of the production system and the required simplicity of the model in order to allow to understand and share at least some aspects of the developed system model by the different persons involved in the design of the planning and control system. On the one hand, the mentioned necessity of reducing the system

complexity means that only a selective input can be taken into account which implies the risk of neglecting fundamental aspects of the production system. As a result, even the recognition of major problems cannot be guaranteed. On the other hand, if the different designers which never have been involved in the production itself must be able to share a model of the production system, it must be simple in translation and acceptable by their different cultural backgrounds. This implies, that the system model on which the design is based, usually cannot be identical to the regulatory model of operators involved in the planning and control. It is unlikely, in fact, that designers (or even the production managers) are able to understand the different regulatory models of operators, because the models in the cognitive domains of the operators have been developed under different influences of knowledge, experiences and interactions. But there is no need for the designers to understand these complex regulatory models, if a participatory design approach is envisioned, such that future system operators are not excluded from the design process. The fundamental conclusion must therefore be, that the primary purpose of the envisioned system model for the design of the planning and control system is to improve the system design, and not a regulatory purpose. In conclusion, system models for such a learning purpose must be simple enough to be shared as precondition for organizational learning, and therefore can usually not be the regulatory models themselves but statements about the interactions between them [cf. Espejo et al. 1996].

Recognizing this insights means to realize, that a successful design of the planning and control system depends decidedly on the quality of interaction between the involved operators and the developed technical sub-system, or more precisely, between their corresponding regulatory models. One should first decide which sub-system should do what and how the sub-systems should communicate. The function allocation between human and technical sub-system is a crucial question for the design of a planning and control system which must be covered by a modeling concept for design purposes. In consequence, the level of automation within a production system, i.e. the opportunities human operators have to organize and intervene, must be adaptable to match the available qualification of the staff as well as the cultural environment. To accept that the adequate interaction between human and technical sub-system provides a performance impossible to human or technical systems alone [cf. Bailey 1989] means that an automated planning and control system supported by simulation cannot limit this adaptability of the production system. Instead, we should try to empower human performance by providing human operators with job characteristics that permit them to gain experience in process planning and control in order to counter the weaknesses of the technical sub-system [cfGrote et al. 1996].

Additional requisites for a complete modeling approach

Designing a planning and control system is a working activity that involves more than thinking. To a great extent it is a knowledge-based development of solutions. Thus, it integrates the reactivation of already known solutions and the development of new ones. Through this integration, the design process becomes an opportunistic procedure. It does not proceed systematically towards the goal, rather the procedure will start at some isolated island of knowledge and try to rearrange and combine them stepwise by thinking on different levels of abstraction [cf. Hacker 1997]. Basically this corresponds to the necessity of supporting the specification of different levels of abstraction with a modeling concept for design support. To provide this support is specially difficult if the two main exponents of opposite design approaches are regarded, i.e. continuos improvement and radical re-engineering [cf. Pielok 1995]. Whereas re-engineering concepts follow a synoptic vision for design based on a constructivistic perception of production systems, evolutionary design approaches postulate, that because of the limited understanding of the system complexity, it is only possible to develop a situated vision for design [Malik 1996]. In consequence, for the design of planning and control systems comprising human operators and technical sub-systems very different levels of abstraction are required, and a dual modeling approach which differentiates between a meta- and an object-model is needed [cf. Beckmann 1996].

On the aggregated meta-level, the modeling of the relations between the elements of the systems as well as between the system and its environment, together with the formulation of design principles which should allow to reach specified goals, are of interest. This must be done in a more abstract way, because the purpose is to enhance the understanding of the system complexity through the recognition of behavioral patterns. The meta-

model figures as an orientation framework for the incremental design on the object-level, where the situated, and therefore more detailed analysis of the system is the center of attention. This differentiation of modeling levels, allows the combination of two types of changes. For incremental modifications, the design discussions can normally be reduced to the object-model as long as they do not affect the orientation framework of the meta-level (e.g. questioning some fundamental design principles). In order to guarantee the continuity of the design process both modeling levels have to be linked by a common process model. Summarizing, this differentiation allows the integration of periods of radical innovation within a superposed evolutionary design process.

Furthermore, the exposed differentiation is also important regarding the different phases of the design process in detail. Decisions taken during the early phases of the design process, that is first, the identification and clarification of the problem and then the development of the frame conception of solutions, will have the highest impact on innovativeness and costs of design [Ehrlenspiel 1995]. But in this first phases of design we have to deal with high uncertainty concerning its results. This offers a challenge to proceed to innovative solutions, but it also means taking the risk of detrimental decisions and failures. For this phases we believe that an abstract system model in the sense of the meta-model is very helpful, because it could make the desired feature patterns of selected solutions more explicit, and consequently make the solution space more transparent, whereas for the later phases, that is, the design of the favored solution, and the working out of the details, the object-model is a bigger support.

A modeling approach for complete system design

For the mentioned purpose of developing a complete model of production systems as a valuable base for the design of planning and control systems, we developed a system-theoretical framework for system modeling which comprises to modeling levels, the meta- and the object-level. Due to the mentioned dynamic complexity of production systems, processes are the center of attention of the modeling framework. Through the specification of a common generic process model, the two model levels are linked.

The development of the meta-model of production systems focused on the specification of the fundamental dimensions of a complete system description as a generic system model, in order to guarantee a complete design of the planning and control system which considers the complexity of production systems [cf. Moscoso et al. 1997]. Therefore, a feature catalogue within a conceptual modeling framework was defined. The catalogue should be understood in the sense of showing the levers for tackling the system design. The chosen set comprises seven modeling dimensions: management, tasks, structures, processes, resources, marks and skills (cf. Fig. 1). Whereas marks and skills are consequences of system design, management, tasks, structures, processes and resources are dimensions of system design. The former are dependent, the latter independent variables of system design. In the management dimension the framework for the specification of tasks is determined. Together with the tasks dimension, the management concept establishes the base for the design of processes, structures and resources. To consider the specificity of operators within the work-system, criteria for complementary design were integrated into to the modeling framework. These criteria were developed by the work and organizational psychology unit of the Swiss Federal Institute of Zurich, and concern design on the three levels of socio-technical system, individual work task and human-machine-interaction [cf. Grote et al. 1996].

According to the purpose of supporting also more detailed design phases, the concept of the object-model was developed in a second step. It follows an object-oriented modeling approach (cf. HIDES) using the close relation between this approach and the general system theory. In addition, this modeling approach allows the use of object-oriented simulation platforms like HIDES for design purposes. This is of fundamental importance, because simulation is the best methodical tool in order to gain insight into the dynamic complexity of production systems. For dynamic systems it is necessary to go around the loop and consider the effect of feedback. This is a basic requirement for the design of planning and control systems.
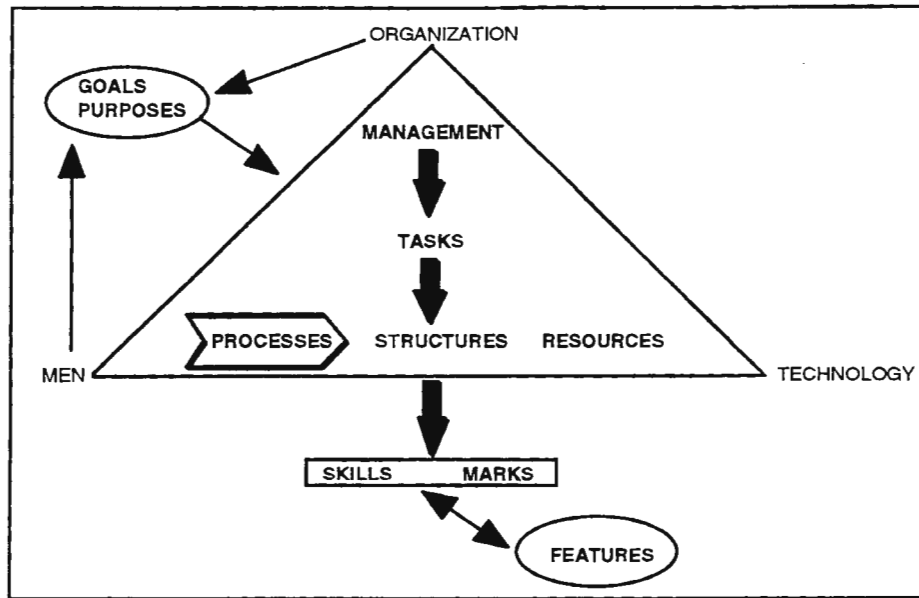
Fig. 1: The meta-model framework

The adaptation of the object-model to the meta-model was possible through the specification of four recursive levels of description. Starting from the object-independent description of tasks, functions are described by relating a task to a specific object-class entities, the instances of which have certain attributes that are transformed by the task. On the next level activities are described by relating the functions to an object-class resources, the instances of which allow the transformation, and an object-class agents, the instances of which control the transformation. Using this generic object-classes, a specific activity is described completely by some instances of this classes. The concepts of functions and activities represent the formal connection between the task and the process dimension required for the concrete object-model. When activities are structured following a defined sequence a process is formed (cf. Fig. 2). This definition of processes represents our generic process model.
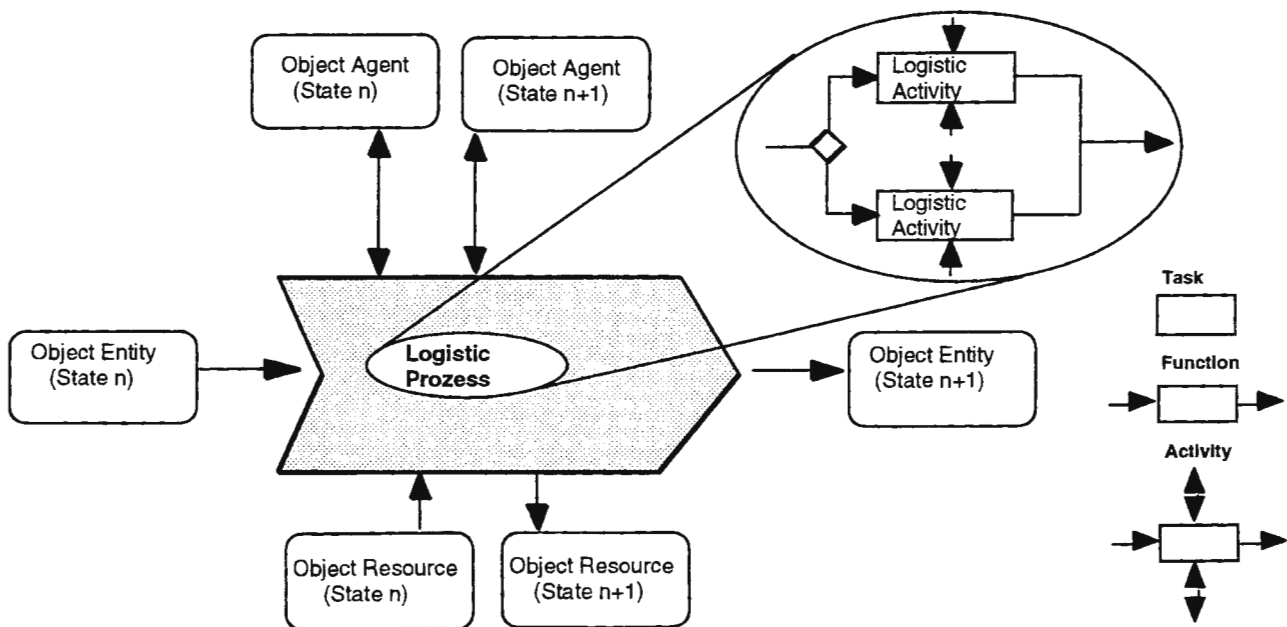
Fig. 2: The object-model framework

447

This object-oriented modeling concept describes a system by identifying the key object classes, and specifying their behavior and their relationships with other classes. Several object-oriented modeling technique for design and analysis purposes have been developed [cf. Booch 1994]. For the specification of the object-class agents two point must be taken additionally into account. From a external viewpoint, agents are characterized by their purpose, their responsibilities, the services they perform, the information they require and maintain, and their external interactions. From a internal point of view, we need to impose structures upon the informational and motivational states of the agents and control structures which determine their behavior. Both aspects are considered with the specified agent-level in the HIDES reference modeL[ithi et al. 1996].

With the exposed modeling framework which differentiates two modeling concepts (i.e. the meta- and object-model), we have developed a framework for the complete modeling of production systems. The modeling approach provides through its meta-model a general system model as reference for system design. Selected design solutions can be modeled in detail with the concept of the object-model and simulated with our simulation platform HIDES.

Evaluating design and production strategies

A fundamental requirement to a modeling concept that should support design of planning and control systems is, that it provides instruments for a complete evaluation of design strategies in order to guide further operations. For the design of planning and control systems the evaluation must take into account in an integral manner the basic objectives of manufacturing control, i.e. inventory, flow time, utilization, and lateness. This necessity for a transparent evaluation was already considered through the specification of the mark dimension in the exposed modeling framework. In the following, we outline an instrument which provides the required features.

The quality of the logistic performance is primary determined by lead time and schedule performance in the job shop. On the other hand, loading of the production facilities and work-in-process (WIP) influence the profitability of the manufacturing process. From these partly conflicting requirements concerning the production planning and control of the work-system the four main objectives of production processes were derived (cf. marks in the meta-model). Short lead time and low schedule deviation represent the market-related objectives, whereas low inventory and high and steady loading of the work systems are the factory-related objectives. We now can analyze the defined marks of the modeling framework (e.g. load or lead time) and evaluate different design strategies for each process by simulating the logistic operating curves of a given work-system. The logistic operating curve forms the connection between the mean WIP, the mean weighted lead time and the output rate using the capacity and the ideal minimal WIP (order structure) as input parameters [cf. Wiendahl 1997]. For a given curve, we can decide in order to adjust the logistical objectives under the consideration of their mutual dependencies which design strategy seems to be more appropriate. On the one hand we can discuss in which operating state on the characteristic logistic operating curve the system should be driven, e.g. which utilization the work-system ought to reach or, which is the necessary capacity it has to offer for a certain utilization. On the other hand we are able to predict the influences of changes in the system (e.g. increasing production speed) through the simulation of the new derived curve structure.

With the concept of the logistic operating curves we have an instrument, that principally can be used for two different modeling applications. First, it provides a fundamental reference framework for the expedient design of the planning and control systems in accordance to the specific customer demands. Second, it allows a continuos monitoring and diagnosis of the manufacturing process as well as the scheduling of the order throughput, and represents therefore a valuable support for the regulatory model of operators. For the presented modeling approach with its two modeling concepts, the concept of the logistic operating curves is therefore a fundamental complement to the qualitative criteria in order to support a prospective design of production systems.

Conclusions and prospects

The referred case study has emphasized the necessity of a complete design of planning and control systems, as well as the potentials and possibilities of a complete, simulation-supported modeling framework for this purpose. We have shown that to describe the structural and dynamic complexity of production systems the organizational and human peculiarities must absolutely be taken into account. Actually, formulated models are not intended for simulating operators as virtual agents, but for attaining a mutual understanding between customer and designer of system characteristics as a valuable base for system design and prospective learning. The provided terminology is specially important and allows the explicit formulation of often not mentioned assumptions (e.g. the mental models of human beings). However, the presented modeling approach is not to be understood in the sense of leading to one straight forward solution, but as part of a modeling concept which enables the reflection of design decisions and the proper use of simulation for this purpose.

The aim of further research based on more case studies is to improve the concept and to formulate a general methodology for the design of planning and control systems. For this, a more extensive study should examine, among other aspects, the methodological principles to be made explicit in instructions for the use of the modeling approach.

References

Bailey, R.W. (1989). *Human performance engineering*, 2nd ed., (Prentice-Hall International, London).

Beckmann, H. (1996). Theorie einer evolutionären Logistik-Planung. In: A. Kuhn (ed.), *Unternehmenslogistik*, (Verlag Praxiswissen, Dortmund).

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications*, 2nd ed., (Benjamin/Cummings, Redwood City).

Contant, R.C. & Ashby, W.R. (1970). Every good regulator of a system must be a model of that system, *International Journal of System Science*, Vol. 1, No. 2, 89-97.

Ehrlenspiel, K. (1995). *Integrierte Produktentwicklung. Methoden für Prozessorganisation, Produkterstellung und Konstruktion*, (Hanser, Munich).

Espejo, R., Schumann, W., Schwaninger, M. & Bilello, U (1996). *Organizational Transformation and Learning. A Cybernetic Approach to Management*, (Wiley, Chichester).

Gallasch, A. (1996). Unterstützungswerkzeug Simulation. In H.P. Wiendahl (ed.), *Erfolgsfaktor Logistikqualität* (Springer, Berlin), 107-118.

Graber, A., Ulrich, H., Schweizer, D. & Zimmerman, A. (1993). A Highly Interactive Discrete Event Simulator designed for Systems in Logistics. In A. Verbraeck & E. Kerckhoffs (eds.), *European Simulation Symposium Proceedings*, Delft.

Grote, G., Weik, S. & Wäfler, T. (1996). KOMPASS: Complementary allocation of production tasks in socio-technical systems. In S.A. Robertson (ed.), *Contemporary Ergonomics*, (Taylor & Francis, London), 306 - 311.

Günther, H-O. & Tempelmeier, H. (1997). *Produktion und Logistik*, 3rd ed., (Springer, Berlin).

Hacker, W. (1997). Improving engineering design - contributions of cognitive ergonomics, *Ergonomics*, vol. 40, no. 10, (Taylor & Francis, London), 1088-1096.

Lüthi, H.-J., Ulrich, H., & Dürig, W. (1996). Innovation by simulation using the example of an automated work-cell, *Central European Journal for Operations Research and Economics 2-3*, 135-154.

Malik, F. (1996). *Strategie des Managements komplexer Systeme*, 5th ed., (Haupt, Bern).

Moscoso, Ph., Wäfler, T. & Windischer, A. (1997). A modelling concept to support complementary design of dynamically complex systems. In E. Hollnagel *et al.* (eds.), *6th European conference on cognitive science approaches to process control*, (CNR, Rome), 235-240.

Pielok, T. (1995). Prozessketten-Management - Analyse und Modulation von Prozessketten mittels Logistic Function Deployment. In: A. Kuhn (ed.), *Unternehmenslogistik*, (Verlag Praxiswissen, Dortmund).

Ulich, E. (1994), *Arbeitspsychologie*, 3rd ed., (Verlag der Fachvereine, Zurich; Poeschel, Stuttgart).

Wiendahl, H.P. (1997). *Fertigungsregelung, logistische Beherschung von Fertigungsabläufen auf Basis des Trichtermodells*, (Hanser, Munich).

Wooldrige M. & Jennings, N.R., (1995). Agent Theories, Architectures, and Languages: a Survey. In M. Wooldrige & N.R. Jennings (eds.), *Intelligent Agents*, (Springer, Berlin), 1-22.

Hierarchical Simulation Models Improve Production-Accompanying-Simulation

Dipl.-Ing. W. E. Lulay, Prof. Dr.-Ing. G. Reinhart

Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb), Technische Universität München, Boltzmannstrasse 15, 85748 Garching bei München, Germany, lu@iwb.mw.tu-muenchen.de

Summary: The efficiency of decentralized production systems depends essentially on the organization of the production process. To feed the potentials of these production systems, a certain degree of partial autonomy is admitted. To not let local objectives dominate the whole production process, the decentralized units must be aligned to global objectives of the company. One approach is supporting and coordinating decision-taking by using simulation technology accompanying to the production process. Hierarchical simulation models facilitate the suitable configuration of simulation models and allow faster examination of larger production areas.

Problem definition

During the production process in decentralized production systems employees must take and transfer decisions permanently. On the one hand this concerns *short-term-decisions*, like the planning of job sequence order, organisation of machines or strategies for malfunction handling. On the other hand this affects *long-term-decisions* like the adaption of the product- and machine-spectrum (Figure 1).

Short term decisions

- **Job sequence order**
- **Malfunction handling**
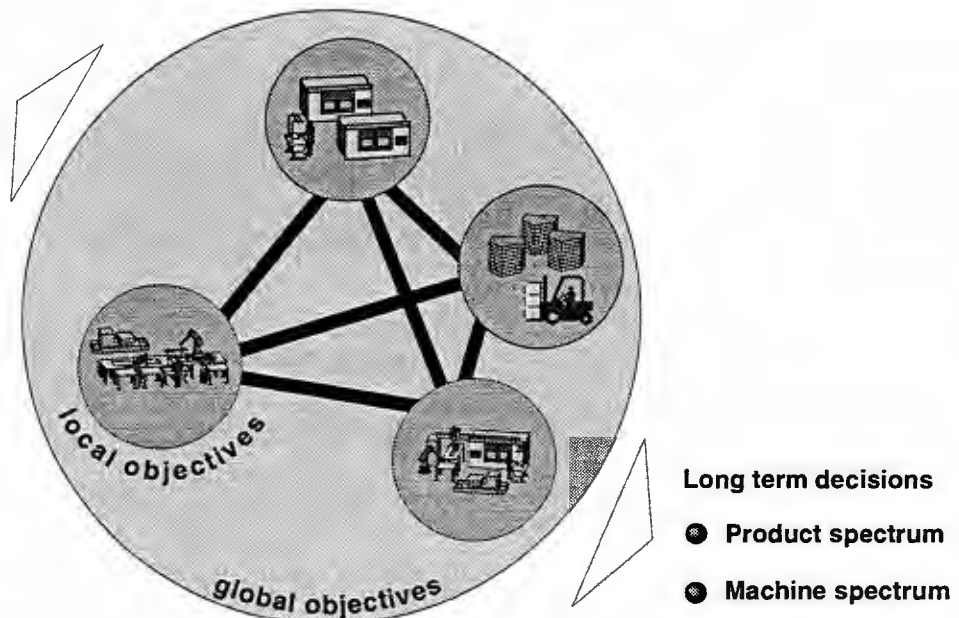
Long term decisions

- **Product spectrum**
- **Machine spectrum**

Figure 1: Local and global objectives in case of decision-taking

The quality of decisions determines substantially the efficiency of the whole production process. To be able to take decisions rapidly, a certain measure of partial autonomy is admitted to the organizational units. Above all the use of local knowledge via local problems is taken as profitable.

This partial autonomy, which should lead to a more dynamic order processing, contains a problem, too: the decentralized process of decision-taking is dominated by local objectives, while global objectives are often underestimated (a local objective is for example *the attaining of the optimal workload in the own organizational unit*, while a global objective is for example *attaining short times of delivery and a high schedule-adherency for all tasks*). Another problem are the hidden effects of token decisions - because even if local and global objectives are considered at the same time, according to the complex interactions the true effects often are not recognized.

Both aspects, *dominating local objectives* and *hidden effects* complicate the decision-taking-process or even lead to false decisions - the primarily planned characteristic of rapid and appropriate decisiontaking disappears.

Requirements for the support of the decision-taking-process

To guarantee the positiv characteristics of decentralized production systems and to support the local competence of employees, the decision-taking-process must be supported „in some way". An environment for experiments, in which the employees can analyse alternative decisions and find the best one would be ideal. The employees must be enabled, to formulate typical decisions easily, if possible just by pressing buttons. As a result, the quality of the decisions and their effects on local and global objectives should be shown. In this environment for experiments it should be possible to oppose local and global objectives and to solve conflicts between them by the use of simple coordination mechanisms.

Approach for the problem solving

To meet these requirements the *Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb)* develops an assistence-system, which gives the demanded decision-taking-support. The project, in which this approach is developed, belongs to the german research network *Forschungsverbund Simulationstechnik (ForSIM)*. The whole research network, as well as the project described here, is joined by a number of german enterprises.

Figure 2: Production-accompanying-simulation as an assistence-system for decision-support

The basis of the assistance-system is a simulation-system, in which the production to be examined is modelled - in this certain case this is the production area of the co-operating enterprises. The model gets actualized by interfaces to the systems for production planning and control, which the enterprises already have.

The employees can have themselves supported in decision-taking. They can examine the quality and the effects of decisions. The assistence-system checks the decisions and verifies possible effects. The result of this verification process is replied in form of logistic characteristic numbers, such as delivery time, adherency to schedules of jobs or the workload of machines.

On this basis employees can decide whether the decision has to be modified or whether it can be implemented.

Hybrid, hierarchical simulation models

Representation of available modelling techniques

For the production-accompanying use of simulation, special demands on the design of the simulation model are made. On the one hand comprehensive models are required, which cover extensive production areas and permit detailed questions. On the other hand the models should be rapid enough, to be able to use the responses actually accompanying to the production process.

These competitive requests (global, detailed, rapid) cannot be met, if only one of the subsequent described modelling techniques is used.

Discrete event modelling maps the characterstic and the behaviour of objects, for instance real-world objects like machines, orders, human beeings and so on. An example for a *characteristic* of an object can be *the availability of a machine*, while its *behavior* can be the *ability of processing orders*. At discret points in time the so modelled objects generate events thus affecting other objects. The behavior of the entire model results from the behavior of its components and can be evaluated after the simulation run.
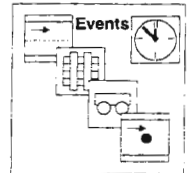
Queues normally arise on operator terminals, where demands for operation (for example the fulfillment of the customer preferences or the processing of an order) exceed the oparating offer. *Queue model theory* specifies the dynamic behavior of systems by describing the distribution functions of the *access rate*, of the *operating rate* and by determing the *system size*.
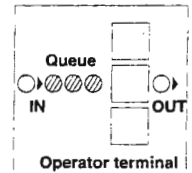
Operating curves describe the static characteristics of production systems. They point out the relationship between logistic characteristic numbers like *workload, delivery time, schedule-adherency* and the *utilization of machines*. They can be detected empirically or by simulation.
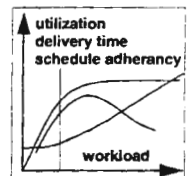
Regression models represent causes and effects of several criterias. It is attempted to analytically describe the interconnection between the mostly empirically determined data series.
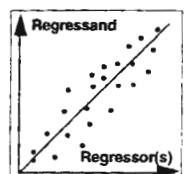
The discrete event modelling technique described above is useful for detailled mapping of the structure and the dynamic of production systems. Particularly if large production areas are concerned, discrete event models become very complex and intransparent. Then they require simulation times, which are too long for the production-accompanying use. More abstract modelling techniques, such as queue models, operating curves or regression models, shorten the needed simulaton times. The problem here is, that the examinable questions are not detailed enough for supporting the decision-taking and co-ordination-process of decentralized production systems efficiently.

452

Integration of available modelling techniques to hybrid, hierarchical simulation models

In the approach presented here, several modelling techniques are integrated into a hierarchical simulation model (Figure 3).
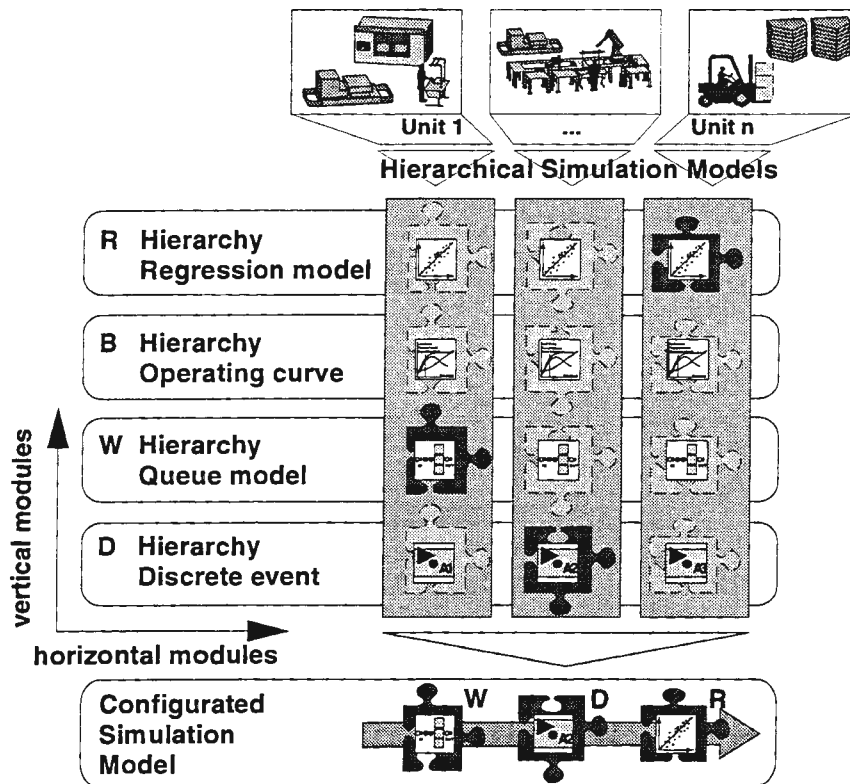
Figure 3: Hierarchical simulation models whith horizontal and vertical modules

The basic structure of these hierarchical models are horizontal and vertical modules.

The *horizontal modules* represent each production-unit by an own partial model. This facilitates the modelling and improves interpretation of the simulation results.

The *vertical modules* in each partial model represent different abstract hierarchical levels. The differing abstraction is reached by the use of the different modelling techniques described above.

The most detailled level is modelled by discrete event techniques - as said, this allows the most detailled questions but requires the longest simulation times. Queue models are used for a more abstract modelling of dynamic interconnections of certain charteristics (access rate, operating rate, system size). Operation curves are utilized for a static view on the interconnection between the logistical characterstic numbers. Regression models are appointed, when interconnections between other data are desired to be described.

The decision, which hierarchical level is suitable, is taken by the user depending on the question to be examined. Parameters for this decision are the recommended accuracy of the simulation results, the available time or the logical distance of the production area, which has to be covered. For areas at a short logical distance (e.g. the own organisational unit) a detailled hierarchical level has to be used (e.g. the discrete event level). Areas at a long logical distance (e.g. storage facilities) can be modelled more abstract (e.g. by the use of a queue model).

Consistency between partial models and compatibility between hierarchical levels

In order that the configuration of the hierarchical simulation models actually functions under run-time, consistency between the several hierarchical levels and compatibility between the several partial models are necessary.

Consistency between levels means, that model variations have to be mentioned in each level according to its specific modelling possibilities. Especially model levels with lower accuracy have to be adapted to alterations in model levels with higher accuracy: for example alterations in the discrete event level (or even in reality) must be considered in the queue-level etc.

Compatibility between partial models means, that the configuration of the entire simulation model has to be a process of „sticking together" several partial models, no matter which hierarchical levels in the different partial models are activated at that time.
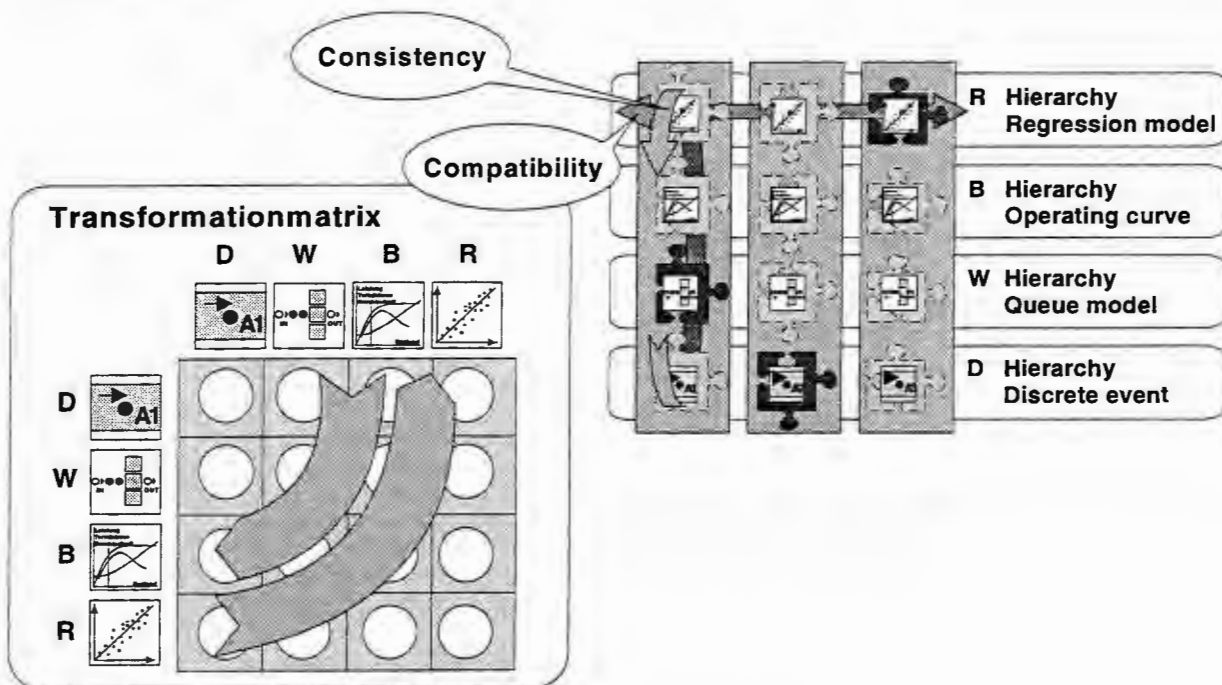
Figure 4: Transformation matrix for consistency and compatibilty[1]

Both, consistency and compatibility, are implemented by *transformation matrices,* which are assigned to each partial model. Every transformation matrix contains several *transformation functions* wich translate the behaviour of one hierarchical level into another.

---

[1] The characters D, W, B, R were derived from the german terms:

Character	German Term	English translation used in this paper
D	Diskrete Modellierung	discrete event modelling
W	Warteschlangenmodell	queue model
B	Betriebskennlinie	operating curve
R	Regressionsmodell	regression model

Figure 5 shows in exemplary fashion the transformation of discrete event behaviour (D) into the behaviour of the other modelling techniques (W, B, R). The transformation function D→W (Figure 5, top) monitors the discrete event level and detects the distribution function and its parameters of the access rate, the operating rate and determines the size of the system. The transformation function D→B (Figure 5, middle) monitors the discrete event level and approximates the operating curve out of characteristic numbers as the maximum capacity, the minimum delivery time and the limiting workload. The transformation function D→R (Figure 5, bottom) monitors the discrete event level and calculates the regression model of choosen events - as an extension to the queue model or the operating curve the events to be monitored can be choosen totaly free.
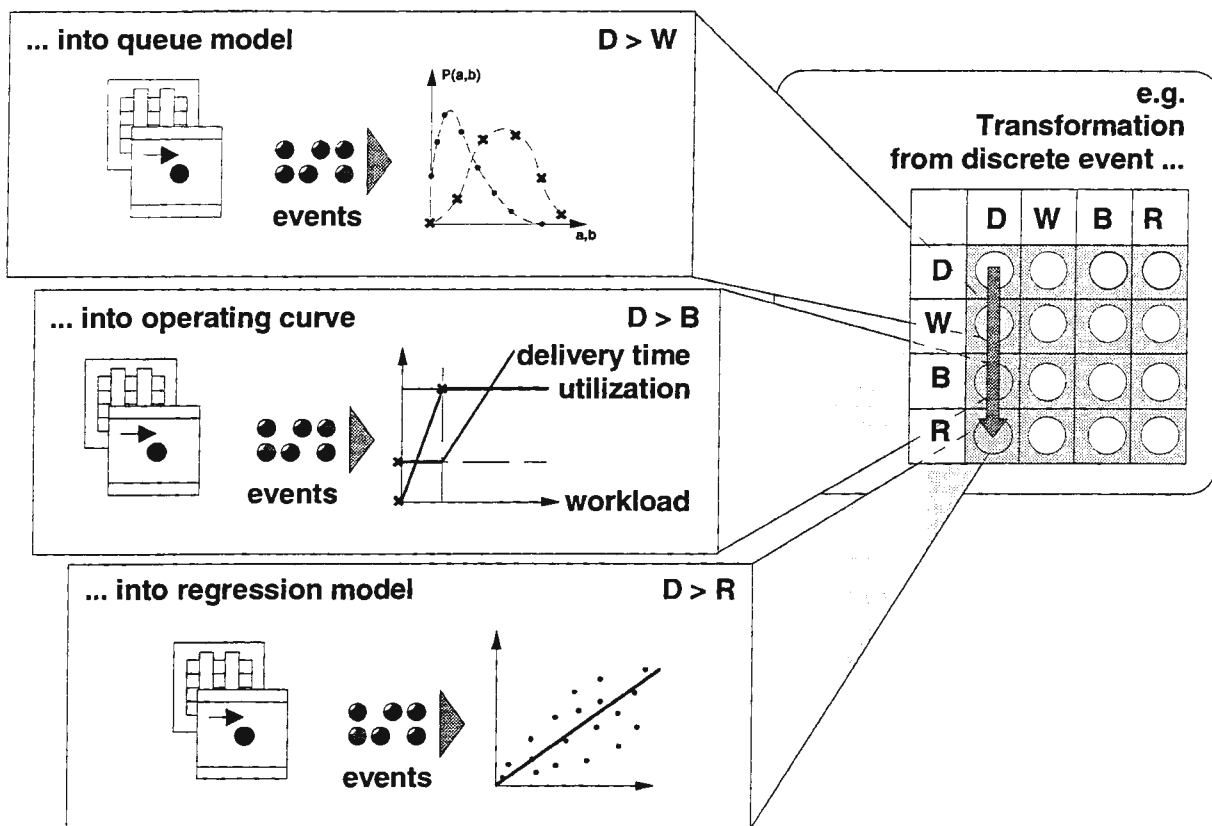
Figure 5: Consistency between hierarchical levels within partial models

Figure 6 shows in exemplary fashion the transformation of the queue model (W), the operating curve model (B) and the regression model (R) into the discrete event level (D). The transformation function W→D generates events (such as creation of orders) - the basis therefore are the distribution functions of access rate and operating rate). The transformation function B→D generates events too, but bases on the relationship between workload, utilization, delivery time and schedule adherency. So does the transformation function R→D, depending on the relation of the freely choosen numbers of the regression model.
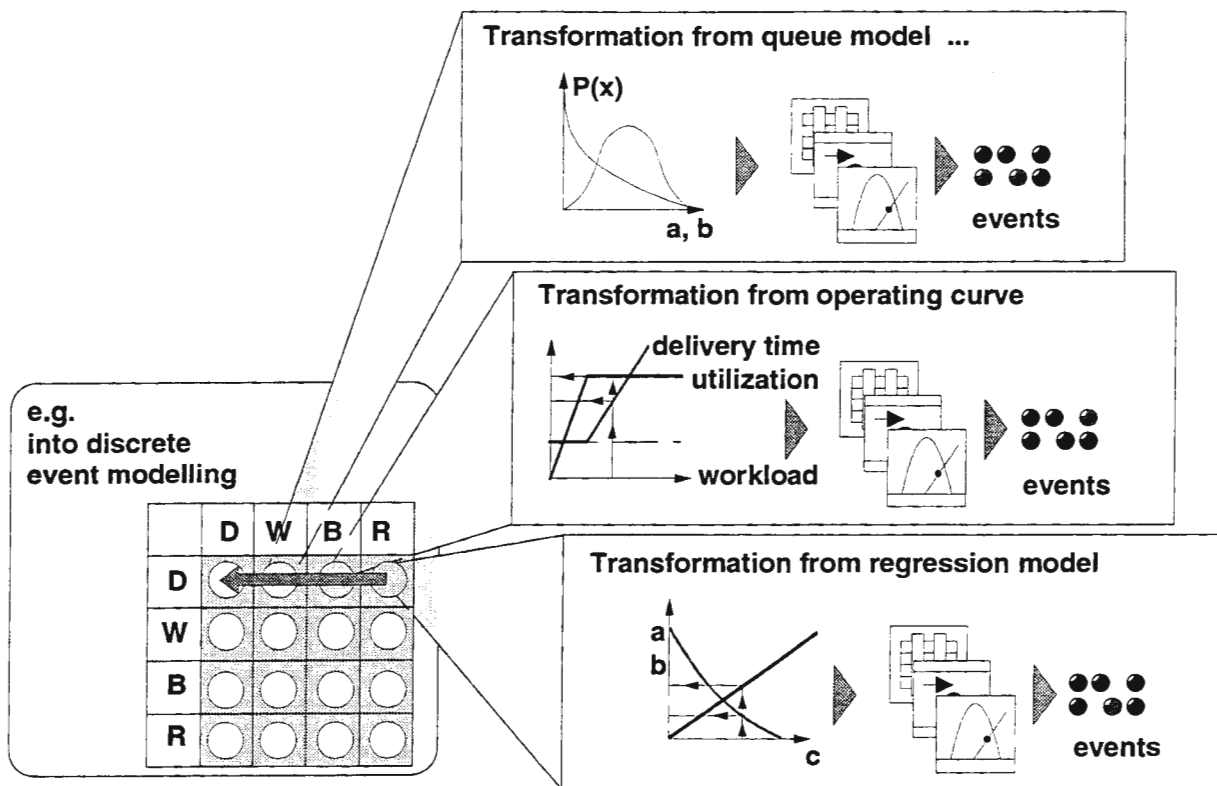
Figure 6: Compatiblity between different modelling techniques of different partial models

Of course there is a loss of accuracy, especially when detailled levels are transformed into more abstract levels and therefore less detailed views. One instead receives levels of discription, which produce the wanted simulation results more rapidly or on the basis of less detailed master data. It is important to mention, that the level of abstraction of the entire simulation model can be variated by the user individually and according to his needs. Thus the user is able to choose the qualities of the used modelling technique very precisely:

- if there is enough time for the simulation experiment, the period for the forecast is short, the master data is accurate, the production unit to be modelled is at a short logical distance and the results have to be very explicit, more detailled modelling levels will be choosen

- if there is only little time, the period for the forecast is long, the master data are not very precise, the production unit to be modelled is at a long logical distance and if only basic statements are necessary, more abstract levels are sufficient

Summary and perspective

The assistence system for production-accompanying-simulation described in this article provides an environment for experiments to production companies. In this environment the quality and the effects of decisions can be analyzed and modified, until they can be implemented in the real production area. The supporting of the process of decision taking bases on a dynamic and therefore realistic model - the real production process does not get disturbed. The global view as well as the local view on the supplied characteristic logistic numbers facilitate co-ordination of decentralized production units onto a common enterprise-wide goal.

The choosen approach of *hierarchical simulation models* enables to meet the competitive requests of *comprehensive* and *detailled* and *rapid simulation models*. The simulation models can be configurated under run-time and thus adapted to the momentary needs while decision taking.

During the modelling process additional efforts are necessary of course, because different abstract and parallel valid levels have to be created. This additional effort is partially compensated by the developed modelling rules. The current works attempt to reduce the necessary efforts and, where possible, to partially automate the modelling process.

An important focus at the moment is the question, how the accuracy of the transformation functions can be increased. Another important focus is laid on user interaction. Especially in production areas the number of simulation experts is rather low - therefore the user interaction must be orientated on the needs of simulation laymen. In particular the system-complexity, as it is described in form of hybrid, hierarchical models, must be concealed to the user. For this an interaction concept was developed and is now implemented into a prototype to be used in the production area of the co-operating enterprises.

Acknowledgements

References

[1]   Banks, J.; Carson, J. S.; Nelson, B. L.: Discrete-Event System Simulation. Prentice Hall, New Jersey 1996

[2]   Bosch, K.: Statistik Taschenbuch. Oldenbourg, Wien, 1993

[3]   Law, A. M.; Kelton, W. D.: Simulation, Modelling and Analysis. McGraw-Hill, New York 1991

[4]   Martin, C.: Verfahren und Systeme zur simulationsbasierten Produktionsregelung bei komplexen Produktionsstrukturen. In: Kampe, G., Zeitz, M. (Hrsg.) 9. Symposium Simulationstechnik, Stuttgart 1994

[5]   Reinhart, G.; Decker, F.; Heitmann, K.: Möglichkeiten zur Integration der Simulation in das betriebliche Umfeld. Maschinenmarkt 101 (1995) 36, S. 48-53, Würzburg

[6]   Reinhart, G.; Lulay, W.: Koordination dezentraler Produktionsstrukturen durch betriebsbegleitende Simulation. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb. 1-2 (1998) , Carl Hanser München

[7]   Wiechert, W.: Modellkonsistenz und Konfliktlösung bei Hybriden Modellansätzen. ASIM 97 - 11. Symposium Simulationstechnik, Dortmund 1997

[8]   Wiendahl, H.-P.: Belastungsorientierte Fertigungssteuerung - Grundlagen, Verfahrensaufbau, Realisierung. Carl Hanser München 1987

[9]   Zetlmayer, H.: Verfahren zur simulationsgestützten Produktionsregelung in der Einzel- und Kleinserienproduktion (iwb Forschungsberichte Nr. 74), Springer Berlin 1994

[10] Zimmermann, H.-J.: Operations Research - Methoden und Modelle. Vieweg Braunschweig 1992

PROCESS NETWORKS ENGINEERING:
CONTROL-LOOP-BASED MODELING
OF DECENTRALIZED FACTORIES

Horst-Artur Crostack[1], Markus Becker[2], Michael Saal[3]

[1] Universität Dortmund, Lehrstuhl für Qualitätswesen, 44221 Dortmund / Germany, tmielke@rif.fuedo.de
[2] Universität Dortmund, Lehrstuhl für Qualitätswesen, 44221 Dortmund / Germany, mbecker@rif.fuedo.de
[3] RIF e. V., Joseph-von-Fraunhofer-Str. 20, 44227 Dortmund / Germany, msaal@rif.fuedo.de

Abstract

Competitive advantages of manufacturing strategies can only be achieved by designing corporate structures and processes purposefully. Instead of improving production methods individually, which in any case are usually highly developed, they should be linked together into an efficient complete process. Material flow, manufacturing, and inspection processes must not be treated separately, but have to be seen as part of a complex system. Since information flow is a crucial factor for process performance, it must not be neglected in production models. Information processes are not only additional processes in a system; they also interact with other production processes. This can be described as a control loop. By means of modeling and simulation based on a control loop element, it is possible to analyze, systematically develop, verify and implement structures and processes of decentralized production systems.

The Task: Modeling Decentralized Factories

Holonic or Intelligent Manufacturing Systems, Modular Organization, Fractal Factories, Virtual Companies, Lean Manufacturing and other concepts of decentralized production have one basic concept in common: Largely autonomous organizational units manufacture products and services and exchange them via well-defined interfaces in a network (fig. 1).
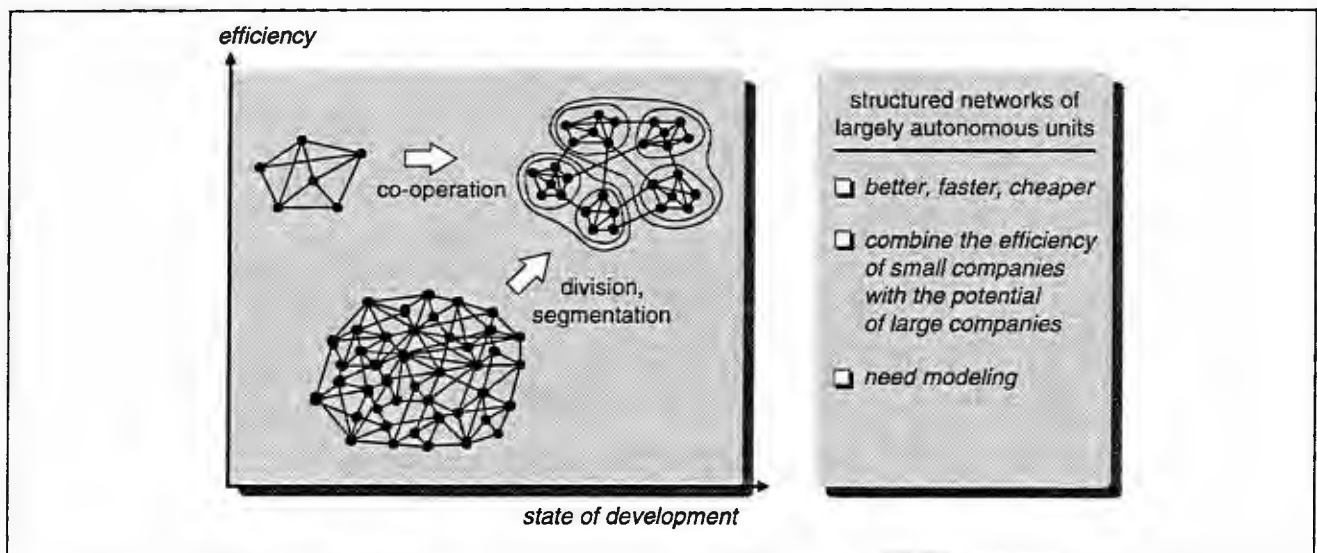
Fig. 1: Decentralized factories

Resources required for production are available in each unit and can be utilized efficiently and effectively due to self-controlling mechanisms *within* the subsystem. Interfaces *between* organizational units are designed to enable an exchange of products and services without frictional loss. Thus, the individual units themselves, as well as the whole system are particularly productive.

Concepts of decentralized production apply to a broad range of organizations. In order to fit the specific case, basic concepts need to be tailored. Starting with an analysis of customers' needs, market structures, core competencies and corporate objectives, management must develop a *vision* in order to carry out the initial idea for reorganization. On this basis, a framework of requirements for technical, organizational and personnel-related process conditions has to be derived. This is not yet a specific applicable solution. But how can a company find one? Whereas there are impressive case studies in management literature, the basic concepts themselves are usually quite general and give no specific *guidelines for implementation*. Therefore, development and implementation of tailored solutions is up to management and experts of process modeling.

The Challenge of Quality Process Modeling

World-class organizations are process-oriented, i. e., their structure is focused on the support of corporate production of products and services, as well as conducting process assessments regularly. On the other hand, processes in common enterprises have never been planned systematically, or they have changed over time, so that structure and process are not suitable anymore.

Depending on specific conditions and objectives, there are numerous degrees of freedom for the quality oriented design of production systems and processes, regarding e. g.:
- the *origin* of quality (single steps and links in the value chain)
- the *proof* of quality (inspection strategies within the process) or
- the *control* of quality (management and coordination of processes)

Concepts of decentralized production make use of a wide range of instruments from various scientific disciplines. But they still need to establish a general method of their own. A system for the integration of indirect tasks is notably missing, namely, an answer to: Which tasks should be executed centrally, which in decentralized units?

Moreover, state-of-the-art concepts of process design usually consider material flow (or work flow), but neglect important cross impacts: the network of manufacturing processes and aspects of inspection are treated separately. For instance, product related data (CAD/CAM/CAQ) and order related data (PPC) are usually treated separately. This means that available information and potential know-how is not utilized. Models of production like material flow simulation or applications of PPC, if not based upon a quality model, also neglect these interrelations. An insufficient model, however, causes unsatisfactory results of planning and trouble in production (fig. 2):

- Product characteristics can be produced in a *single* activity, in a *series* of activities (sub-process), or by *combining* the results of two or more activities. Whereas connections in material and work flow are obvious, there are many interrelations with less evidence: Activities can influence each other or have effects on remote sub-processes. These hidden impacts are a major cause for trouble in production processes.
- Pure (material flow) simulation and most PPC systems ignore measuring equipment. Time consuming inspection in test laboratories or *bottleneck* capacities of special measuring equipment like 3D measuring machines can cause delay to processes.
- Deviations of quality characteristics can cause trouble in current production programs: *Rework* is likely to occupy machines needed for other orders. Eventually, *substitute* orders have to be written and considered in PPC. This affects not only schedules: If lot sizes are lower than usual, inspection plans must be updated, too.
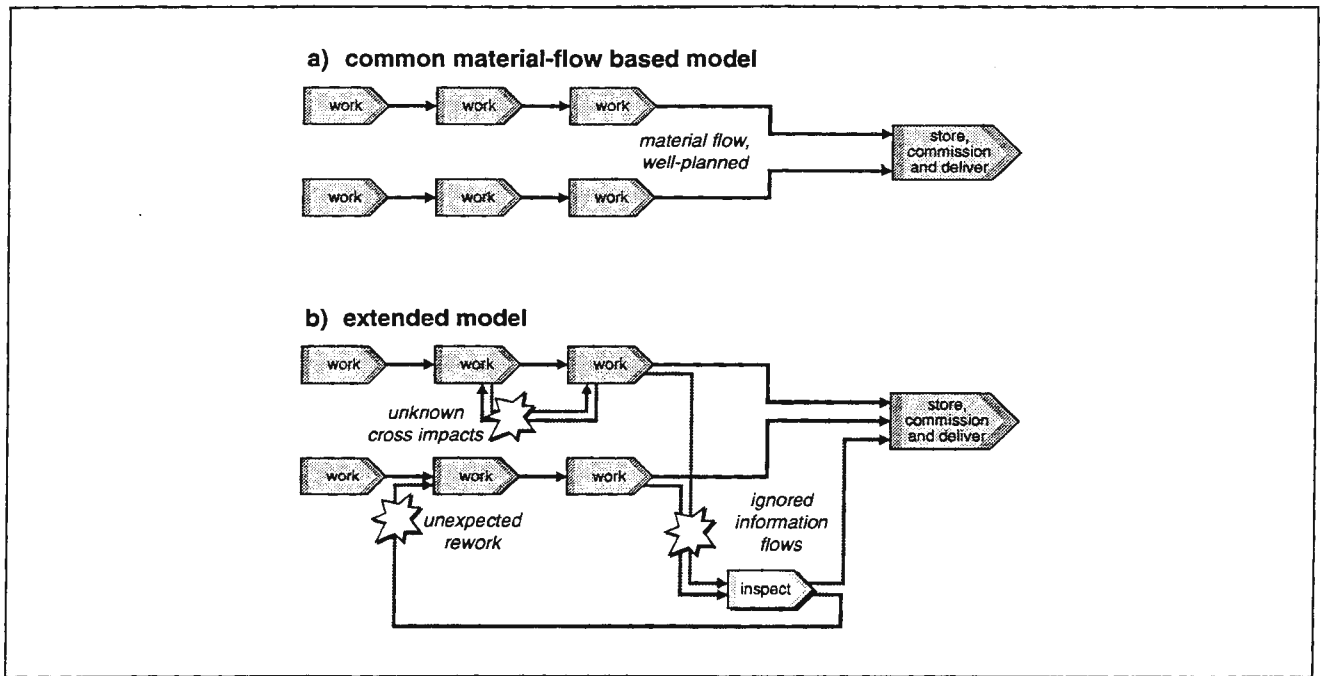
Fig. 2: Extended models of production processes have to take quality aspects into consideration

Similar interrelations exist in every organization. Due to organizational autonomy and spatial separation in decentralized production systems, their effects are potentially stronger than in common organizations. As a consequence, many tasks of planning and simulation require a more differentiated, but still manageable process model.

Usually, production planning models have to consider a system of objectives, covering aspects of time, costs, and quality. In decentralized production systems, these objectives have both *internal* aspects (regarding production of products and services) and *external* aspects of exchange between sub-systems or customers and suppliers respectively (table 1).

Table 1: Performance objectives in decentralized production systems (examples)

	internal: performance	external: competitiveness
speed	cycle time, process time	delivery lead time
dependability	reliable operation	on time delivery to customers
costs	total productivity	price, costs of transaction
quality	error-free processes, scrap, rework	quality of product, delivery, use
quantities	stock, work in progress, throughput	output, delivery lot size
flexibility	lot size, use of capacity	product range (variants), delivery

The Concept of Control-Loop-Based Process Modeling (c/pm)

Production structures can be regarded as a system, that is a defined set of elements with relations. Therefore it is irrelevant for the mere *definition* of units, whether they are intra-corporate or inter-company decentralized production systems. The real *differences* are found in special requirements for logistics, which are obvious due to the distance between cooperating plants, and also in the basis for cooperation, which is ruled by market mechanisms on the one hand and by organizational structures of power on the other.

Relations between elements make up a process. Its behavior can be described in terms of cybernetics, whose elementary basis is a control loop: After executing an activity, the result is being checked and compared to the set point in order to affect the process purposefully. The categories „*execute - check - compare - affect*" can describe every activity in an organization. Thus, „system", „process" and „control loop" make up a simple universal model, which can be used for the description, analysis and design of corporate processes. When applied to decentralized production systems, four different kinds of control loop can be distinguished (fig. 3):
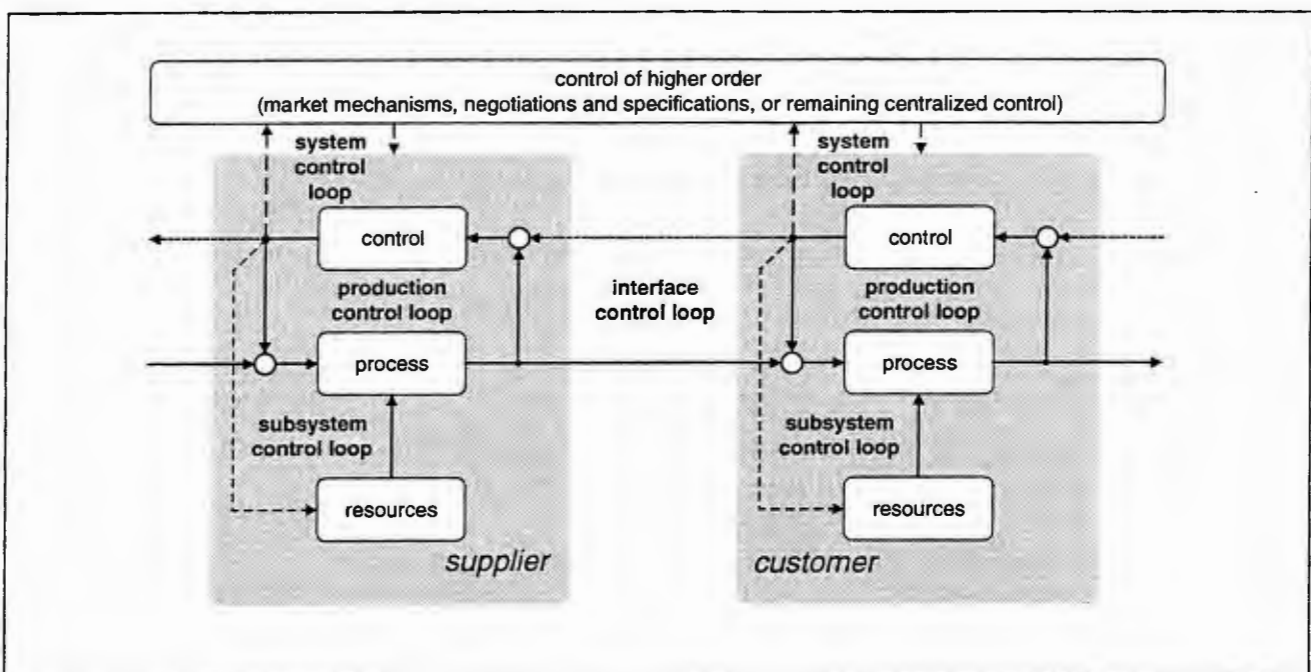
Fig. 3: General control loop model of decentralized production (simplified)

- In *production*, manufacturing process data or product characteristics are being compared to the desired value in order to affect the production process. Production control loops are, for example, process control in the narrow (technological) sense, inspection or statistical process control (SPC).

- For each exchange of products (material, energy, information) or services, the recipient can give feedback to the supplier regarding the condition of products or their delivery. Delivery and feedback make up a control loop in a figurative sense. Interface control loops are, for example, raw materials delivery, acceptance inspection and corresponding payment or even complaints.

- Experience in production often leads to measures of improvement. They usually affect the process by changes of *resources* such as machines or devices, by the design of work systems (ergonomics) or by new general know-how, like procedures, work schedules, recipes, or qualification of employees. Autonomous organizational units can carry out improvements on their own (subsystem control loops), for example as part of a continuous improvement process. This leads to higher productivity, shorter cycle time, less stock and work in progress, less scrap and rework.

- Even the *structure* of decentralized production systems and the *cooperation* of organizational units is being controlled. Negotiating contracts, strategically selecting and assessing suppliers, jointly developing new products and processes or solving problems in mutual task forces, exchanging employees and information make up what has been defined as system control loops.

Production and interface control loops represent *everyday work*, whereas subsystem and system control loops describe processes of *change* within decentralized production systems. Thus, organizational learning, learning from the customer and learning in market structures can be regarded as control loops in a more figurative sense. To summarize, the four general kinds of control loop can describe all business processes. They are the basic constituents for modeling complex structures and processes of decentralized production systems as hierarchically nested control loops.

Application of the Concept: Results

As a *modular* and *hierarchical design method*, c/pm can be used to model different levels of production - cooperation of various companies and plants, single units of the production system, single orders (material flow and process level) and the level of machines and single activities. Since the basic concept of activity and control is similar, it is possible to link various hierarchical levels together in one general model. Thus, production and exchange of products can be described with all aspects of cooperation like interrelations (including the hidden ones) and interfaces.

The c/pm model can be applied in different phases of *organizational process design*: from the first idea, the concept, specification and implementation phase to everyday work in business processes. It supports the main areas of process planning, control and improvement. In particular, this facilitates not only the analysis of sophisticated cause-and-effect mechanisms, but also the prediction of how changes in manufacturing or inspection processes lead to better, easier and more robust procedures, as well as what their effect is on quality, time, and costs (fig. 4). Based on c/pm, simulation studies start from a better basis, or vice versa, the c/pm solution of process organization can be further refined by simulation.

In industrial projects, c/pm is being used for verifying and validating *concepts* of decentralized production systems. Since the simple model makes complex systems easy to understand, it supports implementation and management of new organizational structures. Moreover, c/pm allows to develop *specific solutions* as well as general *rules for the design* of decentralized factories.

So far, c/pm has been applied in the metal-processing, mechanical engineering and chemical industries for various problems:
- Reengineering administrative processes like order processing;
- Reorganization of logistics and value chains in modular production;
- Improvement of incoming goods processes and reliability of assembly processes;
- Analysis of manufacturing processes and organizational design of autonomous production cells;
- Development and simulation of inspection strategies (manufacturing, in-process and final inspection).

Improvements achieved are within the range known from re-engineering-projects. The most striking results are process improvements: Typically, cycle time could be cut by up to 60 %. Along with process precision, dependability has been improved too (roughly 20 % more on time deliveries in the process and to the customer). Less machining or process time and better schedules lead to an increase in productivity (about 25 %) and to a reduction of stock and work in progress, but at a higher availability. A reduction of non-control and multiple control prevents processes from trouble. This leads to higher product quality (up to 80 % less scrap and rework) and employees' motivation and satisfaction are better. All in all, the essential benefit of the application is both an increase in flexibility and a significant reduction of costs.

462

a) control-loop analysis of the old process (simplified)

(buying dept.)   (supplier)   (incoming goods)   (production)

- time consuming orders
- no supplier assessments
- no improvement

- stock, WIP
- mess
- time consuming

- faulty materials
- missing mat., WIP
- defects

b) control-loop design of the new process (simplified)

☐ production materials:
  inspected by production
☐ materials from subsidiaries:
  not inspected again
☐ special samples:
  inspected by QA

strategic buying dept.: general contracts and supplier assessment

(buying dept.)   (supplier)   (incoming goods)   (production)

Fig. 4: Application of the c/pm concept (example: incoming goods process)

Along with process improvements, *technological improvements* could be achieved. Manufacturing and inspection activities have been re-arranged in order to be executed at a more suitable position within the whole process context. Hence, processes are more robust and productive. Most processes can even do with less inspections. In distinct, closed control loops, information is available where needed. This develops know-how in all hierarchical levels, from everyday work in the workshop to production controlling and strategic management in the cooperation of plants.

Other *organizational improvements* result from the mere design of control loops (fig. 5): It is possible to derive criteria for the integration of planning and administrative tasks in production areas, to find process structures systematically, determine degrees of autonomy and establish mechanisms of coordination between organizational units.

A first prognosis shows enormous potentials for further, more *general improvement*. Unlike ad hoc solutions, the system of c/pm purposefully promotes completely new approaches of process design, if processes are not treated as time consuming black box elements. Conclusively, to consider interactions of process and control loop structures is a major prerequisite for the design of decentralized production systems, the prevention of trouble and the creative utilization of organizational complexity.
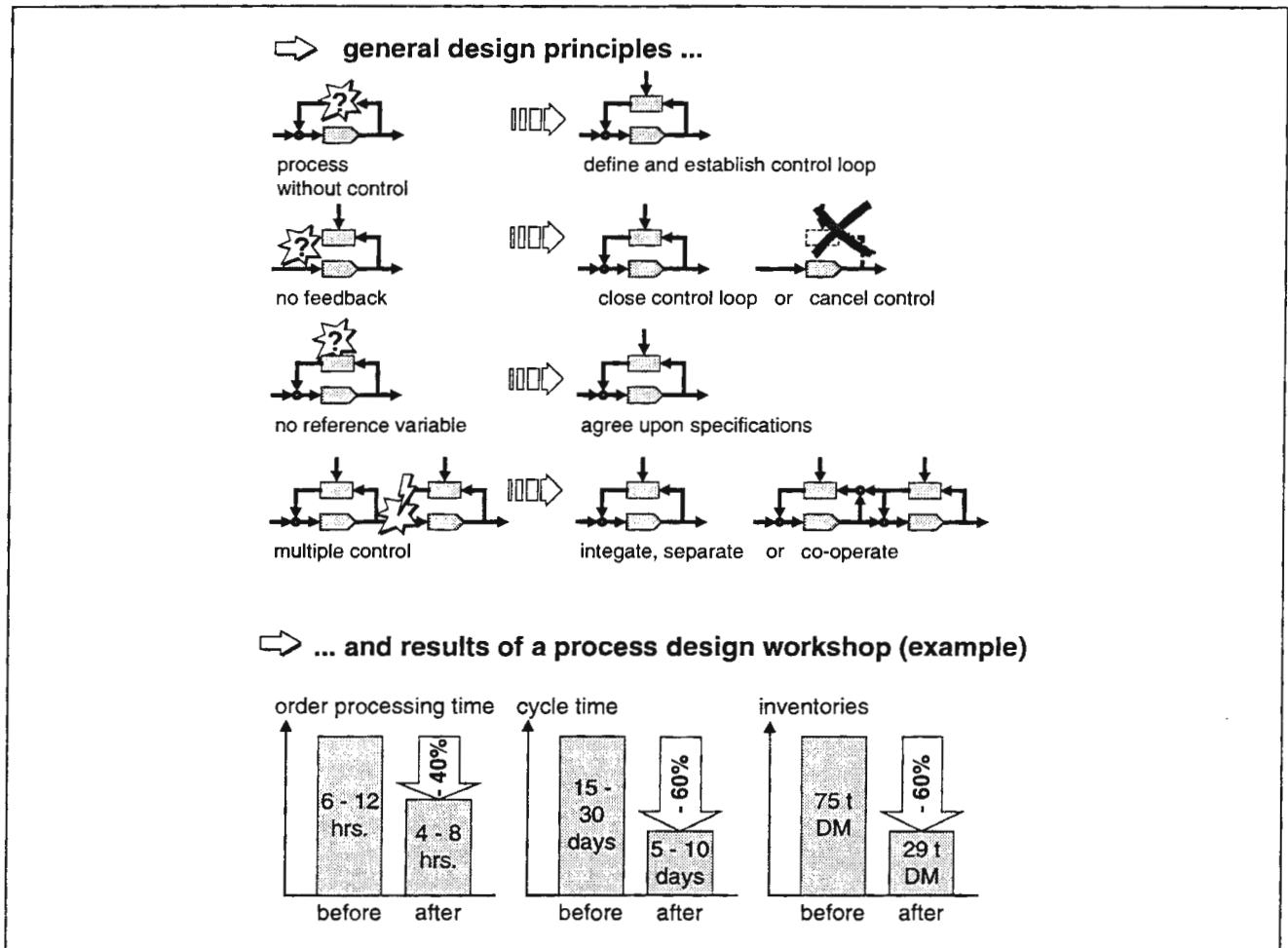
Fig. 5: Design principles and results of a process design workshop (example)

Conclusion

Process modeling based on control loops (c/pm) enhances the scope of strategic management by deriving distinct solutions out of general concepts since it is systematically developing and utilizing manageable processes. Modular modeling and distinct use of a wide rage of different methods do not only allow the systematic design of innovative production systems, but also the development of new, powerful PPC, CAQ and simulation software.

A more complex model than the usually considered, pure material flow model is an important prerequisite for the design and management of real-life complexity. Hierarchical and modular model structures support a comprehensive analysis and hence a thorough understanding of hidden interaction in complex systems. Moreover, this allows to systematically identify possibilities for improvement and innovation. Thus, it is possible to both reduce complexity and utilize its benefits. All in all, significant competitive advantage can be achieved.

Results of research and industrial project experiences in decentralized production systems are summed up in the „Dortmund Q" model (fig. 6). It combines advantages of control-loop-based process modeling (reduction, management and utilization of complexity) with the systematic implementation method, which allows the transfer of research results into real-life processes.
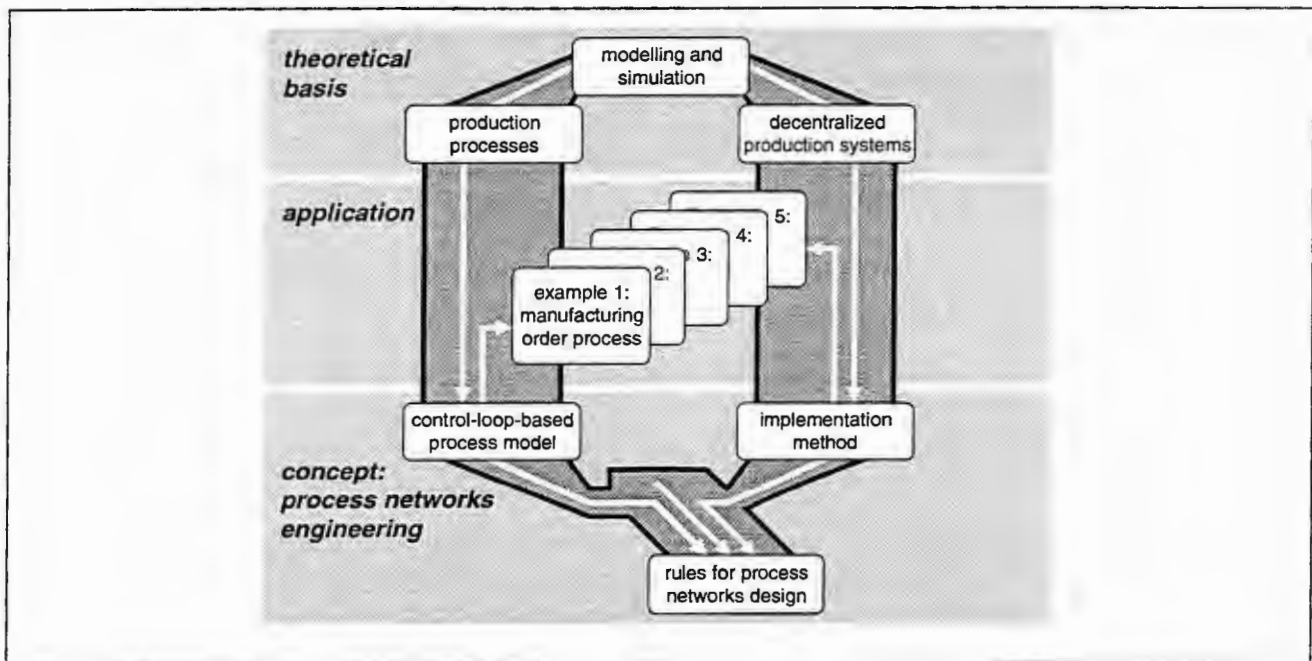
Fig. 6: Constituents of the „Dortmund Q"

Acknowledgment

References

Becker, M., Crostack, H.-A. and M. Saal: Process Networks Engineering: A Model-Based Approach to Process Design and Improvement in Decentralized Manufacturing Systems, in: 31st CIRP International Seminar on Manufacturing Systems, University of California, Berkeley 1998 (forthcoming)

Becker, M. and Saal, M.: Prozesse in dezentralen Fabrikstrukturen, in: Kuhn, A., Wenzel, S. (Ed.s): Simulationstechnik, 11. Symposium in Dortmund, November1997. Braunschweig/Wiesbaden: Vieweg 1997, pp. 397 - 403

Becker, M.: Process Quality Management Between ISO 9000 and Reengineering, in: EFQM (Ed.): 12 Fresh Views on TQM. Brussels: EFQM 1996, pp. 42 - 51

Becker, M., Crostack, H.-A. and M. Saal.: Process Networks Engineering: Modeling Decentraliazed Factories, in: McDonald, D., McAleer, M. (Eds.): MODSIM 97 International Congress on Modelling and Simulation, 8. - 11. December 1997, University of Tasmania, Hobart, Vol. 4, pp. 1498 - 1502

Crostack, H.-A., Becker, M. and M. Nürnberg: Modellierung und Simulation im Qualitätsmanagement, in ASIM (Ed.): 8. ASIM-Fachtagung Simulation in Produktion und Logistik, 16. - 17. Februar 1998 Berlin (forthcoming)

Crostack, H.-A.; Heinz, K.; Nürnberg, M.; Nusswald, M. and A. Schwerdtfeger: Bewertung von Prüfstrategien, in: Qualität und Zuverlässigkeit, Vol. 41 (1996), No. 11, pp. 1307 - 1312

Daenzer, W. F. and F. Huber (Eds.): Systems Engineering: Methodik und Praxis. 8. Ed., Zürich: Industrielle Organisation 1994

Malik, F.: Strategie des Managements komplexer Systeme: Ein Beitrag zur Management-Kybernetik evolutionärer Systeme. 5. Ed., Bern; Stuttgart: Haupt 1996

Using Simulation for Evaluation of Re-engineering Actions

J-M Lehtonen

Helsinki University of Technology, TAI Research Centre

Po. Box 9555, 02015 Espoo, Finland

Tel: + 358 9 451 1, fax: + 358 9 451 3665

email: jmlehto@tuta.hut.fi

Abstract

The research objective of this paper is to experiment the applicability of discrete event simulation model to evaluate the re-engineering potential in a case distribution problem. The use of simulation methodology proved to be able to assess the re-engineering potential by producing quantifiable results for the performance measures. Simulation model as such does not give any indication of the possible changes in order to improve desired logistics system performance measures. Its function is to assess the improvement potential of re-engineering actions derived elsewhere.

Keywords: Simulation, modeling, logistics

1 SIMULATION AS A RESEARCH METHOD IN LOGISTICS

Today the use of discrete event simulation as a research method is commonplace in logistics research. In fact, the importance of simulation can be described by the viewpoint that there are three pillars in scientific and technology research; analytic methods, experimental methods and numerical simulation (Strandhagen, 1994).

According to Bowersox (1986), the main advantage of discrete event simulation in logistics is that it incorporates the impact of time into performance evaluation. Hoover and Perry (1989) view simulation as a powerful problem solving method. According to Mott and Tumay (1992), simulation is a thorough and economic way to express and evaluate ideas and alternatives before implementation. Simulation, however, will provide merely the best answer of the solutions tried and thus limited in its ability find the "best" answer (Menzer, 1989).

The popularity of simulation can partly be explained also by the following advantages of simulation in logistics research:

- In comparison to analytical models, it is flexible. You can include details, which are impossible with analytical models (Hoover and Perry, 1989). Even in simple systems analytical modeling of queuing disciplines and priority rules is extremely difficult (Johnsson, 1992)

- In comparison to the real system, the researcher has control over the other variables (Johnsson, 1992). In fact, simulation could be viewed as a test laboratory of logistics research.

- Non-existent systems can also be studied (Hoover and Perry, 1989).

2   SIMULATION PROJECT MODELS

A few authors have attempted to give guidelines for carrying out a simulation modeling project. Strandhagen (1994) has a summary of the simulation project models and he concludes that for many purposes it is convenient to distinguish only three major tasks:

• feeding information into a system

• running the model

• extracting information from the system
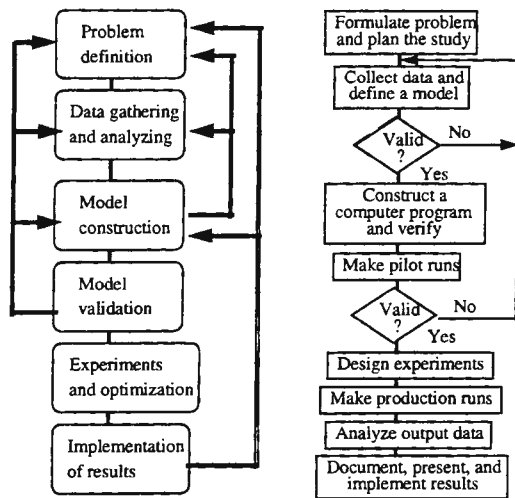
Figure 1 Simulation project models, left: Hoover and Perry, right: Law and Kelton.

Figure 1 shows two simulation project models by Hoover and Perry (1989) and Law and Kelton (1991). Both project models are almost in perfect agreement on how to perform a simulation study, although Hoover and Perry view the project even more iterative than Law and Kelton. Also Law and Kelton add finer detail in some of the phases than Hoover and Perry. Also Banks & Carlson (1984) give similar project guidelines. In the following, the major stages in simulation project models are discussed.

Problem definition

The problem definition is very important and as Hoover and Perry (1989) put it, one is not likely to find good solutions to ill defined problems. Simulation textbooks and literature that cover simulation project, however, give only very vague guidelines concerning problem definition. For example, according to Law and Kelton (1991) a model should contain only **enough** detail to capture the **essence** of the systems and Mitrani´s (1982) advises it is necessary to abstract from the real system **all** those components that are considered **important** enough for inclusion in the model.

Model definition and data gathering and analyzing

According to Hoover and Perry (1989), an important part of model definition is to define system inputs, that can either be decision variables or given variables and outputs. Hoower and Perry (1989) advise to first make a conceptual model by either drawing a flowchart or by using state change approach before implementing it. Defining the model is almost as vaguely covered as problem definition. According to Robinson (1994) there is a tendency to want to model everything without stopping to consider exactly what is **necessary**. He advises to include the minimum amount of detail to achieve project's objectives. Law and Kelton (1991) advice that one should not have more detail in a model than is necessary to address the issues of interest.

Validation

A valid model corresponds to the real system. As Strandhagen (1994) point out, no model is ever completely valid corresponding perfectly to the real system. One approach to validating a simulation model is to see whether the simulation model can produce similar results as the real system has produced (Hoover and Perry, 1989). Law and Kelton (1991) call this the correlated inspection approach. According to Naylor et al. (1966) the ultimate test of a simulation model is the degree of accuracy with which it predicts the behavior of the actual system in the future.

Result analysis and experiments

Simulation model as such does not give any indication of the possible changes in order to improve desired logistics system performance measures. There are a number of techniques for finding improved or optimized values for the decision variables that are based on probability and statistics theory.

The outcome of a particular stochastic simulation is a random variable. However, the usual independence of errors assumption of standard statistical methods is normally not met in simulation study (Hoover and Perry, 1989). Both serial correlation of errors and heteroscedasticity can be present (Hoover and Perry, 1989). It can be proved (Pindyck and Rubinfeld, 1991) that the expected values still remain the same but any confidence intervals are affected. Another problem is when input data is taken from the real figures the data itself is random variable and does not represent a true value in the statistical formula meaning (Vaughan, 1996). This means that the confidence interval of a performance measure represents the confidence interval inside that model and not the one of the modeled system.

In a "what-if" experimental setting we need to decide between different scenarios and also estimate their difference. This can be achieved by establishing the confidence interval of the performance measure in question. The situation becomes much more complicated if the research objective is to find out how, given a set of decision variables, the performance measures could be improved or optimized. One possibility to deal with multiple input variables is to apply statistical design of experiments approach. Other possibilities in analyzing the model include optimization methods (response surface methodology) and gradient method (Law and Kelton, 1991). Goldberg (1989) even suggest genetic algorithms for search of local optimum. However, all these methods are ill developed in a non-independent errors environment (Hoover and Perry, 1989). The situation can further be complicated when there more than one performance measure that cannot be combined to a single one.

3   SIMULATION MODELING CASE

The research objective of this paper is to experiment the applicability of discrete event simulation model to evaluate the re-engineering potential in a case distribution problem. The immediate case objective is to evaluate the re-engineering potential.

3.1 The present system

The case mill is a large newsprint mill with several machines producing standard newsprint cut to customer specific widths at the mill. The mill has its main markets in Western Europe. The case mill delivery chain is shown in figure 3. The mill has a customer dedicated call-off stock at the port terminal from which deliveries can be made by customer call-off. This is shown with (B) in figure 3. This port terminal stock is replenished from the paper machine (A).
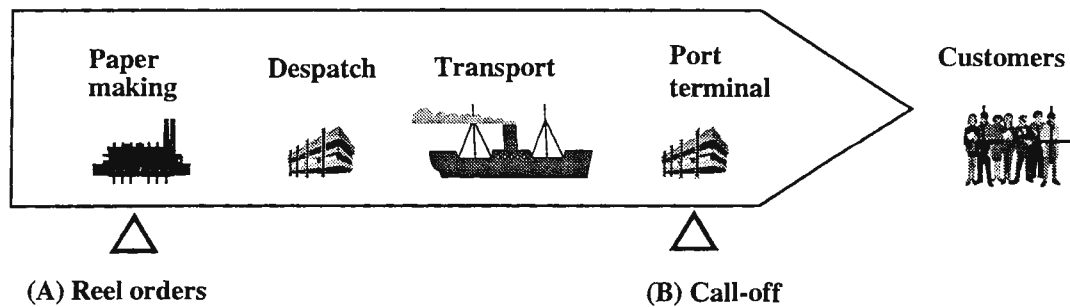
Figure 3 The logistical chain of the case.

The product range in the mill is limited to few standard newsprint grades so there were no paper machine production cycle. The paper is transported to the customers by using two ships; one of which serves three and the other serves two terminals. To be able to reach high shipping capacity utilization and to avoid port visiting charges, the ship takes a full load to one terminal at a time. During the company interviews it was indicated that the high shipping capacity utilization was a high priority goal for the production planning.

This transportation system has two disadvantages. First, the shipping frequency to each terminal is low. Secondly, the system is inflexible in respect to demand variations and changes. In fact, cross-deliveries were found where the customer normally served by some terminal was in some cases served via another thus incurring excess delivery costs.

Based on the understanding of the situation, the problem statement for the simulation study was now be formulated as to find out the inventory savings achievable by a new transport system where ships would serve more than one destinations on the same trip.

3.2 Simulation model

Customer demand was modeled by 60 customers that were divided in large, medium and small customers. The customer dedicated call-off stock is replenished with a reorder point control. If there is not enough paper in the terminal to fill customer order, the first order is backlogged i.e. served when the replenishment arrives but all subsequent orders before replenishment are lost. If there are no orders at the mill, a shutdown is started until order book contains three day's production.

The simulation study was carried out by making five independent 360-day simulations with both the present and scenario transport system. The comparisons between the systems was made at equal availability levels achieved by adjusting both reorder points and replenishment quantities.

3.3 Simulation model of the present system

Figure 4 shows the present transportation system. After production, the paper waits in the home terminal for the right ship going to the destination terminal. If there is not enough paper to fill the ship in the home terminal, the ship leaves and does not wait for more paper. The ships visit only on terminal in each trip. Ship 1 serves three terminals 1, 2 and 3. The back and forth trip to each of terminals 1-3 takes seven days. In five weeks terminals 1 and 3 are visited two times and terminal 2 only once. Ship 2 serves terminals 4 and 5 in a 12-day-cycle
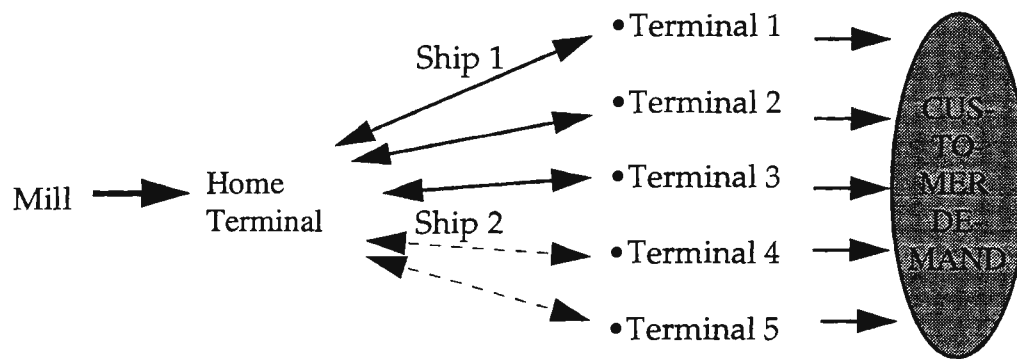
Figure 4 present transportation system.

The present system simulation results are shown in table 1. The average-row shows the five-run average. Rows marked 2,5% and 97,5% give the 95% confidence interval for the average under assumption that the average is normally distributed.

The availability was a little over the target 95% but inside the 95% confidence interval. Average lead times varied from 11 to 27 days among the terminals due to the differences in shipping frequencies. The average total inventory was 31 700 tons.

Table 1 Present system simulation results

	Inventory (tn)			Lead time (days)			Availability
	Home term.	In-transit	Terminal	Term. 1&3	Term. 2	Term. 4&5	Call-off
Average	7 292	2 966	21 440	16.5	27	10.9	95.70 %
Low 2,5 %	6 619	2 665	20 666	15.8	22.3	10.6	94.40 %
High 97,5 %	7 966	3 267	22 214	17.2	31.7	11.2	96.90 %

3.4 The scenario model

The scenario transport system is shown in figure 5. Both ships start making seven-day round-trips. Terminal 2 activity is discontinued which enables ship 1 to achieve weekly service. The customers served via terminal 2 are offered service via terminals 1 or 3. Under the scenario transport system the ships´ routes are longer but the total available shipping capacity per week still exceeds the mill's weekly production by 4,7 %.
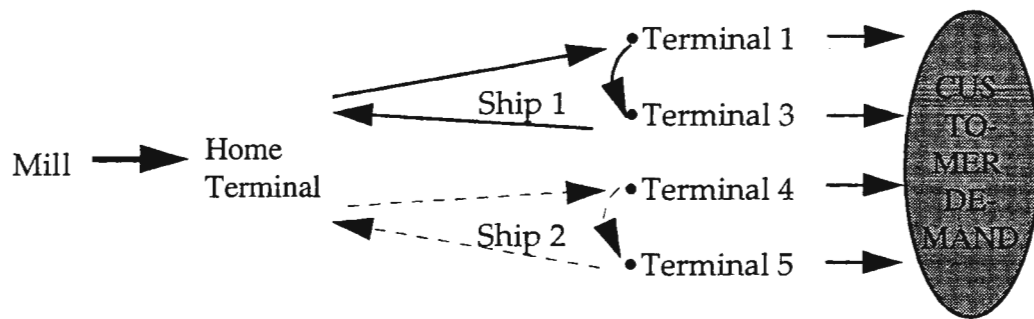
Figure 5 Scenario transport system.

The scenario simulation results are shown in table 2. Average lead times are now cut to 10 days for the remaining terminals. The same availability target is achieved with only 22 700 tons of inventory.

Table 2 Scenario simulation results

	Inventory (tn)			Lead time (days)		Availability
	Home term.	In-transit	Terminal	Term. 1&3	Term. 4&5	Call-off
Average	3 802	3 024	16 051	10.1	10.2	95.30 %
Low 2,5 %	3 608	2 956	15 489	9.8	9.9	94.30 %
High 97,5 %	3 997	3 091	16 614	10.5	10.5	96.30 %

3.5 Assessment of the improvement potential

The randomness in call-off terminal stock replenishment orders together with high capacity utilization make lead time longer than pure transport time and frequency induced delay. The simulation experiment indicated a reduction potential in the total finished goods inventory from 31 700 tons to 22 700 tons or close to 30%. This reduction comes from increased transportation frequency. Increased frequency allows the same availability to be reached with both 25% smaller call-off terminal inventory and with 50% less home terminal inventory due to more flexible transport capacity usage.

Three main benefits achieved by the alternative shipping system can be summarized

- Increased transport frequency allows faster inventory replenishment and therefore smaller inventory.
- With more than one terminal visited each time, variations in just one terminal demand can be balanced with the other terminal demand.
- Response time to orders not on stock is decreased.

4 CONCLUSIONS

The approach in this article calls for few models only: the base case model and the scenarios. The benefit of this technique compared to other results analysis techniques like design of experiments or steepest ascent is that it reduces the time-consuming sets of simulation runs and elaborate results analysis. This is in line with the view that simulation is an economic way to evaluate alternatives before implementation (Mott and Tumay, 1992), although perhaps not that thorough as they claim. In their extensive article on production scheduling approaches Rodammer and White (1988) reach

the conclusion that once problem has been defined and suitable candidates generated, the performance of candidate schedules could be verified using simulation models.

The drawback of the simulation is that model optimum is not even tried to find out (Mentzer, 1989). However, not all the re-engineering actions are found by optimizing present system parameters. The alternative shipping arrangement was not a part of the original, base case simulation model and cannot therefore be found by optimizing the base case model decision variables like reorder point and replensihment quantity. Simulation as such did not prove to be a powerful problem solving method as Hoover and Perry (1989) believe. Instead, the improvement actions were discovered by the use of researcher's previous experience or from existing logistics theory.

The correlated inspection approach to validate the simulation models as suggested by Kelton and Law (1991) or Hoover and Perry (1989) to validate the base case models was not properly carried out. The exact replenishment policies and demand structures were not known for the inventory. It was, however, noted that the true inventory was larger than in the model.

Another consideration is that the role of simulation modeling can be a decision support tool. The improvement actions are decided elsewhere. Then the function of the model is to assess the improvement potential of the suggested improvement actions. For this purpose simulation modeling suits well as it was able to produce quantifiable results for the chosen performance measures.

REFERENCES

Banks J. and Carson J (1984) *Discrete-event System Simulation*, Prentice-Hall, New Jersey.

Bowersox, Donald J., Closs, David J., and Helferich, Omar K. (1986) *Logistical Management - A Systems Integration of Physical Distribution, Manufacturing Support and Materials Procurement*, 3rd edition, MacMillian Publishing, U.S.A.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Massachusetts.

Hoover S. and Perry R. (1989) *Simulation - A problem-solving approach*, Addison-Wesley, USA.

Johnsson M. (1992) *Discrete Event Simulation, An Evaluation Tool for Logistic Systems*, Lund Institute of Technology, Lund, Sweden.

Law, A and Kelton, W. (1991) *Simulation Modeling & Analysis*, McGraw-Hill, U.S.A.

Menzer, J. (1989) Symposium Feature: Approaches to Logistics Modeling, *Journal of Business Logistics*, vol. 10, no 1.

Mott J. and Tumay K. (1992) Developing A Strategy For Justifying Simulation, *Industrial Engineering*, July 1992.

Naylor, T., Balintfy, J., Burdick, D. and Kong, C. (1966) *Computer Simulation Techniques*, John Wiley, U.S.A.

Pindyck R. and Rubinfeld D. (1991) *Econometric Models & Economic Forecasts*, McGraw-Hill, Singapore.

Rodammer, F. and White, P. (1988) A recent survey on production scheduling, *IEEE transactions on systems, man and cybernetics*, Vol. 18, no.6.

Strandhagen, J. (1994) *Operative simulation in production management*, Doctoral thesis, NTH Trondheim, Norway.

Vaughan, T. (1995) The effect of sampling variability on statistical order point computation, *Production and inventory management journal*, 3rd quarter.

SIMULATION ENVIRONMENT FOR AUTONOMOUS VEHICLES

J. Scholliers[1], P. Peussa[2]

[1]VTT Automation, Machine Automation, P.O.Box 13021,FIN-33101 Tampere, Johan.Scholliers@vtt.fi
[2]VTT Automation, Machine Automation, P.O.Box 13021, FIN-33101 Tampere, Pertti.Peussa@vtt.fi

Abstract

The control algorithm is one of the most critical components of an autonomous vehicle. A graphical tool is desired, in which the control algorithm can be tested in different environments. At VTT, a simulation environment is developed based on 3D Studio MAX, a commercial animation tool for PC. A simulator plug-in has been developed, which uses dynamic linked models of both the vehicle dynamics and of the environmental perception sensors.

Introduction

Autonomous vehicles perceive the environment by means of environmental perception sensors, and adapt their motion in function of the sensor data e.g. to avoid collisions. The critical component is the control algorithm, which fuses the data of the sensors and generates the commands for the steering motors. A graphical simulation tool, which allows generating different environments rapidly, is desired to test the control algorithm. The simulation environment should also allow testing for the optimal position of the sensors and could also be used to demonstrate the behaviour of the autonomous vehicle and for learning. The environment should be modular, so that the effort required implementing a change in the control algorithm or in the vehicle dynamics model is as small as possible.

Simulation of an autonomous vehicle in 3D Studio MAX

3D Studio MAX [1] from Autodesk is one of the world's most popular professional 3D modelling and animation software for PC. The program has advanced graphical possibilities and uses object-oriented techniques to generate complex graphical environments. New geometrical objects and graphical effects can be added as plug-ins. 3D Studio MAX is an animation program, in which the position of an object at each moment is determined by a "controller", which is assigned to the object. The controller can be converted to a simulator by calling at each time step an external routine, which models the vehicle dynamics and returns the position of the vehicle. The simulator-controller plug-in for autonomous vehicles, which has been developed at VTT, consists of the following parts (Figure 1):

- The graphical model of the vehicle in 3D Studio MAX, to which a simulator "controller" is assigned.

- The sensor models, which extract the information from the graphical environment.

- A dynamical model of the vehicle, developed with e.g. MATLAB/SIMULINK. The inputs to the model are the sensor signals and the outputs are the new position of the object.

At each time step in 3D Studio MAX, the simulator controller retrieves the sensor data and sends them to the
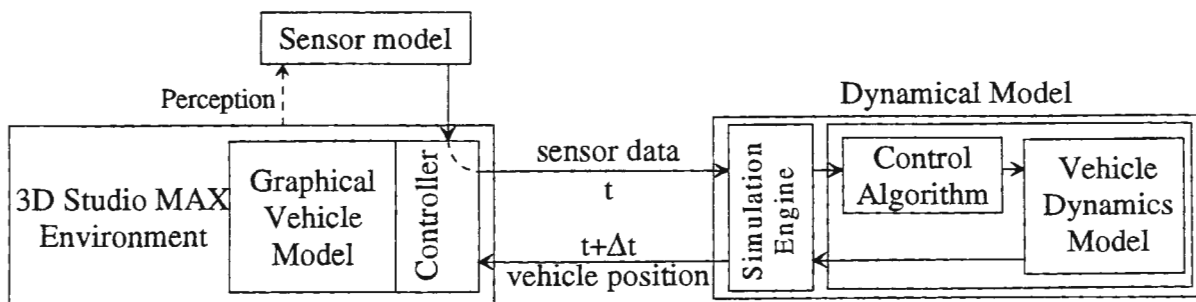
Figure 1: Information flow between 3D Studio MAX and the dynamical model

dynamical vehicle model, which calculates the position of the vehicle for the next time step. The vehicle moves on a flat surface, which is a very common environment for autonomous vehicles, like office environments, mechanical workshops and harbours. The main application within VTT at this moment is the development of the control algorithm of an assisted wheelchair.

Model of the Sensors

The environmental perception sensors are modelled in an object-oriented way. The sensor model uses 3D Studio MAX routines to retrieve information from the graphical environment, in the same way, as the physical sensor would measure in the real world. A roll-up window in the graphical environment allows manipulating the sensor parameters, as its position, the range and the beam width. The sensor model communicates through a proprietary interface with 3D Studio MAX and is linked dynamically. New sensor models can be added by writing the appropriate functions and compiling to a dynamic link library (DLL).

All sensors in the wheelchair are ultrasonic sensors, which return a signal proportional to the distance to another object along the emitted ray. No reflections are noticed if the distance to the object is too small or too large or if the angle between the ultrasonic ray and the normal to the surface is larger than a predetermined angle. The ray is modelled as a line. The sensor model checks for the intersection between this line and the different objects in the graphical environment, and for the validity of the intersection.

Dynamical model of the vehicle

The dynamical model calculates the position of the vehicle in function of the sensor data and a set of optional external inputs, like the desired velocity. The external inputs can be controlled from a roll-up page in the 3D Studio MAX user's interface. The model is compiled to a DLL, which is linked dynamically to the model.

A proprietary C-structure contains the information on the model, as the number of inputs and pointers to the input vectors, the next time step, scaling information, and a void pointer to data defined by the model. The model interfaces to 3D Studio MAX through a set of proprietary routines for loading, updating and exiting the model and for saving model data.

The model consists of two parts: a simulator engine, which drives the simulation of the vehicle model, and the
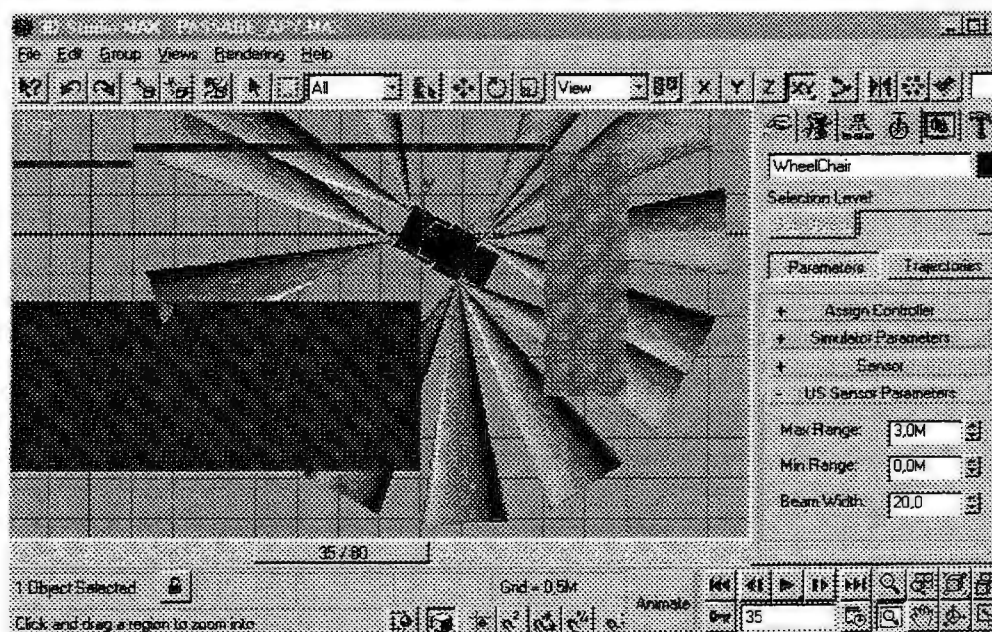
Figure 2: Top view of the Wheelchair moving between obstacles in the graphical 3D Studio MAX environ ment
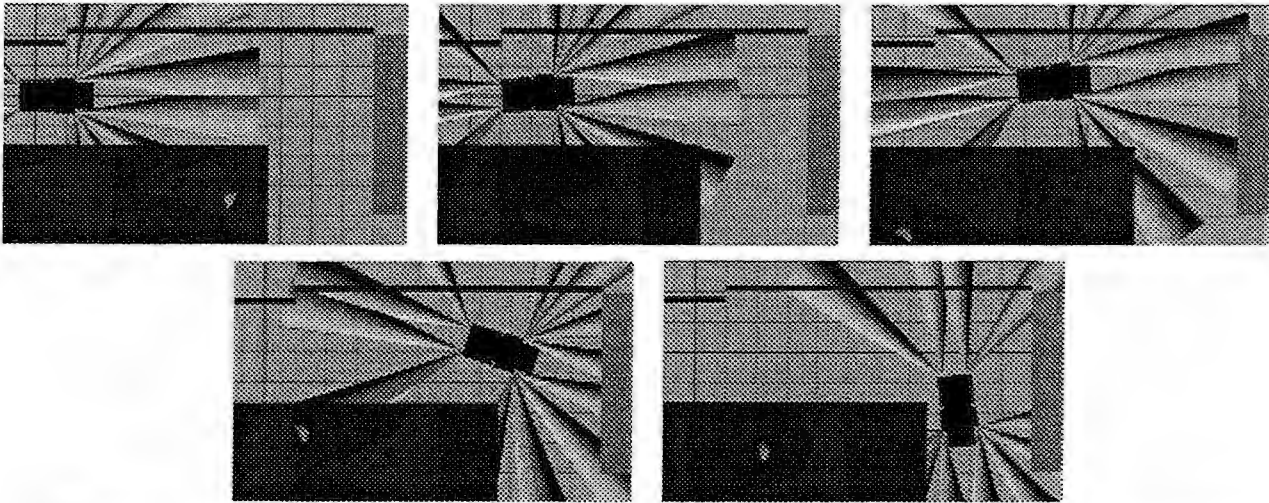
Figure 3: Simulation of the motion of the wheelchair, with as command input straightforward motion

actual vehicle model. The actual vehicle model is, in the case of the wheelchair, developed with MATLAB/SIMULINK. C-code for the model is generated with Real Time Workshop [2]. The simulator engine has to be developed only once, and can be linked to any other Real Time Workshop code.

The actual vehicle model consists of two components: the control algorithm, which calculates the motor steering commands based on the sensor data, and the vehicle dynamics model, which calculates the new position from the steering commands. If the actual controller is developed in a higher programming language like C, the actual control algorithm can be used almost directly in the model. The only difference is that the interface between the control algorithm and the hardware has to be replaced by an interface routine to the MATLAB/SIMULINK model.

Example: Assisted Wheelchair

The wheelchair contains 12 ultrasonic sensors [3]: 4 at the front left, 4 at the front right, 2 at the rear left and 2 at the rear right. The sensor model is dynamically linked to the vehicle, and represented as cones in the graphical environment (Figure 2). The cone does not represent the actual behaviour of the sensor, but it allows verifying the entered parameters of the sensor, as its position and the range.

The driver controls the wheelchair with a x-y-joystick. In the 3D Studio MAX model, the joystick inputs can be set on a roll-up page in the 3D Studio MAX environment. The vehicle model is developed with MATLAB/SIMULINK, and the code generated with Real Time Workshop. The vehicle model uses the actual control algorithm, which is developed in C, and which includes collision avoidance. Figure 3 shows a simulation of the wheelchair, with as input straightforward motion, and how collisions are avoided.

Conclusions

The animation program 3D Studio MAX is used to simulate automated vehicles in VTT. The system allows testing the control algorithm of the automated vehicle for different environments, without having to implement the control algorithm on the actual vehicle. The system allows also to test the control algorithm for hazardous environments, which are difficult to realise or dangerous to test. By testing the algorithm for various environments, the simulation environment helps to increase the reliability of the control algorithm.

References
[1] 3D Studio MAX User's Guide, Autodesk Inc., 1996
[2] Real-Time Workshop For Use with SIMULINK, User's Guide, TheMathWorks Inc., 1994
[3] J. Alanen, O. Karasti, P. Mattila, V. Säntti, A. Virtanen, A Drive Assistant for a Wheelchair, CESA '96 IMACS Multiconference, Lille, France, July 9-12, 1996, pp. 376-381

MULTIBODY SIMULATION OF A NEW SPINDLE DRIVE DESIGN

Dipl.-Ing. Matthias Balázs [1], Dipl.-Ing. Martin Spieck [2]

[1] DLR (German Aerospace Center), P.O.Box 1116, D-82230 Wessling, Germany, matthias.balazs@dlr.de

[2] DLR (German Aerospace Center), P.O.Box 1116, D-82230 Wessling, Germany, martin.spieck@dlr.de

1 Abstract

At the German Aerospace Center (DLR), a new friction drive, the DLR planetary roller spindle drive, was invented to transform fast rotation into powerful translation. The new operating principle raised severe difficulties to precalculate the characteristics of modified versions. To predict the dynamic behaviour of DLR planetary roller spindle drive design versions, e.g. transmission ratio, load dependent slippage, dynamic contact pressure or unsteady operating characteristics, the planetary roller spindle drive was modelled as a multibody system (MBS) and computed in MBS simulation. A new force element was implemented into the MBS simulation code to describe the contact mechanics between the drive elements.

2 Operating Principle of the PRSD

A new spindle drive, named DLR planetary roller spindle drive (PRSD), was developed by DLR German Aerospace Center to transform rotation of a servo motor into translational motion, [1]. A section of a standard PRSD with its components is shown in figure 1.
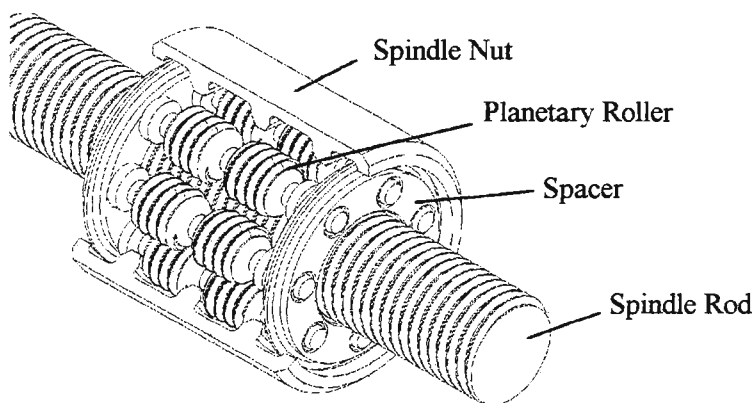
Figure 1: Section of the PRSD

The planetary rollers circle in the spindle nut on their large guide grooves which intermesh with the guide rings of the nut. With their fine profile, the rollers intermesh with the spindle rod thread, guiding the rod radially like needles of a needle bearing.

If the spindle nut is axially and radially fixed and the spindle rod rotates, the planetary roller set circles at a lower rate of rotation expediting the rod like a screw nut, but at a feed motion which depends strongly on the type of fine profile of the rollers and the diameters of the rolling circles. Based on this principle, any arbitrary conversation-ratio is realisable - very high, backdriveable feeds as well as very small ones.

Despite the simple operating principle, the complex contact geometry and the construction-related clearances, leading to considerable relative movements, complicate a detailed theoretical comprehension. Existing theories on gears and transmissions proved to be valid only under substantial restrictions. The need to predict the effects of alterations and modifications of the PRSD design on the overall dynamic behaviour and the performance led to the multibody modelling and simulation of the PRSD.

3 Single Contact Mechanics

Analyzing the contact mechanics between the PRSD elements, two problems have to be solved:

- determination of the contact points,
- evaluation of the contact forces acting at these points.

Finite element analysis of planetary roller and spindle nut has shown that the elastic deformations of the bodies themselves can be neglected compared to local deformations of the flank surfaces at contact points. The PRSD elements are therefore considered as rigid bodies which can virtually immerse into each other. The applied forces depend on the depth of their "penetration" (quasi-elastic contact). Basing on this modelling, the point of

attack of a contact force, i.e. normal and friction force, is defined at the location of the deepest immersion of the contacting surfaces, see figure 2.
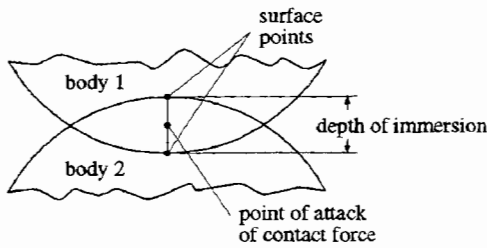
Figure 2: Point of attack of the contact force and immersion depth

3.1 Contact points

External forces on the PRSD are transmitted via two different types of contact:
- contact between spindle nut and planetary rollers,
- contact between spindle rod and planetary rollers.

Contact forces between the spacers and other PRSD elements can be omitted.

The contact problem between spindle nut and planetary rollers can be reduced to the intersection between simple geometric bodies: ball and cone, ball and torus or cone and torus (depending on the PRSD version). Contact points can therefore be calculated analytically.

More difficulties arise at the contact of spindle rod to planetary rollers. Contact pairings which are possible, and have also been realised, are thread to annular groove, thread to thread, thread to opposite-handed thread, each shaped either with straight or spherical thread flank. Parametric representations of the contact surfaces deliver surface vector and normal vector on the surface for each body. A potential point of contact, i.e. where the distance between the surfaces reaches a minimum, is characterised by two conditions:
- the normal vectors of the surface points of contact have to be parallel,
- the normal vectors of both surface points have to be align.

The conditions can be formulated as a set of nonlinear equations. It can be solved by numerical algorithms (e.g. NEWTON method, "linesearch" algorithm, see [2]). Figure 3 demonstrates a parametric representation of spindle rod and roller surface and the calculation of their contact points.
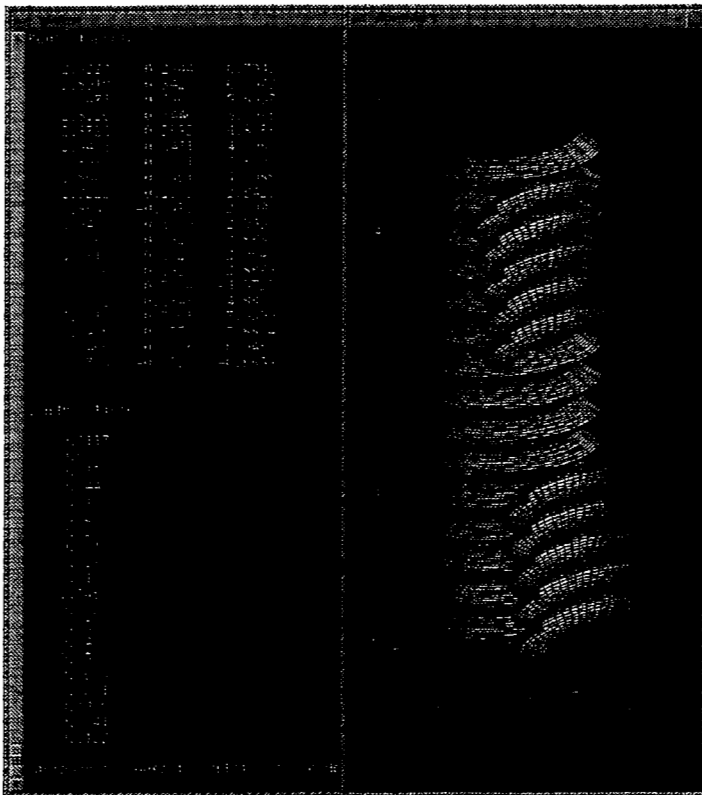
Figure 3: Calculation of contact point coordinates and immersion depths at rod-roller contact

3.2 Contact force

The normal component of the contact force is calculated by the HERTZIAN approach, [3]. It is a function of the characteristic material values elastic modulus and POISSON ratio, the HERTZIAN radii of curvature of the contact surfaces and the depth of immersion of the two bodies at the focused contact.

The direction of the normal component of the contact force is given by the normal vector, the frictional component acts at the contact point in opposite direction of the relative movement between the surfaces.

4 Multibody System Approach

The applied multibody code SIMPACK has been developed at DLR as a tool for the simulation of complex dynamic systems in aerospace, transportation systems and robotics applications. A most valuable feature of SIMPACK is the possibility to generate user specific MBS elements. n this application, two new force elements were defined to describe the complex contact conditions of the PRSD, one for the contact between planetary roller and spindle nut (force element *nut-roller*), the other for the contact between roller and spindle rod (force element *rod-roller*).

At a new time integration step, a measurement matrix is passed to each contact force element in the model containing transformation matrices, spatial position, orientation and absolute velocities of the concerned bodies. Each force element determines the contact points and the contact forces between the regarded surfaces, combines all contact forces to a single force and moment vector and passes this result vector to the time integration formalism of SIMPACK where the calculation is continued. Figure 4 shows an animation picture of an existing PRSD model simulating low speed feed.
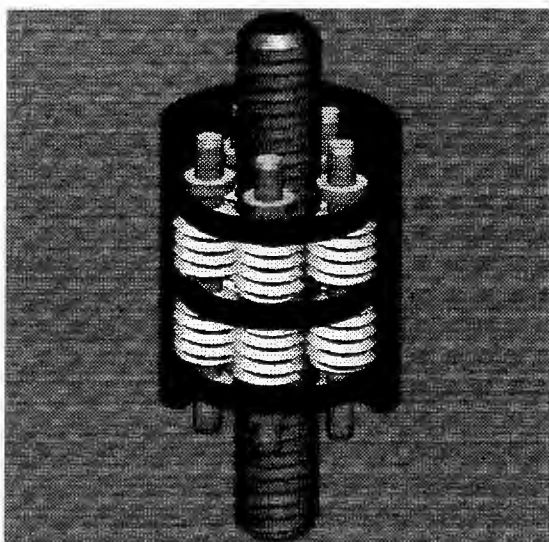
Figure 4: Animation picture of a PRSD version in the testbench

A particularity of this modelling approach is that two numerical equation solvers are run in parallel: the internal SIMPACK solver for the equations of motion and the implemented algorithm for the determination of contact points. A specially developed algorithm margins the allowable solutions of the contact point solver to a bandwidth in which the focused point of contact is the only extremum. Therefore, interference between the solvers and resulting unstable effects can be avoided.

5 Results

For verification, an existing PRSD test-bench measurement was modelled and computed. The simulation results did not only show an excellent accordance with the measured overall performance, but gave, for the first time, access to MBS measurement data of each PRSD body separately. This enabled the investigation of component interactions and other effects where the dynamic behaviour of single PRSD elements are important. As an example, for improperly designed or manufactured PRSDs, certain working conditions may lead to jamming; an effect which the MBS simulation proved to emulate.

The investigation of different PRSD designs has lead to a better understanding of the PRSD and has reduced the number of design loops for new PRSD versions considerably, saving both money and time.

6 Summary

An MBS model of the PRSD has been set up and implemented into an MBS simulation code. The resulting PRSD simulation tool allows detailed examination and optimisation of new PRSD designs.

7 References

[1] Dietrich, Gombert: European Patent No. 0320621 „Vorrichtung zur Umwandlung einer Drehbewegung in eine Axialbewegung", 1988, (in german).
and: Dietrich, Gombert: United States Patent No. 4926708 „Apparatus for converting rotary motion into axial motion", 1990

[2] Dennis, J. E.; Schnabel, R. B.: Numerical Methods for Unconstrained Optimation and Nonlinear Equations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[3] Hertz, H.: Über die Berührung fester elastischer Körper und über die Härte., in: Journal für reine und angewandte Mathematik, 92, Celle, 1882, (in german).

8 Authors

Matthias Balázs
DLR, Institute of Robotics and System Dynamics, P.O.Box 1116, D-82230 Weßling, Germany
Phone: +49 (8153)28 -1622, Fax: +49 (8153) 28-1134
E-Mail: matthias.balazs@dlr.de

Martin Spieck
DLR, Institute of Robotics and System Dynamics, P.O.Box 1116, D-82230 Weßling, Germany
Phone: +49 (8153)28 -2432, Fax: +49 (8153) 28-1850
E-Mail: martin.spieck@dlr.de

A Simulink block library for steam turbine simulation

Juha-Pekka Jalkanen, Martti Välisuo

Control Engineering Laboratory, Helsinki University of Technology, Otakaari 5 A, 02150 ESPOO, Finland,
IVO Technology Centre, Power Plant Technologies, 01019 IVO, Finland,

Abstract:

A simple steam turbine simulator toolbox is made for Matlab® Simulink® 1.3c.
At the moment toolbox consist of four blocks: a valve, a steam storage, a turbine block and an electric generator. A complicated simulator can be constructed from those basic blocks. This paper covers description of the toolbox and a test run with comparison to SOLVO®, a static simulator by IVO Technology.

1. INTRODUCTION

A need for a simple tool for steam turbine simulations occurred. There are various types of turbines so the simulator should be flexible. Main purpose is to simulate new control and automation strategies so the models do not have to be exact. In stead, the simulator should have an easy way to implement different kinds of controllers.

There are many simulation tools for power plant simulations such as APROS by Technical Research Centre of Finland and IVO Technology Centre. Unfortunately, simulation programs are too complicated for average personnel of power company because of too exact physical models and as a result too many free parameters. Another point is that in stead of learning a new simulator tool for every problem personnel would like to use a familiar one to simulate their newest inspirations.

Matlab® Simulink® is an easy to use simulation software widely used by universities and students. It has graphical user interface with lots of basic blocks from which simulators can be constructed. Simulink® offers several different integration routines and simulated data is handled flexibly with Matlab® features.
The turbine simulator toolbox was made using an older version called Simulink®1.3c.

2. SIMULINK® BLOCK LIBRARY

At the moment toolbox consist of four blocks: a valve, a steam storage, a turbine block and an electric generator. A complicated simulator can be constructed from those basic blocks. The idea is much like the one introduced in [Ordys et.al 1994] but unlike in the previous mentioned literature in this simulator physical values of the steam are evaluated from the steam tables. Tables offer easy and effective way to calculate for example thermodynamic power from the steam turbine. Components except the steam storage and the generator are modeled as static ones. This is possible because of the very small time constant of the real system. Dynamics of the steam occurs in the steam storage. Therefore, a steam storage must be placed between every static component. The mass flow of the steam is evaluated in valves and turbine blocks. Storage takes care of the pressure level of the steam. Other major quantities are temperature and enthalpy of steam.

2.1 Physical presumptions and constructions

It is difficult to do a simulator from energy section without presumptions. A common presumption is that steam acts like ideal gas, which of course does not happen, although errors are quite small. A widely used

solution to prevent ideal gas presumptions is to use steam tables. However, relative error is about the same, when small size tables or ideal gas equations are used and large enough tables are difficult to handle. There are some iterative functions for accurate steam calculations but when functions are evaluated with integration methods that change integration step size, such as Runge-Kutta 45, simulations become too slow. In the simulator constructed a compromise was done. In most places, ideal gas equations were used. Steam is usually superheated and so acts like an ideal gas. In low pressure turbines larger error comes from presumption that steam do not condense.

In some critical places steam tables were used but were modified a little. They have to start from very low values if the start-up procedure of the turbine is simulated. To prevent huge size of the table, accuracy of the table is good only in the common steady states.

Steam tables were provided by IVO Technology Centre.

2.1.1 Valves

It is assumed that neither enthalpy nor temperature does change in a valve.

Outlet pressure of the valve is calculated in the next storage. Mass flows trough the valve is calculated as follows

$$q = C_{V2} f(x) \sqrt{\rho * (p_1 - p_2)}$$

(1)

where:

q is mass flow

C_{V2} is constant characteristic to a valve

$f(x)$ is flow factor as a function of opening of the valve

ρ is density of the steam and

p_1 and p_2 are pressures before the valve and after it

Usually turbine systems have approximately four parallel valves that are controlled so that overall mass flow is linear. If $f(x)$ is choused wisely, a valve's model can vary from linear to highly nonlinear when some parameters in $f(x)$ are changed. Therefore, instead of four different valves the model can cope with only one. In the other hand more complicated models with nonlinear valves can also be simulated.

2.1.2 Steam storages

Enthalpy of outlet steam is evaluated from mass and energy balances. Equations provided by [Ordys et.al 1994] operated well enough and were used in the simulator. Difference is that unlike in [Ordys et.al 1994], where only incoming mass flow is input quantity and storage had first order dynamics, in this simulator both inlet and outlet mass flows are input quantities for steam storage.

2.1.3 Turbine section

The mechanical power from the turbine is evaluated from the enthalpy difference. Enthalpy in the end of the turbine is calculated from entropy and pressure. Pressure is input quantity to the system and steam tables are used both in calculation of enthalpy and entropy. Mass flow trough the turbine is evaluated from a little simplified form of the ellipse law as follows

$$q = \frac{K}{\sqrt{\frac{p_1 \frac{1}{\rho_1}}{p_1^2 - p_2^2}}}$$

(2)

Where K is turbine constant, that operates as a kind of tuning parameter.

3. NOTES ON SIMULATIONS

3.1 How to choose integration algorithm

A basic choice for integration algorithm has usually been Runge-Kutta 45 method. In this simulator there are many steam tables included and Runge-Kutta becomes very slow. Adams and Gear methods are more than ten times faster, even thought system really is not a stiff one, unless the generator is installed in the model. Unfortunately, Adams and Gear methods do not cope step responses, that is why they are more or less useless, although step responses do not occur in real systems.

3.2 Initial values

Thermodynamic equations of the simulator constructed are very sensitive respect to initial values. Considerable oscillation occurs before the simulator reaches the steady state. Amiss initial values lead pressure easily beyond of the steam table's regions. For better results, each block should be tuned probably. Some physical initial values can be calculated from thermodynamic laws, but the rest must be tuned by simulating the system. Tuning becomes easier when the initial values of the pressures are given by the user. Then mass flows are tuned by simulating the system for half of a second.

3.3 Problems with combining static and dynamic components

System dynamics occur in steam storage's. The capacity of the storage acts as a time constant for the dynamics of the pressure. As other components are modeled as static, they do not include any dynamics. Therefore, changes in any other component are unnaturally fast. If white noise is added to input quantities of the system, the steamflow will have unnatural dynamics.
Fast dynamics should be included to every block to prevent rapid changes.

4. COMPARISON TO SOLVO

The steady state values of the dynamic turbine simulator blocks were compared to SOLVO, that is a static simulation environment for power plants used for many years by IVO Technology Center.
Comparison was done with a simple system with high pressure and semi pressure turbine sections and one steam storage between them. Two step changes were made to pressure from initial value of 80 bar, first to 90 bar and then to 50 bar.
Turbine constants and isentropic efficiency are 0.00175 and 0.8 for high pressure- and 0.01166 and 0.9542 for semipressure turbines.
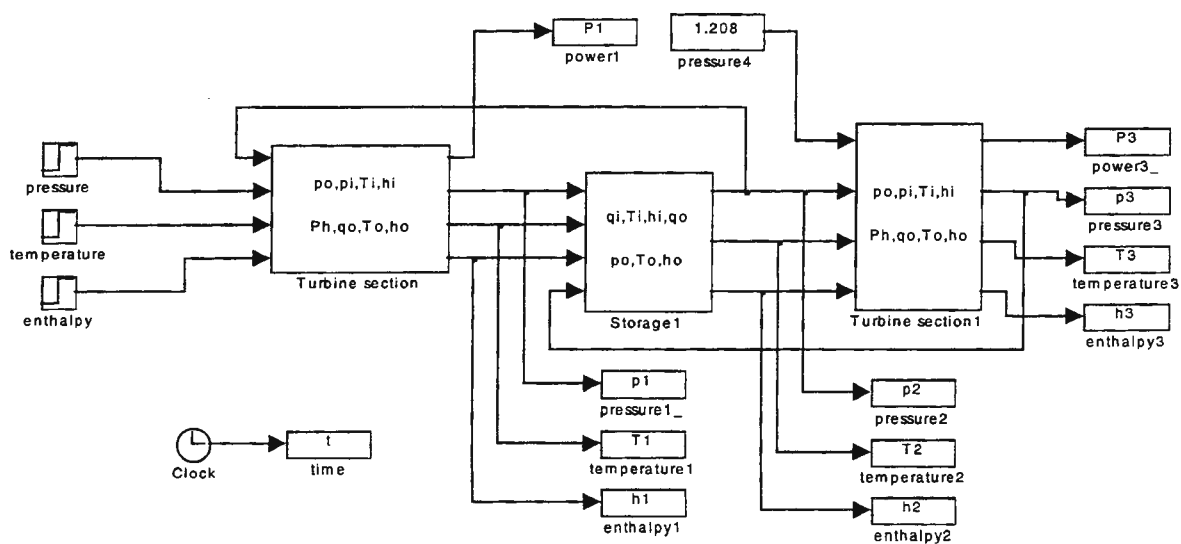
Figure 1 Comparison model

Table 1 Pressures before, in the middle and after the turbines [bar]

SOLVO	89.08	10.9	1.208
DTS	89.08	10.945	1.208
diff	given	0.41 %	given
SOLVO	80	9.8	1.208
DTS	80	9.866	1.208
diff	given	0.67 %	given
SOLVO	50.52	6.398	1.208
DTS	50.52	6.332	1.208
diff	given	-1.04 %	given

Table 2 Temperatures before, in the middle and after the turbines [K]

SOLVO	717.01	482.99	378.15
DTS	717.01	482	375.83
diff		-0.21 %	-0.62 %
SOLVO	717.55	485.15	378.15
DTS	717.55	484.51	375.83
diff		-0.13 %	-0.62 %
SOLVO	717.55	492.02	378.15
DTS	717.55	489.9	375.83
diff		-0.43 %	-0.62 %

Table 3 Mass flow trough turbines in different pressures [kg/s]

DTS	25.346
diff	-0.61 %
SOLVO	22.75
DTS	22.75
diff	0.00 %
SOLVO	14.25
DTS	14.365
diff	0.80 %

Table 4 Enthalpies before, in the middle and after the turbines [kJ/kg]

SOLVO	10167.9	9856.2
DTS	10179	9708.8
diff	0.11 %	-1.52 %
SOLVO	9172.7	8518.3
DTS	9216	8422
diff	0.47 %	-1.14 %
SOLVO	5905.4	4228.2
DTS	6018.83	4343.4
diff	1.88 %	2.65 %

Table 5 **Power from both turbines in different pressures**

SOLVO	3244.4	2845.6	2459.1
DTS	3244.4	2842.8	2459.7
diff	given	-0.10 %	0.02 %
SOLVO	3260	2856.8	2482.4
DTS	3260	2855	2484.8
diff	given	-0.06 %	0.10 %
SOLVO	3303.6	2889.2	2592.5
DTS	3303.6	2884.6	2582.2
diff	given	-0.16 %	-0.40 %

5. A CASE STUDY

As a case study, we have a valve and two turbine sections (high and semi high pressures). Input pressure of the system contains white noise. Total mechanical power of the system is measured and then controlled with slow PI-controller. A step change is made to set point from 17690kW to 18000kW of mechanical power.

Noise in the pressure input is a problem for dynamic simulator. Static models introduced in [Ordys et.al 1994] are very sensitive for noise. Also in this simulator steam storage that evaluates the pressure in valve's output is not fast enough to react to the white noise. Extra filters should be involved in models to damp the noise. Noise damps towards the end of the turbine chain and turbine sections work already in a normal way.
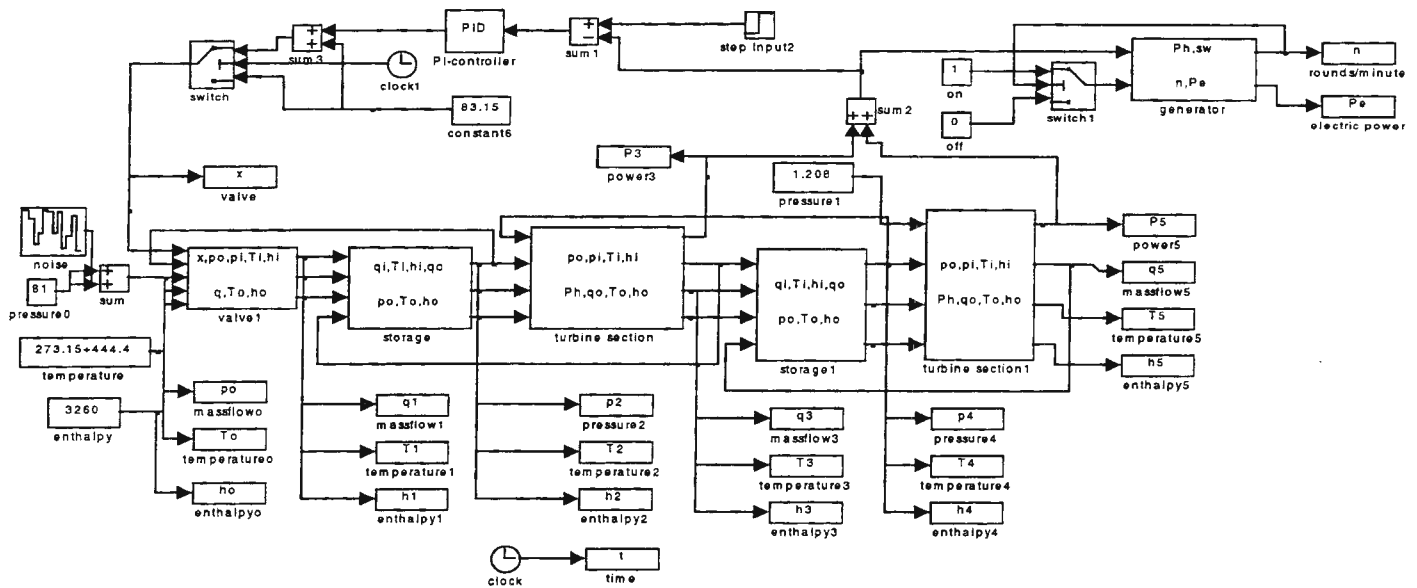
Figure 2 Model of the case study

Even though pressures between the valve have slightly difference in dynamics they seems to correlate well.
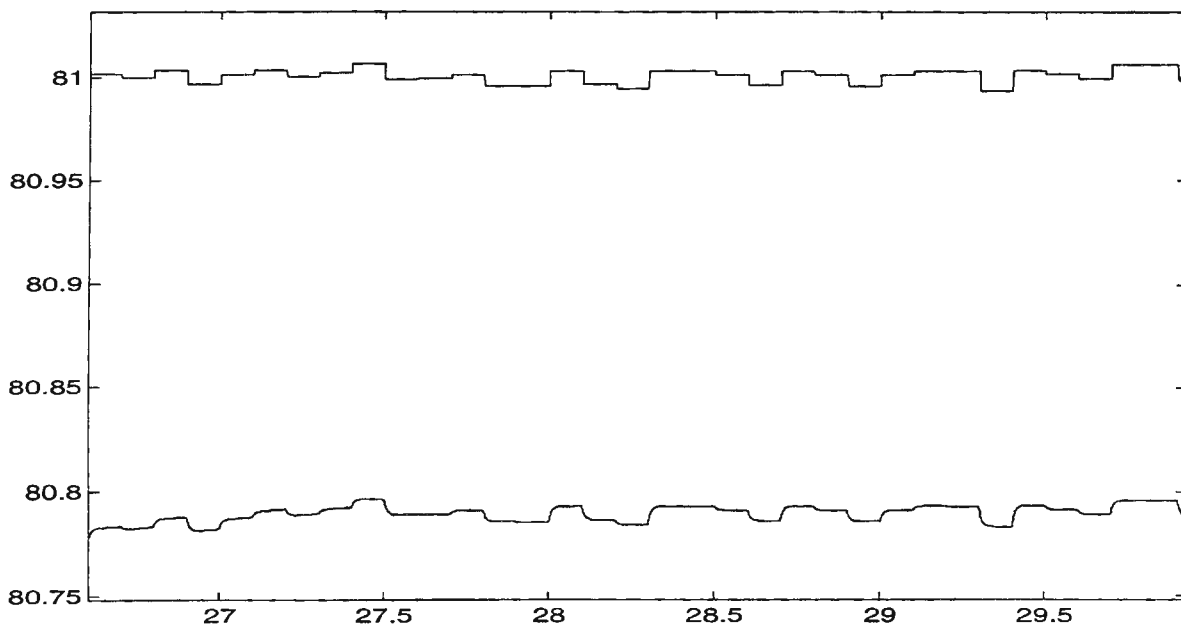
Figure 3 Zoomed pressures before and after the valve.

Overall dynamics of the system are sensible.
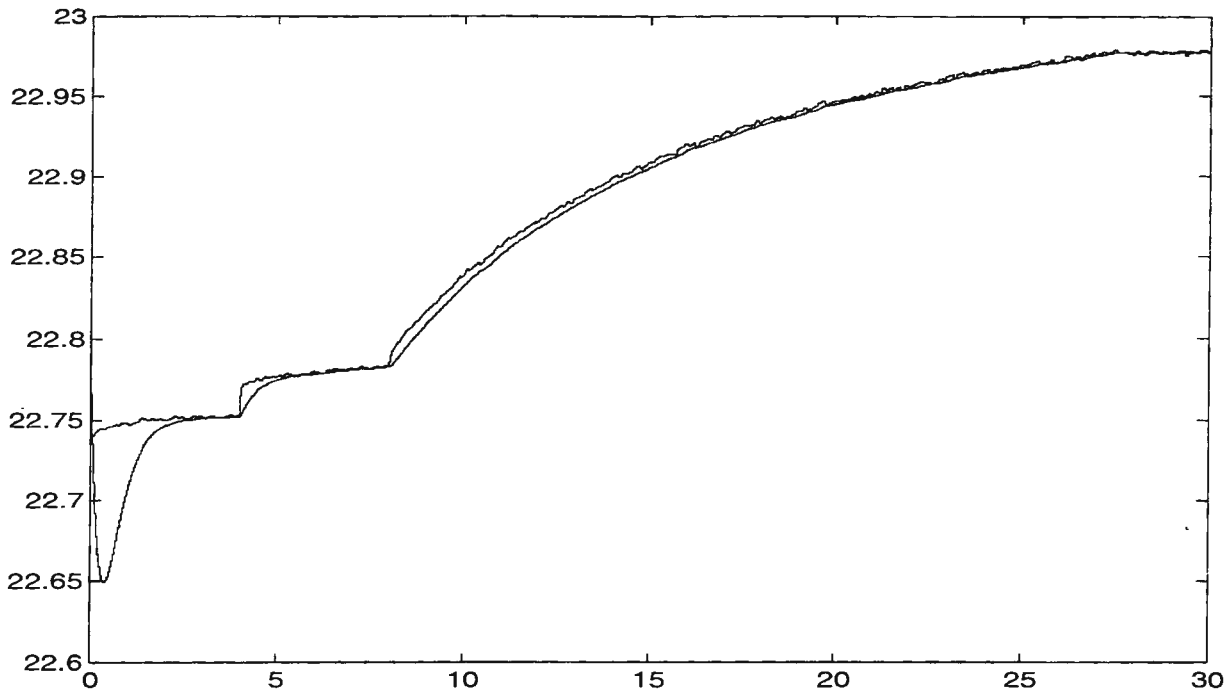
Figure 4 Mass flows trough both turbine sections

It can be seen how even little wrong initial values cause oscillation in the beginning. In addition, the dynamics of the first turbine section is considerably faster than second's but for slower changes simulator acts more affirming. Fortunately, there are seldom quick changes in steam turbine systems.

6. CONCLUSIONS

DTS suits well to small-scale simulations. It is easy to use for anybody who has used Matlab® Simulink® before. Dynamics of the blocks are convincing and accuracy is good enough for control strategy simulations. A toolbox like this would suit well for energy technology students in universities where too often dynamic behavior of the system is forgotten.

7. ACKNOWLEDGMENTS

REFERENCES

A. Ordys, R. Katebi, M. Johnson, M. Grimble

Modelling and Simulation of Power Generation Plants. Springer-Verlag, 1994

HARDWARE-IN-THE-LOOP SIMULATION
OF RAIL-VEHICLES AT ADTRANZ

Thomas Keller, Hans-Peter Wenk, Erich Scheiben
Adtranz (ABB Daimler-Benz Transportation) Switzerland
Dept. BAA, PO Box 8384, 8050 Zurich, Switzerland

Abstract. The present contribution gives a review of hardware-in-the-loop simulation of rail vehicles at Adtranz starting with a purely analog simulator and ending up with a purely digital simulator. The problems encountered and solved with analog and hybrid simulators are discussed and the simulator topology, the simulation model and simulation results of a completely digital real-time hardware-in-the-loop simulation of a converter vehicle are presented and compared to an analog simulation.

Keywords. Analog hardware-in-the-loop simulation, digital hardware-in-the-loop simulation, control system testing, discrete events, hybrid system

IN THE BEGINNING

ABB Daimler-Benz Transportation has over 30 years of experience in real-time hardware-in-the-loop simulation of rail vehicles. In the 1960's, a project to control the speed of a conventional locomotive with mixed load was undertaken. A difficult problem in this project was to deal with the large time constants and dead times of the system in order to meet the goal of constant speed regulation within +/-2 km/h.

For the same reasons that persist today, (i.e., availability of tracks, limited numbers of tests per day, human safety and material loss), it was decided to test the controller with real-time hardware-in-the-loop simulation. At that time, a PACE analog-computer with 27 integrators was used for the computationally demanding part of the simulation. This computer had capability to simulate about one-half of a conventional locomotive. In order to simulate the rest of the locomotive (including the main air tube and the brake valves) special hardware was constructed. A lesson from this project was that it is essential to have a realistic simulation.

AD4

The next generation of simulators arrived in 1969. ABB Transportation Systems purchased the hybrid analog computer AD4 which included 64 integrators (op amps). This simulator featured exceptional amplifiers with good frequency response characteristics. Together with the fast electronic switches and the control-logic, this simulator enabled the simulation of many important projects for nearly 20 years. Projects simulated on this system included vehicles with bridge-converters and DC-drives. During these two decades, a wealth of in-house knowledge was obtained in realtime hardware-in-the-loop simulation.

However, a major disadvantage was that the programming was done "by hand". What is now programmed in software was at that time hardwired on the patch panel. That is, the interconnection between integrators (op amps) as well as the potentiometers and function generators settings had to be done by hand. Consequently, a successful programmer required a great deal of knowledge about the electronics behind the patch panel of the analog computer.

SIMSTAR

In the mid 1980's, after more than 15 years with AD4, a new generation of simulators were considered. The aim was to minimize the need of hardware know-how about the simulator itself, to simplify the programming and to improve the simulation documentation. The requirements were to simulate the whole locomotive drive system from powerstation to train mechanics. The digital simulator AD100 available at that time attained a

benchmark simulation cycle time of about 300 μs which was too slow to be used for the locomotive drive simulation.

As a consequence, the hybrid simulator SIMSTAR was chosen for the task. A SIMSTAR system consisted of a host digital computer, a digital arithmetic processor (DAP), data conversion processor (DCP), and up to 6 parallel simulation processors (PSP) or patch panels. The particular SIMSTAR system used at ABB Transportation was a stand alone version in that the DAP doubled as a host computer. One and a half fully expanded PSP's (patch panels) were obtained providing up to 60 integrators for a simulation. The primary bottleneck in our applications was the switching of the power electronic converters and the digital inputs.

The STARTRAN programming environment on the DAP provided the user with a convenient method of describing an application in one simulation language source file. STARTRAN used ACSL as its foundation. The user could target for digital, parallel or combined solutions. The PGS-preprocessor parsed the source file and split it up into its digital and parallel parts sending them to the appropriate translator. The low level processing occured after the DTRAN and the PTRAN translator finished. Separate load images are created for the digital and the parallel processors (including switch matrix). Run-time analysis and debugging were supported as well. The SIMSTAR processors were then loaded and control is given to the SIMSTAR run-time executive.

A representative project simulated on the SIMSTAR was the tram for the city of Zürich. This tram had a new electro-pneumatic brake system with unknown time constants and measurements were not available to determine them. The goal for the hardware-in-the-loop simulation was to optimize the slip/slide control for the different conditions that affect adhesion. Though the modeling of the electrical system including the equations of the drives was known, there was no experience in the modeling of this new brake system with all of its various time constants. For this reason, it was decided to incorporate the brake system (including its control logic) as hardware-in-the-loop. The interface between the hardware-in-the-loop and the simulator was chosen to be the brake pressure of the brake shoe. This simulation setup turned out to be very successful. In fact, the controller optimization carried using this simulator was so satisfactory that the commissioning of the Tram did not alter the controller parameters on the vehicle. Based on these test results, a good black-box-model of this brake system was developed for use with other projects.

DIGITAL SIMULATOR

As described in the previous section, analog and hybrid simulators have been used for real-time hardware-in-the-loop simulations for more than ten years. However, a change to a digital simulator offers some major advantages. These are:

- Flexibility
- Modular hardware and software
- Modern programming language
- Lower hardware and maintenance costs

A crucial task is achieving a sufficiently short integration step size and a short dead time for the simulated system. A converter for instance has an estimated dead time of only 25μsec, and a real-time control system (as it is installed on Adtranz locomotives) for motor control has a cycle time under 50μsec. To avoid instability and a loss of accuracy, both times have to be carefully taken into account for the simulation. This leads to special integration methods being applied to handle
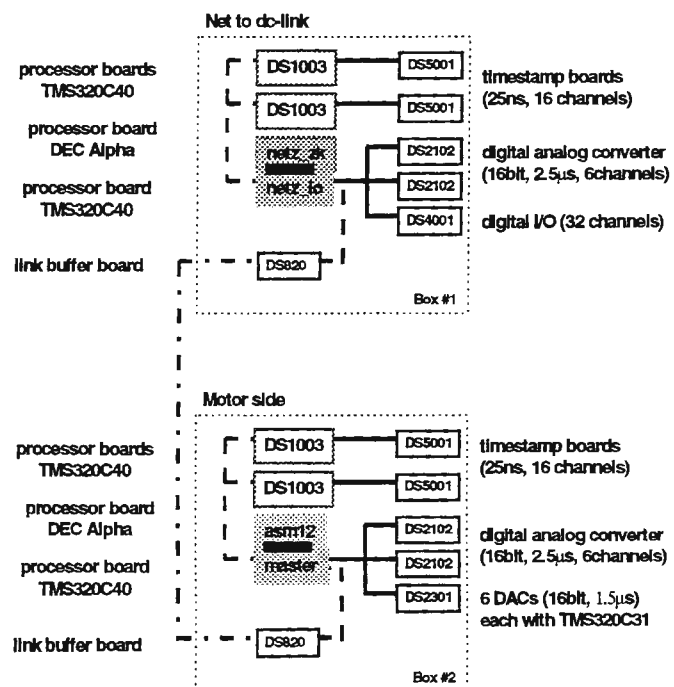
Figure 1: Simulator Topology

the firing pulses of the converter.

Affordable digital simulators with sufficient CPU calculation power to do real-time motor/converter simulations have become available only recently. Nowadays digital simulators are programmable with graphical programming tools, and the resulting models can be downloaded automatically. The German company dSpace offers an integrated development environment [4,5] based on Matlab/Simulink [2,3]. With its latest DEC-alpha processor board the calculation power is high enough to reach step times short enough for our demands.

The graphical programming language used is Matlab/Simulink [2]. In addition to the dSpace being based on this tool, Adtranz already had in-house toolboxs for converter vehicles developed at ABB corporate research centers.

SIMULATION ENVIRONMENT

Simulator Hardware

After benchmarking different configurations, the final hardware architecture shown in figure 1 was chosen. It consists of 2 DEC-alpha processor boards and 6 TMS320C40 digital signal processor boards (DS1003). Semiconductor firing pulses are read with 4 digital waveform capture boards (DS5001) which save the binary input states together with a time stamp in an on-board buffer. Analog outputs are done with 4 high-speed digital to analog conversion boards (DS2102). Communication between processor boards are done with parallel links with a capacity of 20MB/sec between the DS1003 boards. The Alpha-boards are connected to a DS1003 board via dual port memory and communicate via this channel. I/O boards are connected via a peripheral high speed bus with a capacity of 20MB/sec.

Separation of the Model

Due to the high demand for short simulation cycle times it was necessary to distribute the model on two alpha processors (Alpha 1 & 2) which act as the workhorses of the simulation, and on two C40 processors (DSPs) which act as communication managers (see figure 1).
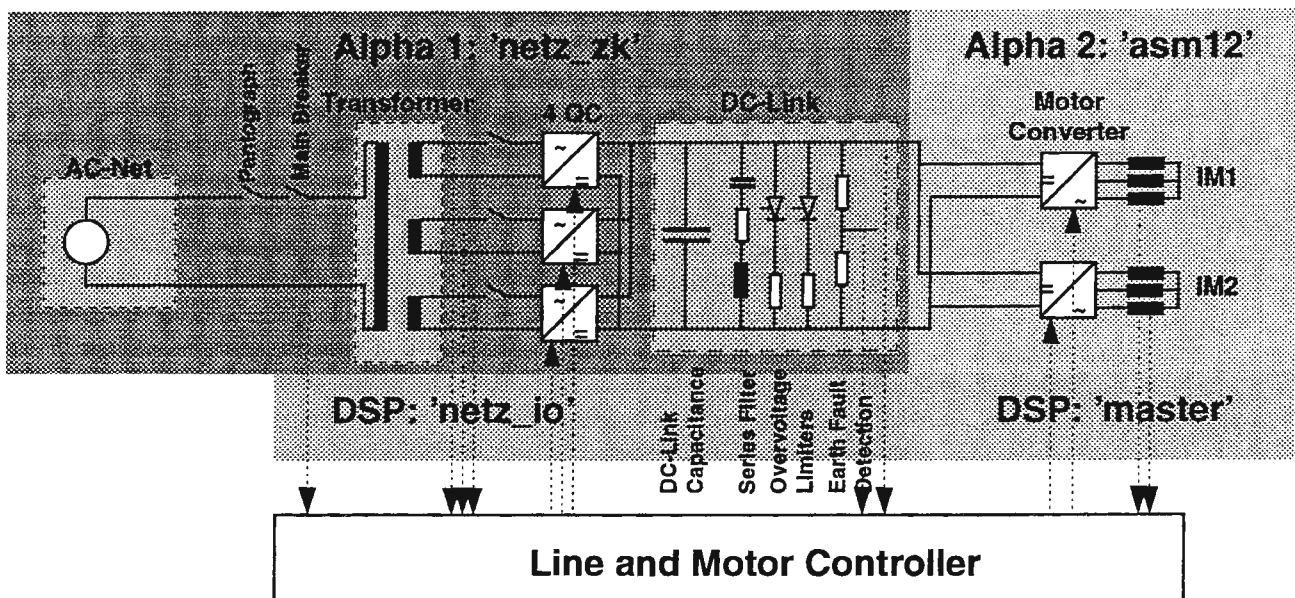
Figure 2: Seperation of the Model

In order to distribute the model on the two alpha processors, the propulsion system model is broken up at the dc link / motor converter interface as shown in figure 2. The left side of figure 2 (Alpha 1) is the line side

submodel which comprises the catenary system, the main transformer, the four-quadrant converters and the dc-link. The right side of figure 2 (Alpha 2) comprises the motor converters, the induction motors and the mechanical model of the train.

The simulation model is programmed using TheMathWorks' Simulink modeling package [2]. The code for the actual real-time simulation is compiled from modules partly implemented in Simulink code and partly in dedicated C code using MathWorks' Real-time Workshop [3], and dSpace's own software [4]. An important advantage for the end-user of such a system (typically a traction engineer rather than a simulation specialist) is the simplicity and flexibility with which he can assemble pre-defined submodels to compose a model of his traction equipment.
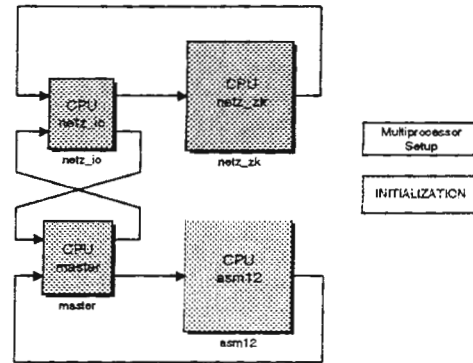
Figure 3: Simulink Model

Most often the interprocessor communication is the bottle neck for fast simulations on distributed systems. Our division of the model of the propulsion system between the dc-link and the motor converters resulted in a minimal communication effort. The communication comprised only the dc-link voltage from the net-side to the motor side and the rectified motor current from the motor side to the net-side.

Figure 3 illustrates now the simulink model is distributed on the processors. This configuration (see also figure 1) allowed us to achieve a simulation step size of only 40usec with a corresponding dead time that was no greater than 1.5 times the step size.

Firing Pulses of the Converters

In terms of the accuracy of the whole propulsion system simulation, the induction motors are the most critical part. Specifically, the firing times of the pulses from the motor converter controller must be accurately determined in order to accurately integrate the input voltages to the motors which in turn are used to compute the stator fluxes. This problem has been solved with the introduction of timestamp boards as shown in figure 5.
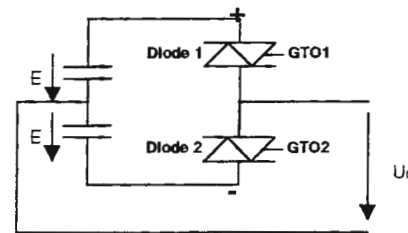
Figure 4: Phase Module
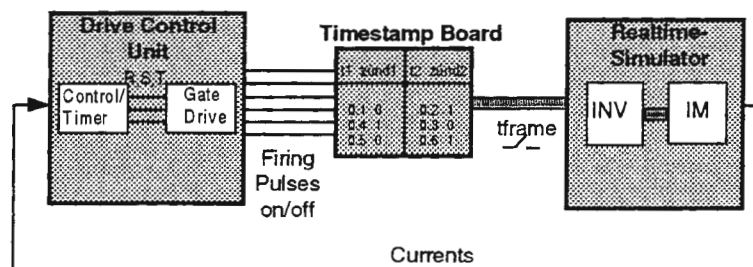
Figure 5: Timestamps

The timestamp boards record the binary value of the firing pulses together with the time when they occured. The time resolution of these boards is 25nsec which is well within the requirements of our application. With this information from the timestamp boards it is possible to determine the exact voltage-time area applied to the induction motor independently of the simulation cycle time. For each phase module of the converter (see figure 4) the resulting information from the timestamp boards consists of a time t_{on}, which indicates how long the positive dc-link voltage was applied, and a time t_{off}, which indicates how long the negative dc-link voltage was applied during each simulation step. The dead time t_{dead_time} of the phase module (neither GTO conducting) is determined by the simulation step time t_{cycle}, t_{on} and t_{off} as $t_{dead\_time} = t_{cycle} - t_{on} - t_{off}$.

489

During t_{dead_time} the current alone determines which dc-link voltage is applied.
The equation for the stator flux of a linear induction motor is

$$\frac{d}{dt}\Psi_\mu = \underline{e}_s - R_s \left(\frac{1}{L_\mu} + \frac{1}{L_\sigma} \right) \cdot \Psi_\mu + \frac{R_s}{L_\sigma} \cdot \Psi_r$$

Ψ_μ may be broken up to the sum of two integrals. The first integral depends only on the applied stator voltages while the second integral depends on the stator and rotor flux. The integral of the stator voltage must be very accurate to guarantee an accurate simulation. The integration of the voltage is done in the 3-phase-system with the time data of the timestamp board. For any single simulation step, it is calculated for each phase as

$$\int_0^{t_{cycle}} \underline{e}_s \, dt = \left(2 \cdot t_{on} - t_{cycle} \right) \cdot E \quad \text{(for positive phase current)}$$

$$\int_0^{t_{cycle}} \underline{e}_s \, dt = \left(t_{cycle} - 2 \cdot t_{off} \right) \cdot E \quad \text{(for negative phase current)}$$

E is one half of the dc-link voltage and varys slowly relative to the integration time step so that it may be taken as constant during any single simulation step.

Simulation of Speed Sensors

In addition to handling the firing pulses, the simulations of speed position sensors is another critical simulation issue. The mechanical sensors considered here put out a set of two rectangular pulses 90° apart. The simulation is required to have
- a pulse resolution down to a few μs in time.
- an accurate displacement angle (typically 90°) between the set of pulses to indicate the direction.
- run asynchronous to the rest of the simulation.
- an output signal that is different from TTL-levels.

These requirements required the use of a separate board called the Direct Digital Synthesis (DDS) board by dSpace. On this board 6 separate processors, each with a D/A converter, are available for C-programming. As cycle times below 5μs are attainable, this board met our requirements. Data is exchanged via dual port memory allowing the speed sensor simulation to run asynchronously to the rest of the simulation. The D/A-converter for each channel gives us the freedom to generate any signal behavior. The limitation is the voltage range of the D/A converter of ± 10V. The interrupt lines of each channel allows a very accurate displacement angle between two channels, which is a crucial detail for certain types of speed sensor logic.

SIMULATION RESULTS

Comparison with Hybrid Simulator

One steady-state operating point in the direct self control (DSC) range has been compared. The comparison between the hybrid and the digital simulator has been done with the same control electronics in closed loop operation. Figure 6 shows the digitally and hybridly simulated motor current for phase r in the time domain. The comparison shows a good qualitative similarity.
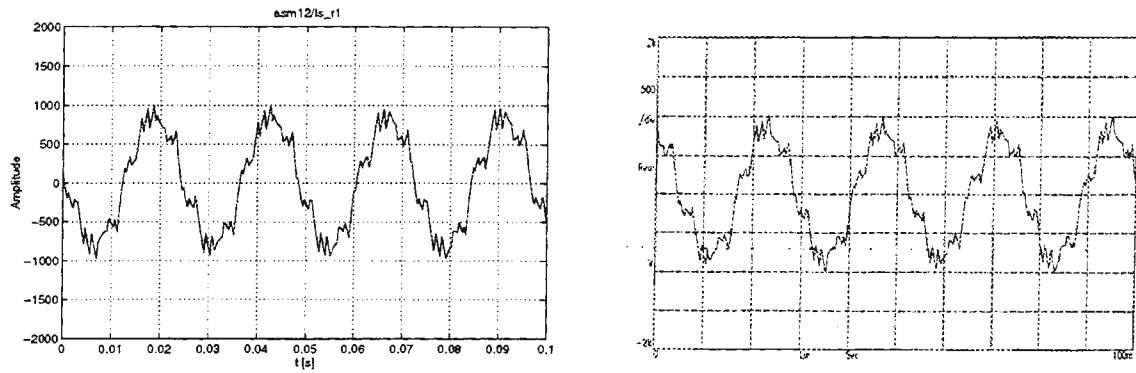
Figure 6: Motor Current for Phase r, digital Simulation (left) and hybrid Simulation (right)

Figure 7 confirms the similarity of the two simulations in the frequency domain. The fundamental amplitude of the motor current is 835Å at 42Hz for the digital simulation and 834Å at 42Hz for the hybrid simulation.
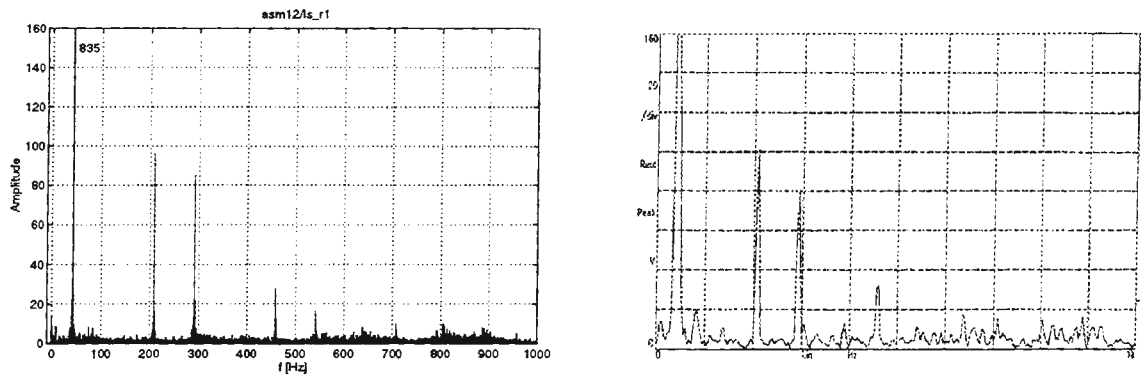
Figure 7: Motor Current for Phase r, digital Simulation (left) and hybrid Simulation (right)

In the digital simulation the amplitude of the 5th and 7th harmonics are 96Å and 85Å, respectively. The hybrid simulation shows slightly different results. The amplitude of the 5th harmonic is 101Å while the 7th harmonic has an amplitude of 81Å. These slight deviations come from the fact that, in closed loop simulation, the operating points are not totally reproducible (e.g. small dc-link voltage variations) on the one hand and on the other hand from the limited resolution of the FFT.

Influence of the Timestamps

Figure 8 shows the stator flux of an induction motor driven by the motor converter. Each figure comprises the nominal stator flux (cyan), the stator flux obtained with the use of timestamps (magenta, dark) and the stator flux obtained without the use of timestamps (yellow, light). The comparison is done for the direct self control (DSC) range, once with full driving torque and once with full braking torque.

In the DSC range the motor converter produces a pulse pattern based on the instantaneous torque which in turn is directly controlled on a hexagon and held within a tolerance band. This results in a pulse pattern that cannot be specified a priori since the instantaneous torque is needed to determine the switching times. The stator frequency of the induction motor is around 10Hz. A time window of 1 second is displayed.

491
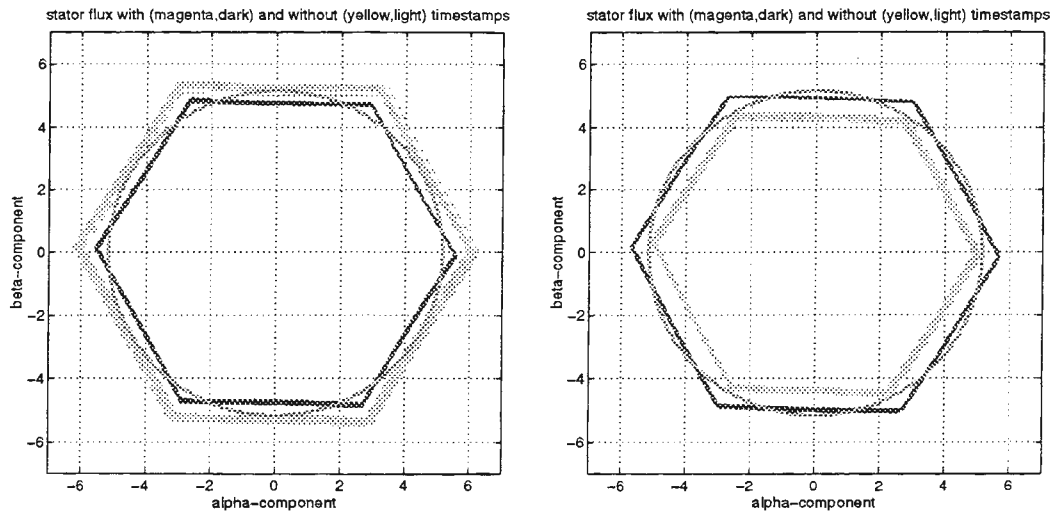
Figure 8: Stator Flux at 'Full Driving Torque' (left) and ' Full Braking Torque' (right)

In the case without timestamps, the amplitude of the stator flux varies depending on the operating point contrary to the operation of the actual system. The use of timestamps however provides a much more accurate result and, in particular, the flux does not depend on the operating point.

CONCLUSIONS

We have presented selected results from a digital real-time simulation application. Particular emphasis has been placed on the comparison between digital and hybrid simulation and on the use of timestamp boards. The comparison shows that a digital real-time simulator can replace a hybrid simulator with no loss of accuracy even in applications that have short simulation cycle times. It was shown that the use of timestamp boards were critical to achieving short simulation cycle times with no loss of accuracy.

REFERENCES

1. Terwiesch et al, in: O. Maler (ed.), Hybrid and real-time systems, Springer Lecture Notes in Computer Science, pp. 199-210, 1997.
2. The MathWorks Inc, Simulink User's Guide, 1992
3. The MathWorks Inc, Real-Time Workshop, 1994
4. dSpace GmbH, Real-Time Interface to Simulink, Version 2.1
5. dSpace GmbH, Software Environment for Modular Systems, Manual 2: Multi-Processor Systems, Version 1.0

SIMULATION-BASED DESIGN OF MODULAR ASSEMBLY SYSTEM-USE OF SIMULATION MODULE LIBRARY

J.Heilala, J.Montonen

VTT Manufacturing Technology, P.O.Box, 1702, FIN-02044 VTT, FINLAND Juhani.Heilala@vtt.fi

Abstract

In the EUREKA research project FAMOS-HIPS, a new modular assembly system was developed. Factory simulation tools were used in the development work, marketing and assembly system design. Graphical simulators were used to visualize ideas and improve communication while creating the HIPS Assembly System Concept. This paper introduces a software tool for the quick modelling, analysis and design of a modular assembly line. The authors review a set of submodels developed in the graphical 3-D simulation package, which gives the user the capability to rapidly model an assembly line while still providing access to a more general simulation environment for model customization and enhancement. Large and complex models can be built using submodel merging. Three-dimensional computer animation based on the real model can be used in the concurrent design process and in the training of operators and managers. Keywords: Assembly system design and simulation, quick modelling, reusable models

Introduction

This paper introduces a modelling tool for the quick modelling and analysis of the flexible final assembly line. A framework for rapid model creation and reuse is being developed using the currently available 3-D graphical simulation product, QUEST [1,2,3]. A similar development has been made for other 2-D simulation products, such as WITNESS and ARENA [4,5,6].

The authors review a set of submodels developed in the 3-D graphical simulation package, which gives the user the capability to rapidly model an assembly line while still providing access to a more general simulation environment for model customization and enhancement. This tool requires less familiarity with simulation construct, allowing it to be used by the industrial and tooling engineers typically involved in production line design and modification. It is a useful tool for the system sales engineer as well.

The need for rapid model development is challenging because production lines are in a constant state of flux due to fluctuations in demands, annual design changes, and the introduction of new processing technologies. The ever-changing design process requires frequent, rapid (i.e., a few days or weeks) evaluation of system changes ranging from simple parameter modifications (i.e., new cycle times) to total line configuration. A model developer must be able to create accurate, realistic models in a short space of time. The second challenge faced by the simulationist is that there is a number of groups in the enterprise that could benefit from the information available from simulation models.

The real system, modular assembly system hardware

In the EUREKA research project, FAMOS-HIPS, - Human Integrated Production System for Assembly and Disassembly, a consortium of manufacturing companies, assembly system suppliers and research institutes from five different countries have developed a new type of modular assembly system for lightweight products, including mobile phones, consumer electronics, and car components [7,8,9].

The core of the HIPS Assembly System consists of manual and automatic workstations equipped with automated material-handling (i.e. pallet flow) and intelligent control features. The intelligent pallets are guided on the conveyors of the workstation module or other transfer system module according to the routing and product information which is carried in escort memory. The modules of the system can be configured in different layouts according to the production needs. The module control unit controls actuators and motors by reading a sensor system of the module and information from the intelligent pallet.
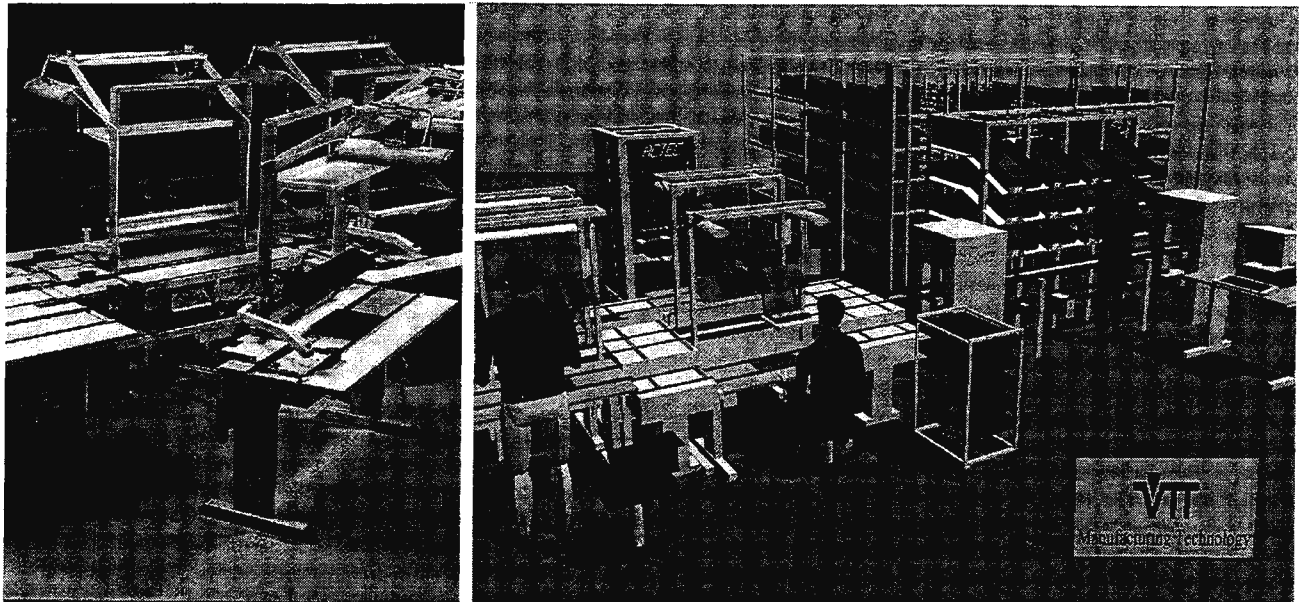
Figure 1. The modular structure makes it easy to expand or modify the assembly system. A module can be changed in 15 minutes, making the changes fast. The HIPS Assembly System is a plug and product system.

Use of simulation in manufacturing engineering

The advantage of simulation as a design and analysis method is clear. The modification and alternatives can be studied using computer software before anything is built or changed. An assembly-line model in the memory of a computer is a test bench, virtual system, where the effects of changes can be studied. Traditional uses of simulation go beyond providing a look into the future. These include the following [10]: better understanding, exploring possibilities, diagnosing problems, building a consensus and training the team.

Simulation tools

There are several commercial simulation tools available [4,11,13]. These tools can be divided into three basic classes: general purpose simulation language, simulation front-ends and simulators. The general purpose simulation language requires the user to be a proficient programmer as well as competent simulationist. The simulation front-ends are essentially interface programs between the user and the simulation language being used. The simulators of today utilize constructs and terminology common to the manufacturing community, and offer graphical presentation and animation.

The discrete event simulators are well-suited for the simulation of a production system [13]. Simulators may exceed the capabilities of the average manufacturing engineer, thus requiring a dedicated programmer/analyst. There are often specialized situations and scenarios that are outside the available set of modelling constructs (i.e., objects), where the user must often resort to more intensive programming to complete the model. Recent developments in simulator packages, especially in the ability to define submodels (reusable model segments), provide them with the flexibility to meet the needs of the development effort. The newest versions of simulator packages have a graphical user interface (GUI), and the use of a mouse brings advantages, i.e., drag-and-drop techniques, to the model development [14]. The computer platform is usually PC Windows/NT or a graphical Unix workstation.

Quick modelling

There is a need to speed up simulation projects. One of the challenges is the shortening of the modelling time. There are different ways to speed up simulation modelling:

1. *Reference Models*: A complete set of model structures together with a description, how these structures apply and how they can be adapted to a given problem. A reference model is not an ideal solution used as a measure, but a typical structure for a special type of problem.
2. *Simulation module library*. Hierarchical modelling allows the user to save whole models as clusters, groups that can be deleted, moved, or scaled as a single object. Groups or clusters can then make up sections of a larger, complex model. Groups, i.e. submodels can be pre-engineered, standardized solutions.
3. *Application Solution Templates* (AST). Industry-specific templates allow customization of the software so the user can automatically start up with specific icon libraries, functions, element terminology, element types, and other industry-specific settings. AST can be a highly graphical, easy-to-use simulation system that allows rapid modelling of manufacturing systems. AST could be, for example, a spreadsheet-like interface to fill in the needed information.
4. The integration with other software tools, like CAD, spreadsheets and databases. It is important to be able to use existing information stored in computers.

The aim was to speed up the simulation model building and make it easier. The solution was to develop reusable simulation models, i.e. create a simulation library of the HIPS Assembly System modules. The HIPS Assembly System concept is a reference model. The speed advantage was estimated to be ten-fold compared to the traditional way of model building. The software provides possibilities for using the template-based user interface in the future as well. The template interface could be programmed using Visual Basic or Visual C in PC/NT platforms, as shown earlier. [12].

Development of reusable simulation models

The modular and intelligent assembly workstation system and transfer system modules have been modelled during the FAMOS-HIPS project. The factory simulator, QUEST, was chosen as the development tool, because of the advanced 3-D visualization capabilities. The development was made with QUEST v2.1 on a Silicon Graphics workstation. The development work was based on the use of a group, a new feature added to the software in 1995. A group, (i.e., submodel) can be merged to a model, thus large and complex models can be built by just using the graphical user interface, without programming.

Material flow connections between submodel resources are defined in the module when creating the simulation module group. The waypoints of conveyors, stackpoints of buffers and workpoints of machines are defined as well.

Functional resource parameters are defined in the submodel (*capacity, conveyor speed, etc.) and part of the parameters can be changed using the graphical user interface with the mouse (for example, process cycle time and statistical distributions).

The generic logic for a resource is defined in the logic library. The user can also create new logic while creating the simulation model. A simulation resource can have several logics, for example, for routing a pallet in the crossing units, route logic, and for processing the assembly task, process logic.

Some of the task-related parameters of the workstation or transfer module can be defined by changing the resource name or by selecting a new logic program from a logic library. Figure 2. shows a blow-up of one of the HIPS Assembly System manual workstation simulation models. Figure 3. shows how to use the visual file interface in building an assembly line.

Simulation-modelling elements that capture the performance characteristics of individual pieces of equipment or groups of equipment have been developed. The intelligent HIPS Assembly System hardware modules have been modelled using the features of chosen simulation software.
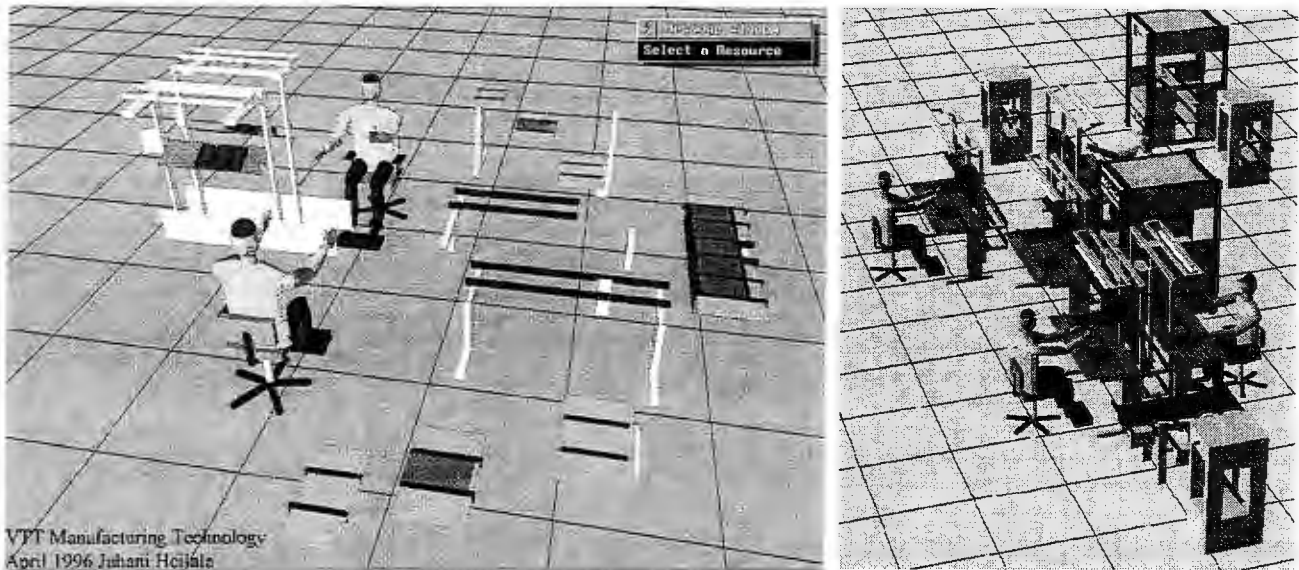
Figure 2. A simulation submodel has several resources, each having its own geometry and logic. The simulation model is built from submodels (plug and simulate).

Use of simulation submodels

The standardized simulation modules are stored in a library. Each of the library modules is actually an independent simulation model. When the visual file interface is used, model builder has a visual feedback from the contents of the file (Figure 3.). The simulation module is a group of independent simulation resources, which can be read to a simulation model. The group and the resources belonging to it can be transferred and rotated as a single-simulation element in the simulation model.

There are only a few simple steps for building a simulation model using the simulation module library that has been developed. The production engineer follows basically the same steps on the factory floor while building the system from intelligent system modules: arrange the modules in the right order, make the mechanical connections, connect the energy supply and optional information cables, set up the parameters for each workstation and start production (plug and product).

Building the simulation model using library elements, submodels:

1. Layout; first the user reads the submodels from the library and transfers them to the right place in the model.
2. Connections; the user makes the connections between the existing and merged submodels to ensure the right material flow in the model.
3. Parameters; the user sets up all the parameters needed for the assembly process and material flow (cycle times, routing information)
4. Create the other environment and connections to and from the HIPS Assembly System (create the other resources and special logic)

The three first steps can be carried out using the batch control language (BCL) of QUEST and a template interface to set up the parameters. The standardized building blocks of the assembly system provide possibilities for using BCL macros and further speed up the modelling. The user has access to a more general simulation environment for model customization and enhancement if needed, i.e. when a new submodel has to be created.
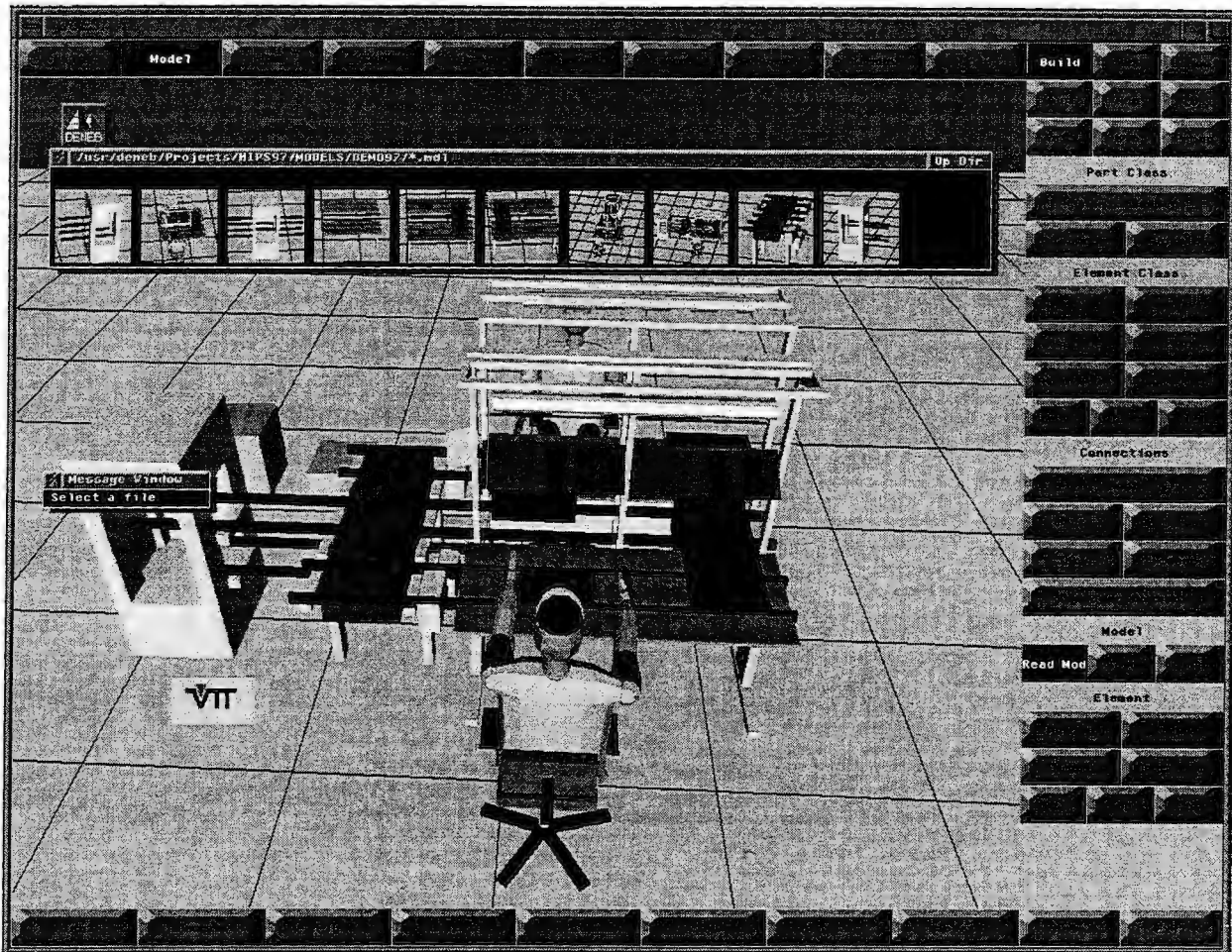
Figure. 3. The visual file interface helps to select the right submodel from the library. The submodels are pre-engineered standard solutions. Different variations of the workstation are stored in different files.

Current status

The goal was to establish capabilities for rapid model development in the simulation system (i.e., QUEST) and make submodels for the transfer and workstation modules of the HIPS Assembly System. The developed system is currently being evaluated. The preliminary reaction has been positive and more extensive evaluation is underway. Initial results indicate that the use of a simulation module library can substantially reduce simulation model development time. The new release of QUEST v3 (1997) offers better possibilities for continuing the development work. Firstly, though, the developed submodels must be converted for the new release of the simulator and new submodels created. The development work will continue in new projects.

Summary

The benefits of this approach are many. The reduction in the complexity of the model development and specification should result in an increase in the use of simulation in the design and development of new assembly production lines. The creation and use of submodels that capture the specialized knowledge of machine operations speeds up the entire simulation development process. In addition, the use of submodels allows the creation of reusable portions of simulation code, which significantly reduces the time required to develop and/or reconfigure models in the future.

Even with a limited number of modules a large assembly system can be built. The simulation software used in this development work gives great potential for this kind of modular, standardized approach. The

developed simulation library could be used by assembly system sales or design engineers, with very short simulation training. The use of a graphical user interface shortens the learning time and a little or no programming skill is required, while the model is built from standardized modules.

The partners in the Eureka FAMOS-HIPS project were VTT Manufacturing Technology, GWS Systems Oy, Nokia Research Centre, Actec Engineering Oy, Helsinki University of Technology, Festo Oy Finland, Festo KG Germany, Almet Ltd of Turkey, Bang&Olufsen A/S of Denmark and Kongsberg Automotive ASA of Norway. The work of VTT Manufacturing Technology was funded by the Technology Development Centre of Finland (TEKES). The development work continues in new projects called Semiautomatic Final Assembly and Information System for Assembly and Disassembly. More information from Juhani Heilala, VTT Manufacturing Technology.

References

1. Heilala Juhani and Montonen Jari. HIPS Simulation Module Library for Assembly System Simulation. Proceedings of Deneb Robotics Inc. User Group Meeting In Finland. Tehdasmallit Oy. 6.-7. June 1996, Espoo, Finland,. p 7.
2. Juhani E. Heilala , Jari P. Montonen and Paavo K. Voho. Developing Reusable Simulation Module Library for Assembly System Simulation. Proceedings. The 14th International Conference on Production Research. Volume 1. August 4-8,1997, Osaka Institute of Technology & Setsunan University, Osaka, Japan. p 250-253
3. QUEST 2.1. QUEST User Manual and Tutorials. Deneb Robotics Inc., Auburn Hills, Michigan USA. 1995.
4. Farington, Phillip A., Swain, James J., Evans, John L. and Rogers, John S.. Developing Reusable Modelling Capabilities for Simulating High Volume Electronics Manufacturing Systems, IEEE Transactions on Components, Packaging, and Manufacturing Technology- Part C. Vol. 19, No 2, April 1996. pp. 89-97.
5. Mazziotti, Barbara Werner. Get More Mileage From Flexible Simulation. IIE Solutions. Vol. 28 No 5. May 1996, pp. 14-22.
6. Mertins, K., Rabe, M, Muller, W; Ohle, F., Reference Model manufacturing Systems - A New Approach for More Efficient Simulation Studies. Proceedings. The 14th International Conference on Production Research. Volume 1. August 4-8,1997, Osaka Institute of Technology & Setsunan University, Osaka, Japan. p 254-257
7. Paavo K. Voho and Juhani E. Heilala. Flexible and Hybrid Assembly System Concept for Light Assembly. Proceedings The 14th International Conference on Production Research. Volume 1. August 4-8,1997, Osaka Institute of Technology & Setsunan University, Osaka, Japan. p 66-69.
8. J.Heilala, P. Voho, Human Factors in the HIPS Assembly System Concept. IEA97 Proceedings of the 13th Triennial Congress of the International Ergonomics Association,. June 29 - July 4. 1997., Tampere, Finland. Volume 1, Organizational design and management. Finnish Institute of Occupational Health, Helsinki 1997, p 196-198.
9. Voho Paavo and Heilala Juhani. Flexible and hybrid production system for assembly. Proceedings of the 5th International Conference on Human Aspects of Advanced Manufacturing: Agility and Hybrid Automation, Maui, Hawaji, USA, August 1996. IEA Press 1996, Louisville, Kentucky, USA. p. 443- 446.
10. Banks, Jerry, Norman Van and Spearman, Mark. A Second Look at Simulation Software, CiME, Computers integrated Manufacture & Engineering. Vol. 1, No 2. 1996. pp. 48-50.
11. Anon. Simulation Software Buyers Guide, IIE Solutions, Vol. 29 No. 5 May 1997, pp. 64-78.
12. Morgan,S. (1994). Automatic generation of QUEST Simulation Models. Proceedings of the Deneb User Group Conference, 1994.
13. Klingstam, P and Gullander Per. Overview of Simulation Tools for Computer Aided Production Engineering. 1997 Advanced Summer Institute - ASI'97, Budapest, Hungary. July 1997. 9 p.
14. Anon. PS-Engine - Plant Design TM. Discrete Event Simulation and Plant Design. A Prosolvia Systems AB White Paper. January 1998.
15. Anon. Quest v3. Deneb Robotics, Inc. 3285 Lapeer Road West. P.O. Box 214687. Auburn Hills, MI 48321-4687

Spatial Planning of Manual and Automated Work Stations Utilising Mathematic Algorithms and Virtual Reality Devices

[1]Prof. Dr.-Ing. Gunther Reinhart [2]Dipl.-Ing. Ulrich Rossgoderer

[1]Institute for Machine Tools and Industrial Engineering (iwb), 85747 Garching, Germany, re@iwb.mw.tu-muenchen.de
[2]Institute for Machine Tools and Industrial Engineering (iwb), 85747 Garching, Germany, rg@iwb.mw.tu-muenchen.de

Abstract

The system described in the following article, simplifies the task of spatial planning of work stations in manufacturing. On the one hand this is done through automation of iterative planning tasks (i.e. generation of layout variants). On the other hand virtual reality devices are used in cases where it is more efficient to include the planner's intuition (i.e. generation of human movements). Applications of this planning method will be shown in the article. One application shows the spatial planning of an automated assembly work station, the other deals with a manual work station. Furthermore the virtual reality equipment, both software and hardware, we are working with at the iwb will be described briefly.

Introduction

The goal of our work is to simplify the task of generating a 3D simulation model of production systems. 3D simulation models are widely used in production planning for calculation of cycle times, for determining work spaces, digital mock-up and so on. There are two major trends in computer aided spatial planning at the moment. People following the one trend are trying to highly automate generation of three dimensional layouts and processes. They are achieving this for example through intelligent algorithms for automated grasp planning /5/ or automatic off-line programming of cutting processes /2/. Approaches that automate the layout generation process are known, too. In /11/ we are describing our method to optimise production systems. Other systems like /8/ and /9/ automate both, the generation of layouts and the generation of robot programs. The advantage of the automated planning approach is that once you have defined your planning target you can be sure that the algorithms will reach the target with a minimum of interaction. The disadvantage is that you have to formalise every single planning constraint (i.e. maximum amount of space for the work station). This can be very difficult or even impossible.

The second trend is caused by the development of virtual reality devices. There exist many systems as described in /1/ and /3/ which are utilising devices like data gloves in order to simplify the interactive planning of production systems. In those systems you can easily rearrange production resources or do programming by just moving a robot's TCP using a six dimensional input device. The big problem in using this kind of intuitive planning is that your intuition is not always right with respect to your planning targets. Imagine a manual work station. It is definitely not easy for a planner of a certain height to design a work place for a worker of a different height considering the ergonomic situation. Also a well defined algorithm for automated robot path planning will surely result in a faster movement than teaching the path manually.

The system we have developed and will be presented in the following facilitates a useful combination of automated and intuitive planning action. The basic principle is to combine mathematic algorithms with efficient input and output devices in order to provide fast and reliable planning results. The system supports the three main steps in spatial work station planning:

- Initial Layout Generation,
- Generation of Movements and
- Optimisation of Arrangement and Movements.

Both automated and manual work stations are taken into account. This is very important as in practice in product assembly you will always have a mixture of manual and automated work stations.
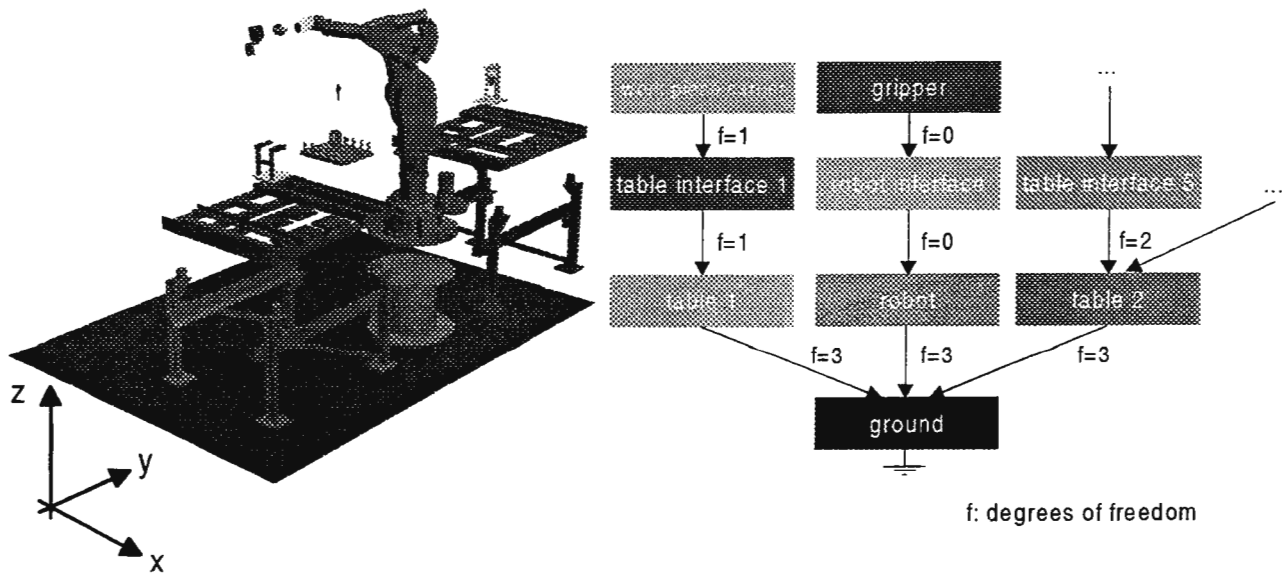
Figure 1. Structural description of an assembly station

Initial Layout Generation

In order to create a 3D simulation model of an assembly station, the engineer first establishes a rough arrangement for each production resource. The information about possible geometrical relationships between two or more components is principally fixed by the designer. He has the knowledge how to place different components to each another. His knowledge is represented in the CAD model by the definition of constraints. Another way of representation are relations between product parts in the products assembled state and relations between product parts and other components of the assembly cell.

Our way of describing those relationships between work station components is shown in Figure 1. We are separating the information about the structure of the layout from the spatial arrangement. The structure can be imagined as a tree which root is the ground of an assembly station. Its nodes represent the production resources and its branches are representing their relationships (i.e. "table stands on the ground"). In order to generate a first version of a cell layout our algorithm is able to interpret this information and then fit together two components in a correct way. This description does not directly fix a specific location between two parts, but specifies a series of degrees of freedom as shown in Figure 1.

The algorithm is able to place the components to each other using its extended CAD-model. By placing components with respect to the components remaining degrees of freedom a search algorithm now tries to find a collision-free arrangement with a fast collision detection. Step by step, the assembly cell is build up by searching the cell hierarchy and placing each single component. Result of this automatic planning step is an initial "reasonable" arrangement of all required components, which does not take account of the handling process so far. After the automatic arrangement the planner is free to rearrange components. Rearrangement is limited to the remaining degrees of freedom. For example the robot can only be moved on the ground and rotated around an axis normal to the ground. As an input medium, traditional devices like mouse or dials can be used. In a more sophisticated environment the planner is able to use data gloves or 6D pens. In both cases the limitation of the degrees of freedom is helpful, because the planner can't create an "unreasonable" arrangement. This planning step is independent from whether you are planning an automated or a manual work station.
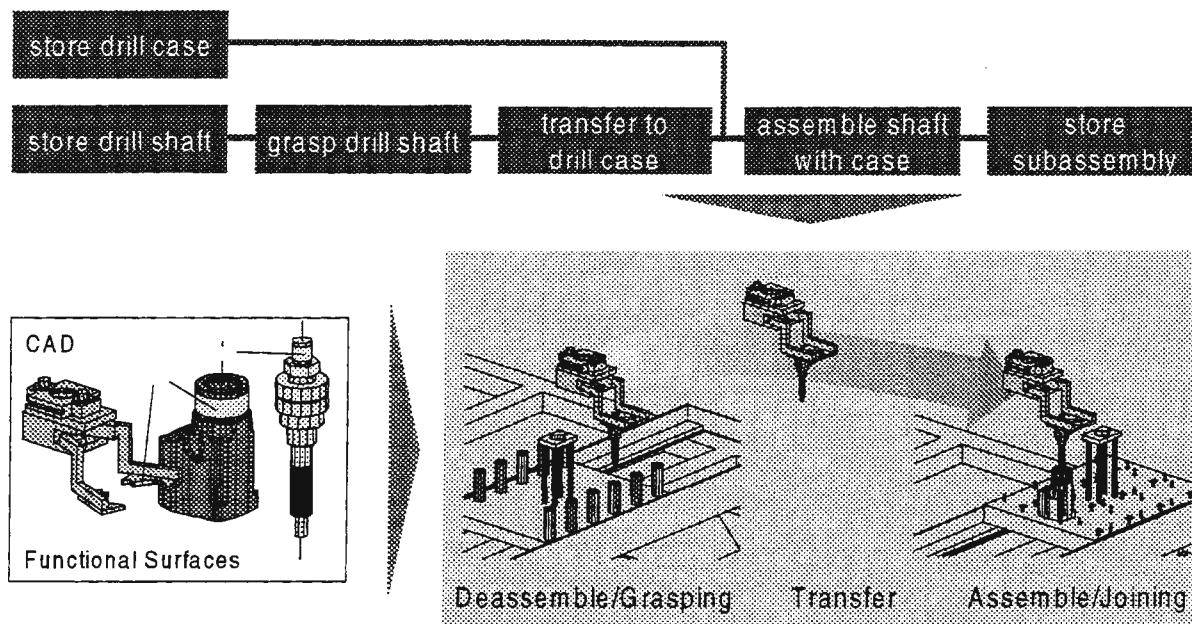
Figure 2. Task oriented offline programming

Generation of Movements

The next step in 3D simulation model generation is to create spatial movements of robots and workers. This time we distinguish between automated and manual work stations. For automated work cells we are generating programs by using task oriented offline programming (Figure 2). As an input we take a robot independent description of all product specific movements. For every important task of the assembly process we have designed solvers that produce the robot specific code /10/. Supported tasks are:

- Grasping,

- Transport,

- Fetching from/Putting to a part storage and

- Fixing.

Similar to the initial layout generation extended CAD data are used to determine spatial relationships for robot movements. The grasp planning module for example uses surfaces defined earlier in the CAD system in order to position the gripper relatively to the part to be gripped.

For manual work stations we are generating the movements of the worker manually. Therefore we are using a model of the human body consisting of 60 joints (see Figure 4). Programming the movement is done by using generators for walking and side steps and inverse kinematics for both arms and legs. The basic functionality is identical to the commercial available AnySIM/Man, also licensed to the company TECNOMATIX under ROBCAD/Man. This functionality includes the capability to analyse the manual movement with respect to the Methods Time Measurement (MTM) and Ergonomics. We will discuss those evaluation functions in the section "layout optimisation".

The great difficulty at the moment in generating the manual movement is, that the planner is limited to using keyboard and mouse. Furthermore it is not possible at the moment to use the task oriented programming approach as used for robot movements. That is because a human has much more degrees of freedom in his movements than a robot. Just think of assembling parts using your both hands. In this case it seems much more appropriate to use the intuitive approach. That is what we are doing. In order to teach the movement of the worker's hands into the simulation system, we are using data gloves combined with a tracking system as input devices for the left and right hand. After putting on the gloves you can operate the simulated human just

as you would do the real assembly. You are able to grasp and assemble the virtual product parts. However, several problems had to be solved to make this working correctly. One thing is that we have to detect collisions whenever the virtual hand has moved. After detecting a collision we have to decide whether the collision leads to a grasped object. This could be the case for example, if you touch an object with two or more fingers. We have implemented a first prototype, but are still working on this problem. Another question you have to solve is: is the inverse kinematics of the human model suitable for this input method? In case of the human model we are using, the answer is "basically yes". An application for a middle-sized company showed us, that it works fine for basic movements in usual grasping positions. Problems occur when grasping objects far away from the centre of the human model's body. To solve those problems, the kinematics of the human model has to be improved, which is what we are discussing with the company AnySIM that designed the model /6/.

Layout Optimisation

The last step in generating a three dimensional layout of the assembly system is called the automatic layout optimisation. Parameters for the optimisation run are the coordinates of production resources and the model of the human. Several target criteria can be weighted and combined to create an optimised assembly system. The measurement of every single target criterion takes place in the simulation system by doing a movement simulation of robots and humans based on the programs generated in the previous step. To simplify the optimisation problem it is subdivided in several optimisation runs. We take advantage of our knowledge about the structure of the assembly station and do a layout optimisation for each node of the layout tree (Figure 4). For each node, the components represented by its following nodes are rearranged using their remaining degrees of freedom. One optimisation run consists of five automated steps:

1. Rearrange Components,
2. Adopt programs of robots and humans to new arrangement,
3. Simulate programs of robots and humans,
4. Assess the arrangement and
5. Feed the assessment into an optimisation algorithm which generates the next rearrangement.

Differences in treatment of manual and automated work stations are made in steps two and five. Whereas programs of robots can be adopted automatically, movements of the human eventually must be adopted manually. The latter depends on the behaviour of the inverse kinematics of the simulated human (see above).

Possible optimisation goals of automated systems amongst others are minimisation of cycle times and driving torque. Criteria for the design of manual work stations are duration of movements and the ergonomics of the configuration. These criteria are measured in the simulation system using the Methods Time Measurement analysis (MTM) for calculating cycle times /4/. MTM is a widely accepted standard for determining duration of human motions. In order to apply this method the human motion is subdivided into basic movements (i.e. fetching, grasping). For each basic movement it is possible to look up in a table how long the movement should take depending on the spatial length of the movement. Those tables are included in the simulation models and therefore the MTM can be processed automatically by the computer. A second important goal in optimising manual work stations is to improve the ergonomic situation. For that we are using the Ovako Working Posture Analysing System (OWAS) /7/. Similar to MTM it is based on tables which show assessment of a working posture with respect to the weight to carry and how often and how long the action resulting in this posture has to be done.

Based on the assessment done automatically in the simulation system, an appropriate optimisation algorithm creates many different layouts and measures the fulfilment of the target criteria for each arrangement. As a result the planner gets the optimised layout together with time measurements, ergonomic analysis and robot programs that can be used for production later. Figure 4 shows an arrangement of a manual work station before and after the optimisation run. Target criteria were equal weighted MTM and OWAS analysis. The improvement in criteria fulfilment was 18% with respect to the initial layout.
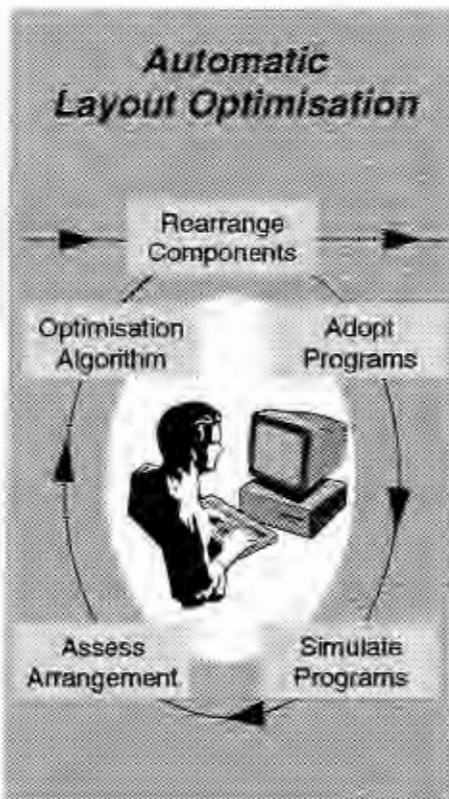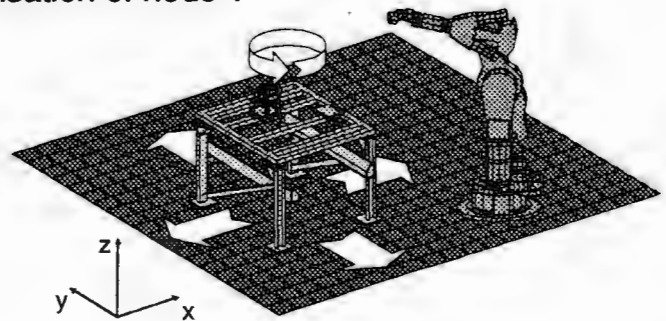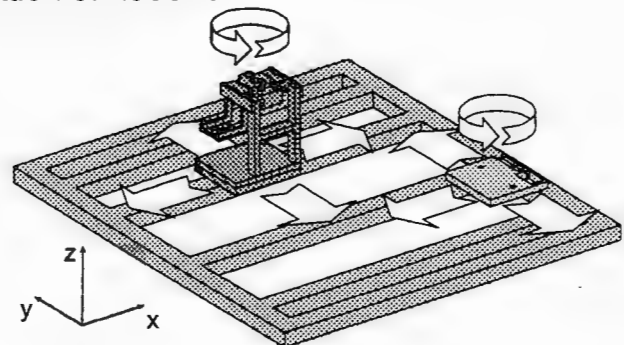
Figure 3. Optimisation of automated and manual assembly stations
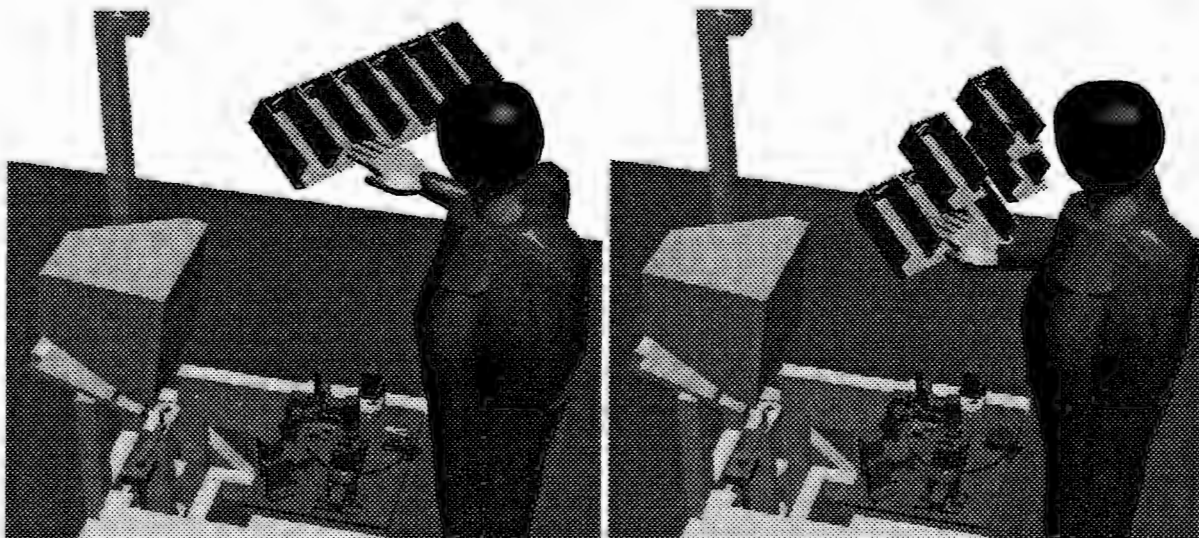
Figure 4. Optimising a manual work station (left: initial, right: optimised arrangement)

Virtual Reality Components

After discussing how our system leads the planner through the three planning steps, we briefly describe the virtual reality components of the system. The virtual reality room at the iwb consists of a stereoscopic projection system called "Dual Workbench". As shown in Figure 5 it consists of two video projectors displaying images generated from a high-speed graphics computer onto two projection planes. Using shutter glasses you get a very good spatial impression of the virtual environment, because, when standing close to the bench, a very large part of your field of view is covered by the projection,. In order to interact with the scene,

we are using a pair of data gloves. Tracking of hands' and head's positions is done using a magnetic tracking system.
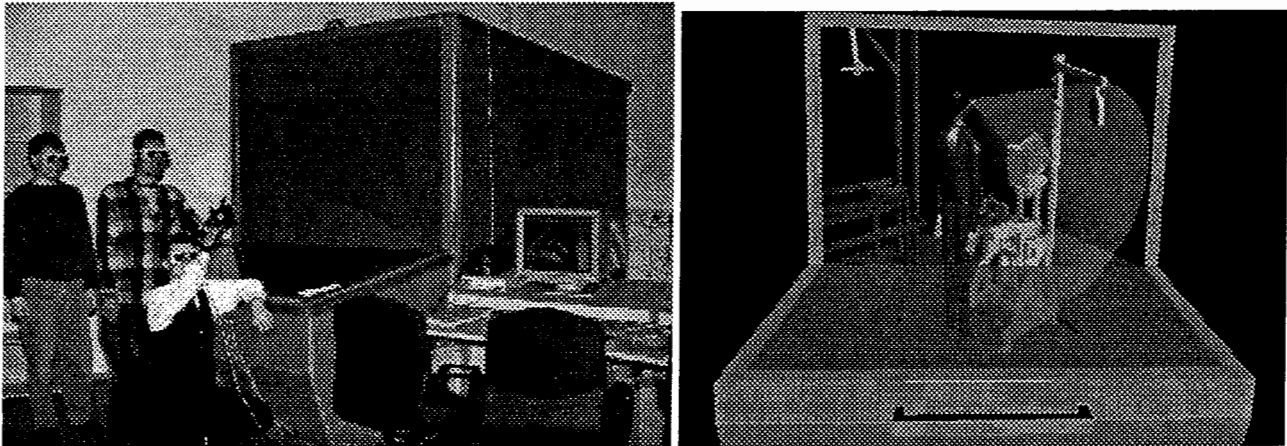
Figure 5. The virtual reality room at the iwb (left: the real, right: the virtual dual workbench)

The key component naturally is the software, which drives the projection system and handles input data from the gloves and the tracker. Therefore we extended our 3D simulation system USIS /10/, which is based on the product AnySIM mentioned before, to creating a stereoscopic view of the virtual environment. The view depends on the planner's spatial position, which is essential to produce a correct perspective on the dual workbench. We also extended USIS to enable interacting with objects in the scene using the data gloves. Therefore a fast collision detection algorithm is used in order to enable "realistic" grasping. At the moment it is possible to identify objects using your index finger, to grasp and move objects using a combination of fingers depending on size and shape of the object in your hand. Gravity or forces are not taken into account yet. But we are currently working on this thing.
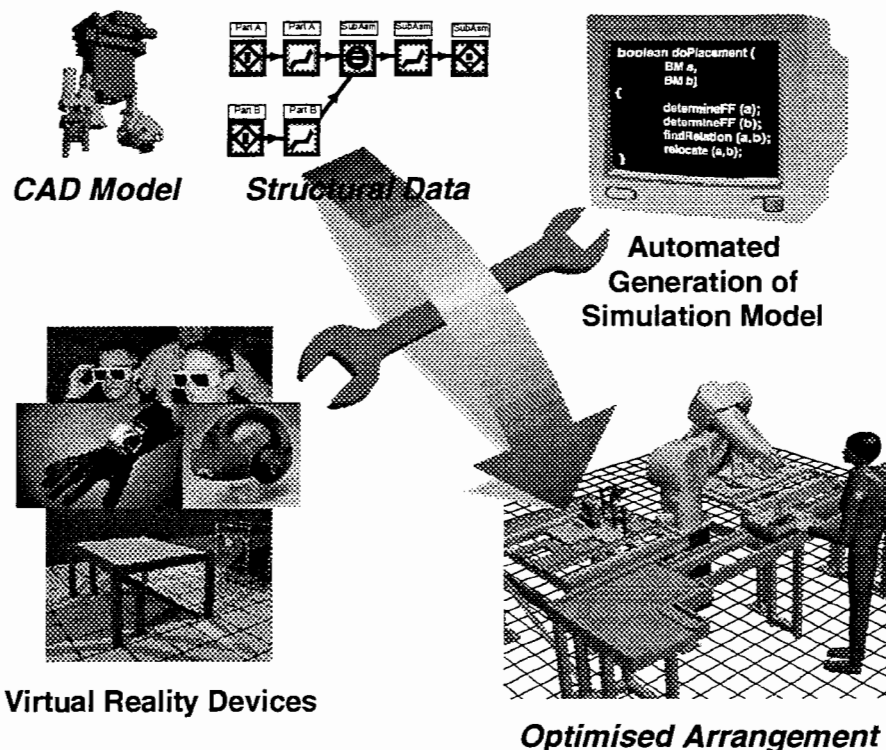
Figure 6. Generating an optimised arrangement using VR devices and model generators

504

Discussion

This paper has presented an approach to combine automated and interactive spatial planning of work stations, in order to make the planning result available more faster while providing a higher reliability (Figure 6). Both manual and automated work stations were taken into account. Challenges we met were to decide when it is more efficient to automate a planning task or to do it interactive. Our implementation of the system contains both code for the automated planning steps and code to support virtual reality devices.

Based on structural data from preceding planning steps and based on extended CAD models, the system is able to generate an initial layout automatically. Generation of spatial movements is done using task oriented offline programming for robots. Human motions are generated using virtual reality equipment, especially data gloves. Improvement of the initial arrangement with respect to certain target criteria is achieved by applying an optimisation algorithm. Target criteria are based on measurements taken during a simulation run. Measurements include MTM and ergonomic assessment using the Ovako method.

Future work will deal with improving interaction with virtual objects. In order to achieve a more realistic "feeling" during assembly for the one who interacts, it is necessary to consider gravity and forces. Once this is solved we can use a haptic interface to provide force feedback. We also want to provide the evaluation functions mentioned above in interactive mode, that is during the planner is rearranging cell components manually. This means, that we're replacing the optimisation algorithm by the planner's experience and intuition which sometimes may lead to a result faster than formalising constraints and doing the optimisation run. Through representing the evaluation results online in a suitable manner (e.g. visual or haptic feedback), we will prevent the planner from doing anything "unreasonable".

References

[1]   *Bauer, C.*: Virtual Reality in the German Speaking Area and the Consideration of the European Circumstances. Proceedings Virtual Reality World 95. Stuttgart, February 1995, pp. 21-30.
[2]   *Bickendorf, J.*: Full-Automatic Off-Line Programming of Complex Cutting Paths. Proceedings of the 25th International Symposium on Industrial Robots, Hannover (1994), pp. 499-506.
[3]   *Dänghaus, R.; Neugebauer, J.-G.; Schraft, R.D.* : The Programming of Automation Systems with Virtual Reality. Proceedings Virtual Reality World 95. Stuttgart, February 1995, pp. 83-87.
[4]   *Deutsche MTM-Vereinigung e.V. (Editor)*: MTM-Handbuch I (Grundlehrgangsunterlage). 7. Aufl. Hamburg: Deutsche MTM-Vereinigung e.V. 1990.
[5]   *Dini, G.*: A Module for the Automated Planning of Grasps in Robotized Assembly Operations. CIRP Annals (1993), Vol. 42/1, pp. 1-4.
[6]   *Geyer, M.*: Flexibles Planungssystem zur Berücksichtigung ergonomischer Aspekte bei der Produkt- und Arbeitssystemgestaltung. Berlin: Springer 1997. Dissertation TU München 1997.
[7]   *Härkönen, R.*: An overall approach to work development and a tool for work analysis – The OWAS. In: The Journal of Methods – Time Measurement 6 (1978) 3, pp. 7-11.
[8]   *Huck, M.*: Produktorientierte Montageablauf- und Layoutplanung für die Robotermontage, VDI-Verlag, Düsseldorf, 1990.
[9]   *Lüth, T.*: Automatisierte Layoutplanung von Roboter-Fertigungszellen. Dissertation Universität Karlsruhe 1993.
[10]   *Roßgoderer, U.*: Optimisation of Production Systems Using Genetic Algorithms. In: Proceedings of the World Automation Congress (Volume 3). Montpellier: TSI-Press 1996, pp. 677-682.
[11]   *Roßgoderer, U.; Bauer, L.; Kugelmann, D.*: Aufbau und Programmierung von Roboterzellen mit USIS. In: Reinhart, G.; Milberg, J.: 3D-Simulation - Schneller, sicherer kostengünstiger zum Ziel. München: Herbert Utz Verlag 1995, pp. 51-66. iwb-Seminarberichte (1995) 10.
[12]   *Roßgoderer, U.; Woenckhaus, C.*: A Concept for Automatic Layout Generation. In: Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya (1995) Vol. 1 pp. 800-805.

Simulation and NC-Programming
for the Milling of Aircraft Integral Frames

Dr.-Ing. Xiaoyi Liu, Dipl.-Inform. Joachim Walter,
Daimler-Benz AG, Research Center Ulm,
Department of Manufacturing Engineering,
P.O. Box 2360, D-89013 Ulm,
Germany

Abstract: During the milling of aircraft integral frames consisting of aluminum malleable alloy, soft spots (material damage in the workpiece) will be formed by some undesirable cutting conditions, e.g. strong tool wear, small pocket thickness or slow path velocity of cutter. These soft spots are caused by overheating of the boundary layer in the workpiece. To avoide such soft spots and to save the time consumed and costly cutting tests at the production machine, an analytical simulation system is therefore developed, making it possible to determine the temperature distribution in the workpiece for the giving cutting conditions. This simulation system has been integrated in the NC verification system VERICUT via TCP/IP protocol. NC programmers and machine tool operators are not only able to verify the thermal loads in the workpiece by using the simulation system above, but also to modify/optimize the milling paths as well as cutting parameters, if necessary.

1. Introduction

During the milling process elastoplastic deformation of material in the cutting zone takes place. The deformed material forms chips that are removed from the rake face of the cutting tool. The majority of heat generated in the cutting zone is removed by the chip. But there is additional friction between the tool flank and the workpiece surface. One part of the heat caused by the friction is conducted into the workpiece. Depending on the thickness of the pocket bottom (and on the wear mark length of the cutter, the milling path and the cutting parameters), the temperature in the workpiece can reach the eutectic temperature of the aluminum malleable alloy (AlZnMgCu 1.5) and the boundary layer of the workpiece may be damaged (soft spot formation). The machining of aircraft integral frames consisting of many thin walled pockets is mentioned as an example.

The research objective is to promote process understanding so that softening can be avoided. The time consumed and costly cutting tests at the production machine can be reduced considerable.

In this paper an analytical thermal model of the milling process will be introduced. On basis of this model a simulation system SICUP Milling (Simulation of the Cutting Process - milling) has been developed and on real components approved, e.g. an aircraft integral frame SPANT. To verify this model, process temperature measurements have been carried out and compared with the simulation results. The details will be described in the following chapters.

2. Simulation of the Milling Process

2.1 Analytical thermal model to calculate the temperature distribution in the workpiece

Figure 1 shows a general configuration of the milling process. Because of the rotation symmetry a cylindrical coordinate system has been selected. The solution area is at the bottom of the workpiece under the cutter and shaded described in Figure 1. As heat sources the shear zone and the friction at the flank face of the cutter are taken into account.
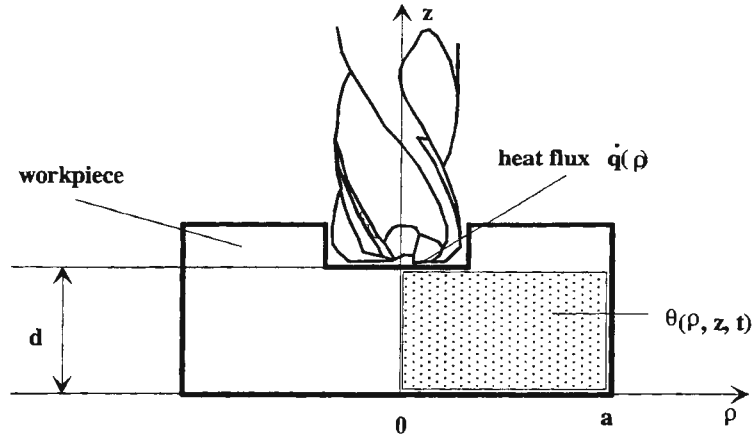
Figure 1: General configuration of the milling process

The resulted temperature field $\vartheta(\rho,z,t)$ in the workpiece can be described with the following partial differential equation:

$$\frac{\partial \vartheta}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial \vartheta}{\partial \rho} + \frac{\partial^2 \vartheta}{\partial z^2} = \frac{1}{\alpha} \frac{\partial \vartheta}{\partial t},$$ (1)

whereby $\alpha = \dfrac{\lambda}{\rho_w\, c_p}$,

λ heat conductivity of materials,

c_p specific heat capacity of materials,

ρ_w density of materials.
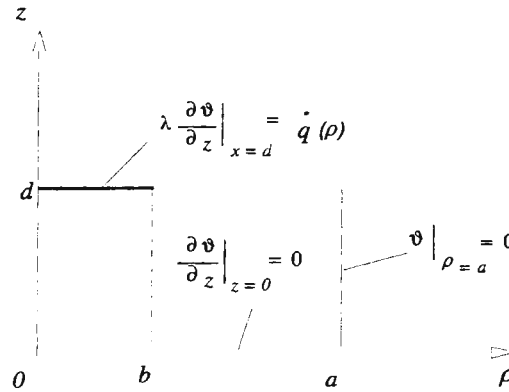
Figure 2: Boundary conditions for the analytical thermal model

For the analytical thermal model presented above following boundary conditions are considered (see Figure 2). At $\rho = a$ the temperature distribution $\vartheta(\rho,z,t)\,\big|_{\rho=a} = 0$ is assumed. The bottom area at the workpiece is thermal isolated, that means:

$$\frac{\partial \vartheta}{\partial z}\,\Big|_{z=0} = 0$$ (2)

There is heat source at $z = d$ (see Figure 3). This heat source at the cutter flank face will be modelled by boundary conditions of 2nd kind (Neumann´s boundary condition):

$$-\lambda \frac{\partial \vartheta}{\partial z}\,\Big|_{z=d} = \dot{q}(\rho)$$ (3)

507

$\dot{q}(\rho)$ is the heat flux and depends on the local line velocity of the cutter, the friction coefficient, the cutting force in the z-axis and the wear mark length:

$$\dot{q}(\rho) = \begin{cases} \dfrac{\mu \cdot F_z \cdot \dfrac{n}{60} \cdot 2\pi \cdot \rho}{n_s \cdot VB \cdot b}, & 0 \leq \rho \leq b \\ q_k, & b < \rho \end{cases} \tag{4}$$

with

n spindle rotation rate,

n_s number of cutters,

VB wear mark length,

b radius of the cutter,

μ friction coefficient between the tool flank face and the workpiece surface,

F_z cutting force in z-axis,

q_k cooling power of the cutting fluid.
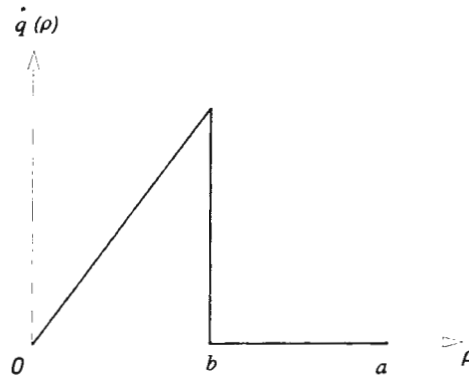
Figure 3: *Heat flux*

An analytical exact solution for the above described area with the corresponding boundary conditions can be obtained. By means of the orthogonal development these boundary conditions will be handled, which give the kernel of the analytical solution. The whole analytical solution results from the convolution integral of the kernel. Considering the zylindrical coordinate system, the analytical solution is written as follows:

$$\vartheta(\rho, z, t) = \sum_\beta C_\beta \cosh(\beta z) \cdot J_0(\beta \rho) \cdot \left[1 - e^{-2\alpha \beta^2 t} \right] \tag{5}$$

with

$$C_\beta = \frac{1}{\lambda} \frac{1}{a^2 J_1^2(x_r) \beta \sinh(\beta d)} \int_0^a \dot{q}(\rho) \cdot \rho \cdot J_0(\beta \rho) \, d\rho$$

$$\beta = \frac{x_r}{a},$$

and x_r are the solutions from $J_0(x) = 0$ (roots of Bessel function, first kind of order zero).

2.2   *Interactive simulation system SICUP Milling*

An interactiv simulation system SICUP Milling has been firstly implemented (see Figure 4). The temperature distribution in the workpiece can be calculated and rendered graphically on a workstation. The following parameters will be needed for the simulation computation:

- cutter geometry,
- wear mark length,
- spindle rotation rate,

- dry or wet cutting,
- milling path geometry and path velocity.
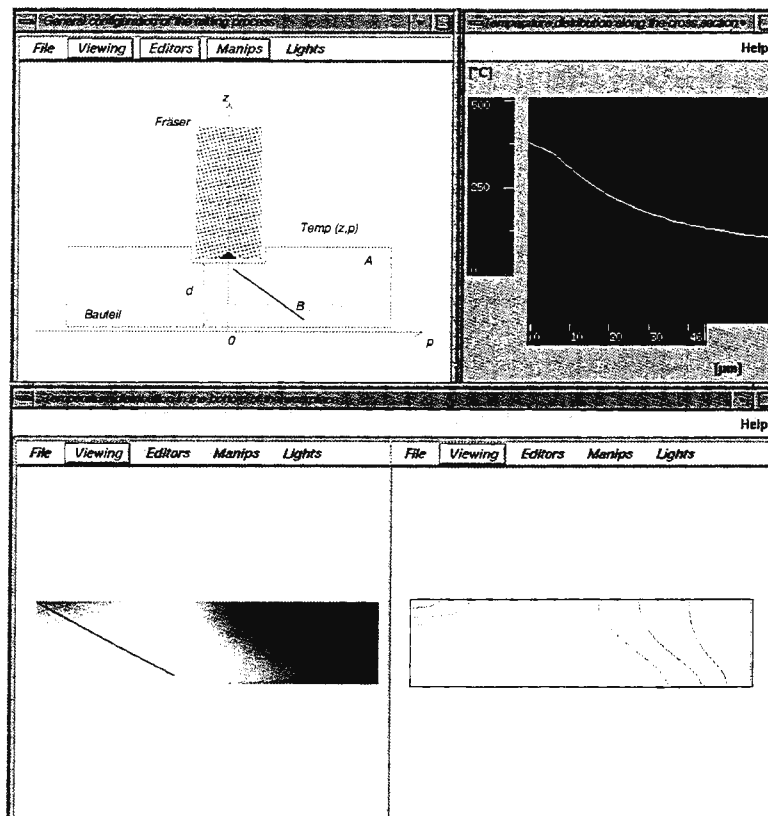
Figure 4: Interactive simulation system SICUP Milling

Figure 4 shows the draft of the cutter with the workpiece, solution area (A) and cross section (B). The temperature distribution in the bottom of the workpiece for given cutting parameters is displayed in false colors as well as in isothermal lines.

3. Integration of the analytical simulation system SICUP Milling into an NC verification system

To simulate a complete NC program and to detect critical positions on the workpiece, it is necessary to link the simulation system SICUP Milling with the existing NC verification system VERICUT (by CGTech, Irvine/CA, USA). VERICUT is one of the mostly used geometric verification tools for the manufacturing engineering in praxis. During the testing of possible NC program errors the geometric data of the workpiece, the cutter geometry and the cutting parameters are available for each time of machining.

The communication between SICUP Milling and VERICUT is based on the client-server network architecture. After the starting of VERICUT, SICUP Milling has full access to the geometry data of the workpiece and cutter as well as the data of cutting parameters, via TCP/IP protocol (see Figure 5). With the help of these parameters the local temperature distribution in the workpiece can be obtained immediately for each cutter position.

509

Dataexchange with VERICUT

- record number	- path velocity	- depth of cut
- xyz-coordinates	- spindle rotation rate	- width of cut
- cutter diameter	- bottom thickness under the cutter	- removal rate
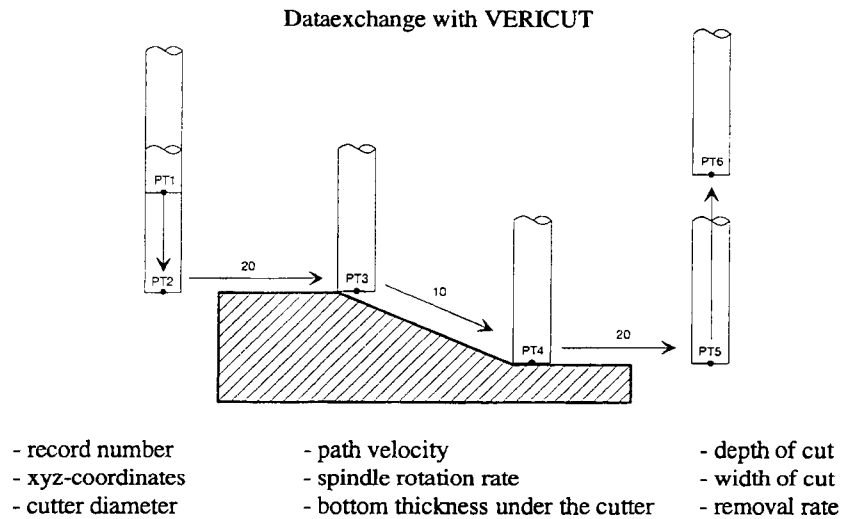
Figure 5: *Data exchange between SICUP Milling and VERICUT*

4. Simulation tests

In Figure 6 the simulation results at a test part will be presented. Considering groove and dip milling, the maximum temperatures on the workpiece surface for the tested NC program has been calculated. The window below in Figure 6 shows these temperatures depending on each cutter position or NC code. If the temperature exceeds the critical temperature, this cutter position will be described with hatched triangles in the upper window. The NC programmer is able to find out these cutter positions on the workpiece, so that the assigned NC codes can be modified, if necessary.
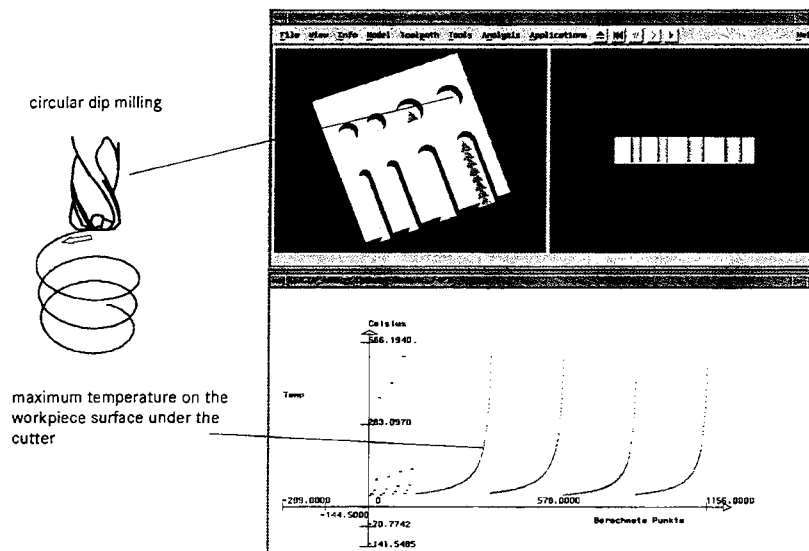
Figure 6: *Simulation on a test workpiece*
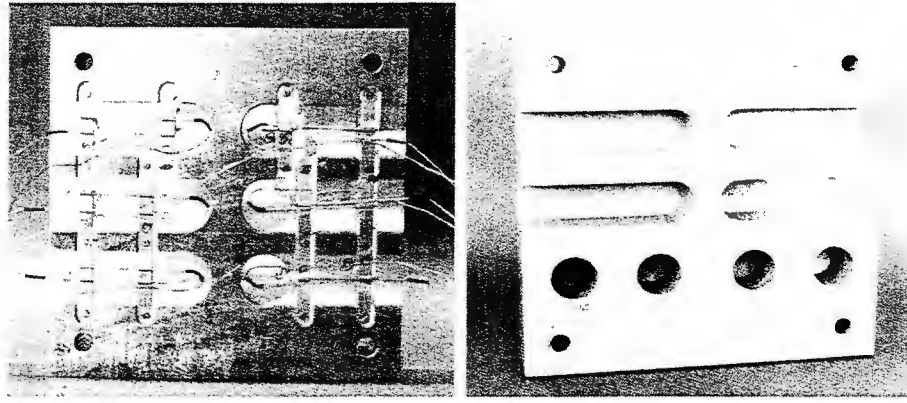
Figure 7:   a) with thermocouples provided test part (back side)
                  b) processed test part (front side)

5.   Comparison between the calculation and the measurement

The temperature on the workpiece surface after the zone of cut can be measured in-process with the help of thermocouples. A test part was prepared (see Figure 7), so that the temperatures directly below the cutter can be recorded. This part was milled with standard milling parameters, whereby originated temperatures in the floor range as well as the cutting forces in x-, y- and z-axis are measured. Housing loose NiCrNi thermocouples are used as temperature sensors. These thermocouples have features of beeing housing loose and of a small wire diameter of 0.1 mm by an extreme short triggering time (< 7 ms).
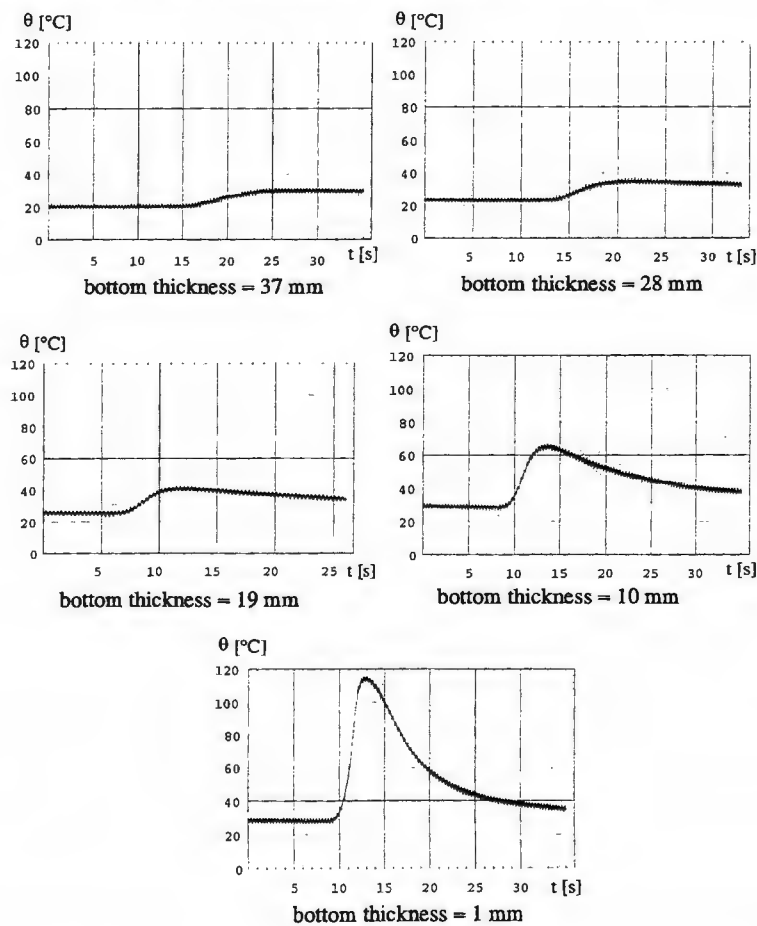
Figure 8: measured temporal temperature distribution by groove milling
for different bottom thickness (from 37 mm to 1 mm)
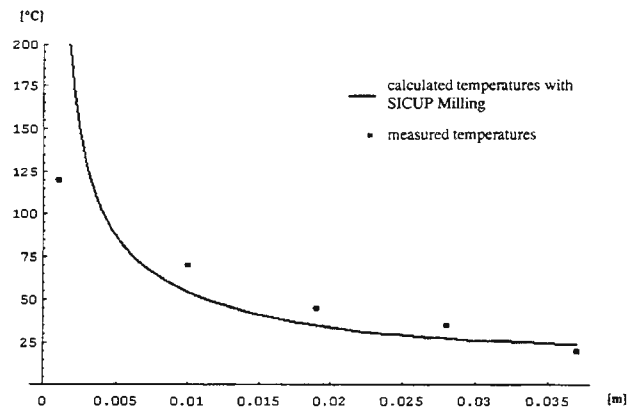
Figure 9: *Comparision between measurement and simulation*

Figure 8 shows the temporal temperature distribution by groove milling on the underside of the test part for different bottom thickness. The comparison between simulation and measurement is shown in Figure 9. The errors between the measured and calculated values are less than 8%.

6. Conclusion

As shown in this paper the simulation system of the milling process, based on the analytical thermal model above, can be used in the manufacturing engineering. SICUP Milling is considered as a simulation tool to analyse the temperature distribution in the workpiece during machining by given cutting conditions (cutting parameters, cutting tools and cutting fluids or dry cutting), so that softening of the boundary layer on the workpiece surface can be avoided and the cost of cutting tests as well as the development time for new parts will be reduced.

NC programmers and machine tool operators are not only able to verify the thermal loads in the workpiece by using the simulation system above, but also to modify/optimize the milling paths as well as cutting parameters, if necessary.

References

[1]   Liu, X.; Dörr, E. L.; Walter, J.; Daimler Benz AG, Research Center Ulm: Analytical Thermal Modelling and Simulation of the Cutting Process for hard Machinable Materials, Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, August 24-29, 1997, Berlin, Germany.

[2]   Dipl.-Ing. E. Berky; Dipl.-Ing. P. Müller-Hummel; Dr.-Ing. M. Lahres; Dr.-Ing. X. Liu; Dipl.-Inform. J. Walter; Daimler-Benz AG, Ulm: Fräsen von Flugzeugintrgralbauteilen aus Alu-Knetlegierungen mit Minimalmengen-schmierung im Serieneinsatz bei der DASA, VDI-Gesellschaft Produktionstechnik, Fachtagung „Umweltfreundliches Zerspanen", 23. und 24. Juni 1997, Aachen.

[3]   Lahres, M.; Linden, P.; Liu, X.; Müller-Hummel, P.; Walter, J.; Daimler-Benz AG, Ulm: Trockenbearbeitung von Aluminium-Knetlegierungen, 2. Karlsruher Kolloquium Fräsen, 6. März 1996, Karlsruhe.

Modelling a bellows with a beamlike component in ADAMS

Mikael Lind

Helsinki University of Technology, Control Engineering Laboratory, P.O.Box 3000, 02015 HUT, Mikael.Lind@hut.fi

Introduction

A micromanipulator consists of three hydraulic actuators. Each actuator consists of a piezoplate and a nickelbellows. The piezo actuator is placed into a tank filled with oil and the bellows is attached to the tank (figure 1). When voltage is applied to the piezo actuator, it deforms and the bellows elongates. The three bellows is attached to a upper plate forming a tripod-like structure (figure 2). By this we can control the movement of the mobile platform (upper plate). A more detailed description of the structure is given in [1],[2]. A bellows is a spring type of element, i.e. force required to deform the bellows is directly proportional to the displacement. Bellows have also bending characters when non-axial forces are applied to them. This bellows' bending might be a problem in the controlling of the manipulator's movement. Therefore a model that shows the bending and behaviour of the used bellows was needed. Furthermore the model could inform if the bending of the bellows could be used as joints instead of the balljoints that is usually used in structures like this. The model was created with ADAMS. Compared to the real manipulator the model is simplified to check only the tripod-like bellows structure (not the piezos), where the bellows one end are fixed.
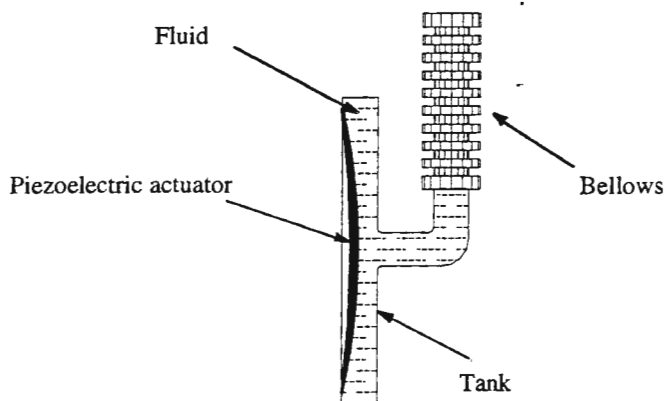
Figure 1. Overview of the actuation system.
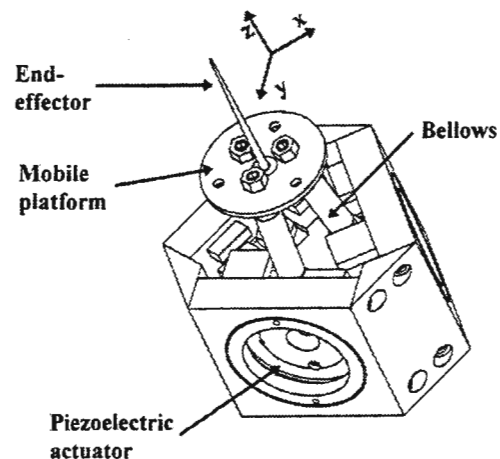
Figure 2. Overview of the manipulator.

ADAMS in general

ADAMS (Automatic Dynamic Analysis of Mechanical Systems) is a worldwide used simulation and modelling program especially for mechanical systems. It is possible to create models including masses mixed with mechanical and dynamical basecomponents. Basecomponents are for example joints between two parts, forceconnections, forced motions, springs and dampers. Markers are used in ADAMS to indicate location of parts or forces working point. The markers form own local coordinate system. The global or ground coordinate system is rigidly attached to the ground part. The movements of the system is usually realized by forcecomponents or with movable joints. Simple solid parts like boxes and cylinders are as standard components easy to utilize. In ADAMS its vice to simplify the models as much as possible. If standard features of the program limit the model creation it is always possible to add Fortran code. The analysis of the systems can give outputs like displacements, velocities, accelerations and forces.

Modelling the bellows

The springdamper function in ADAMS applies a translational spring and damper combination acting in parallel between two parts using linear stiffness and damping coefficients [3]. In figure 3, model 1, the springdamper function (SPDR) has been used. A cylindrical mass is attached on the top of the spring. The movement in the axial direction is done by a single component force (SFORCE). By the Accgrav statement the gravity is added. In model 1 the gravity influences axially along the beam. The springdamper statement could be used to model the bellows, but the problem is that the statement creates a spring, which has stiffness only in axial direction between two points.
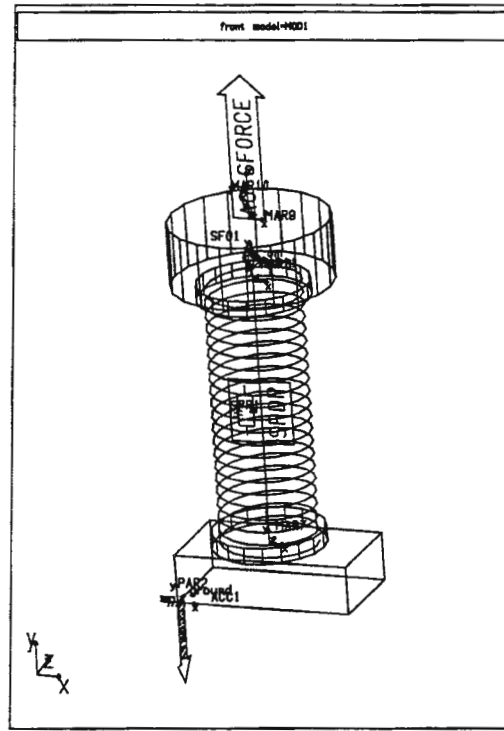
Figure 3. Model 1.

A better alternative is to use the beam element as a simplified model for the bellows. The beam statement defines a massless beam with uniform cross section. This element acts to transmit forces and torques in accordance with Timoshenko beam theory [3]. Thus the beam can be made flexible or rigid. Before modelling the simplified model into ADAMS we have to define the expressions and the values that gives the beam bending and spring characters that equals the bellows properties. The bellows are fixed in one end and therefore we study a beam that one end is fixed and one end is free, figure 4.
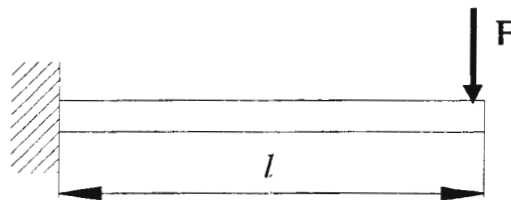
Figure 4. Beam.

If a force, F, acts in the free end of the beam, the maximum bending in the end is [4]

$$\max{}_v = v(l) = \frac{Fl^3}{3EI} \qquad (1)$$

where l is the length of the beam, E is modulus of elasticity and I is cross-sectional moment of inertia for the beam. To obtain the right bending character for the beam we use the manufacturer's aproximation of the modulus of elasticity and moment of inertia of the bellows

$$EI \approx 0.0635 \times O^2 n k_2 \tag{2}$$

O is bellows' outer dimension, n is the length of one convolution of the bellows and k_2 is the spring rate of one convolution. Model 2, figure 5, uses the beam statement.
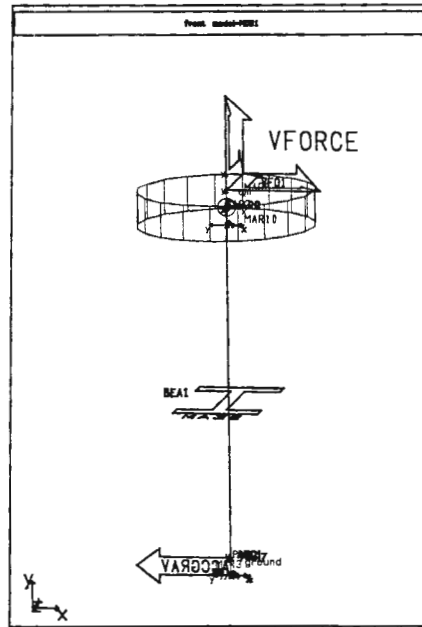
Figure 5. Model 2.

The beam's one end is fixed and in the other a cylindrical mass, m, is attached. If gravity acts in the direction of the negative x-axis, as in the figure, the bending in the end is

$$v = \frac{mgl^3}{3EI} \tag{3}$$
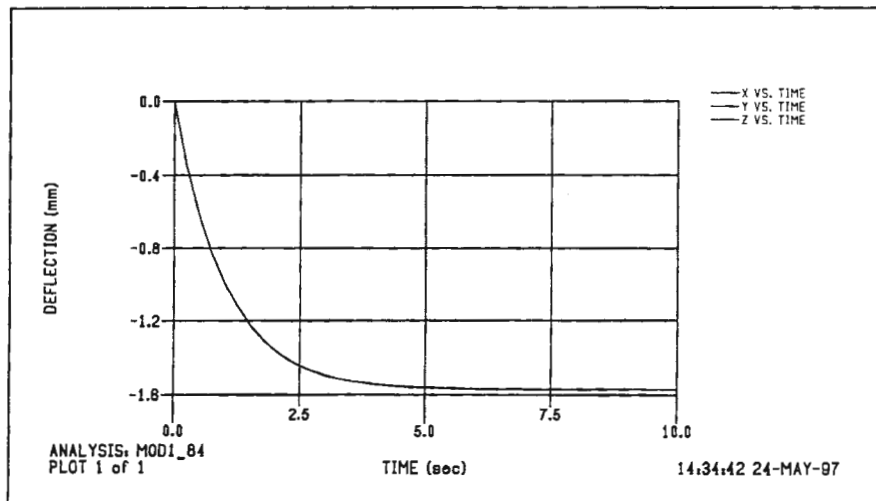
Figure 6. The bending of the beam.

In figure 6 the bending of the beam in model 2 is shown and we see that the maximum bending is about 1.55 mm. The mass m is 3 gram. At this stage the Vforce statement in the model is not used.

In the beam statement we can also define the axial elasticity of the beam. The beam's totally change in length is

$$\delta = \frac{F_{tot} l}{EA} \qquad (4)$$

where F_{tot} is the sum of axial forces, l is beam length, E is modulus of elasticity and A is beam's cross-sectional area. Correspondingly the elongation of the bellows is

$$x = \frac{F_{tot}}{k} \qquad (5)$$

where k is the bellows' spring coefficient. If we want the elongation of the beam to equal the elongation of the bellows ($\delta = x$), the condition

$$\frac{1}{k} = \frac{l}{EA} \qquad (6)$$

must be satisfied. This gives us the beam's cross-sectional area, A. The actual elongation of the beam is achieved by adding a vector force in the end of the beam. The Vforce statement is used to this. In Vforce statement it's possible to attach an additional floating marker to the part to be moved. The floating marker superimpose the vector force, in other words it keeps the vector force all the time in the same orientation as the marker during the motion. By attaching the floating marker in the free end of the beam the vector force acts along the beam even if it bends and this is actual how a bellows behave. A step force was added to model 2 in the vertical direction (axial along the beam) and the direction of gravity was changed to act in the negative y-direction. The used stepsize is 0.25 newtons (the number is corresponding to the maximum measured stroke of the actuators). The simulation result in figure 7 shows that the total displacement is 0.24 mm. The only active force during the first five seconds is gravity. Therefore the beam is first compressed and after that elongated.
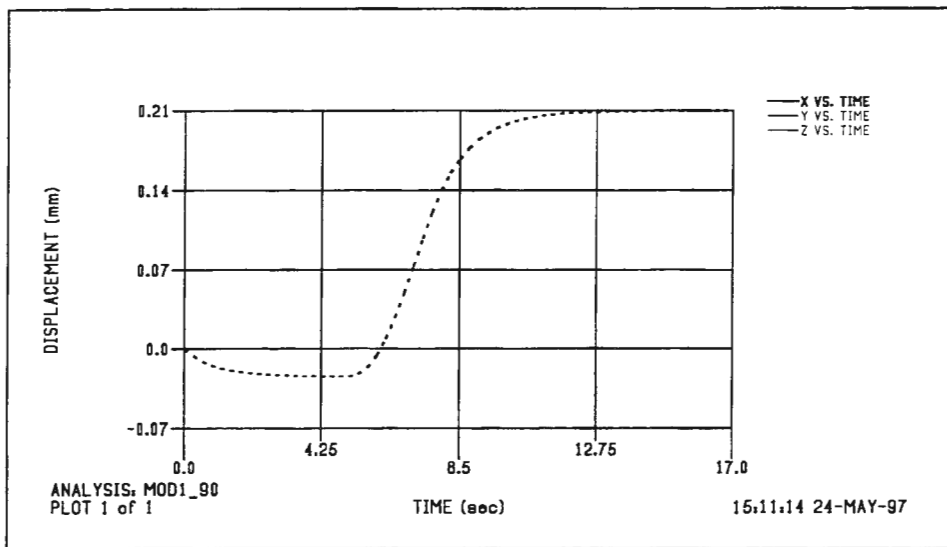
Figure 7. The elongation of the beam.

Modelling the manipulator

By combining three identical beams together by a cylindrical mass (the upper plate) to form the tripod-like structure, we can simulate the behaviour of the whole manipulator. The beams in figure 8, model 3, is inclined so that the angles between the fixed ground and the beams are 87 degrees. A similar vector force (Vforce) than in model 2 has been added to the end of every beam. The additional floating markers are now attached to the cylindrical upper plate together with the vector forces. The Vforce statement creates a translational vector force element that is specified using three orthogonal components. In these simulations we need only those vector components that are pointing in vertical directions along the beams. In the first simulation of this model we give at same time to all vertical vector components 0.25 newtons steps. In Adams you can with help of Output/Request statement place two markers for example into the same point. One is fixed to the ground part and the other is attached to the moving part. During the simulation the program calculates the distances between the markers and by this we get the displacement of the moved part, in this case the center of the plate [5].
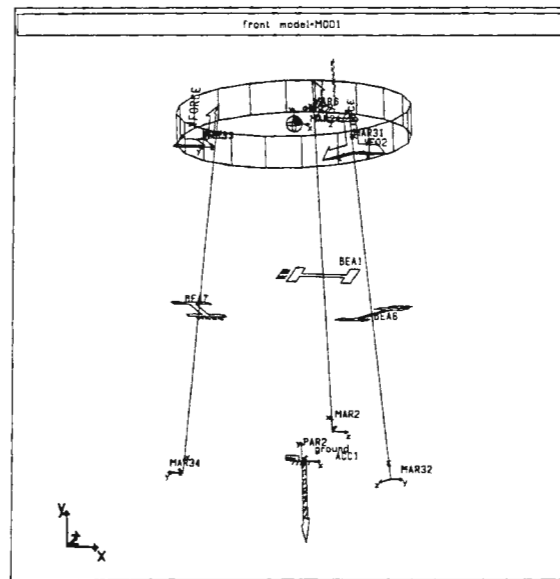
Figure 8. Model 3.
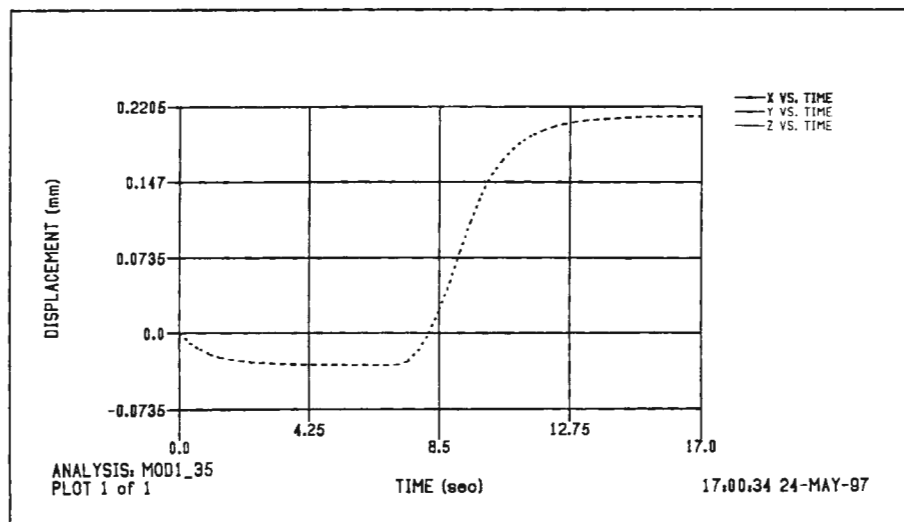
Figure 9. The displacement of the center of the plate.

517

The figure 9 shows the simulation result as the displacement of the center of the upper plate. No orientation of the plate should occur and at this point the model seems to work; we have displacement only in y-direction. In ADAMS it's easy to change the direction of gravity. This makes it possible to check how different orientations effect the movement of the manipulator.

In next simulation we change the direction of gravity to act in the negative direction of x-axis. By this the manipulator is actually positioned horizontally. We give to every vertical force component a step of 0.25 newton at the same time. In figure 10 we see that in the beginning when only gravity is influencing we have a total bending of 0.5 mm and the displacement of the plate in y-direction, when the vector forces are influencing, gives only an additional bending of about 30 micrometer. This shows that the bending in a tripod-like structure is reduced compared to a single beam. We can assume that the beams support each other.
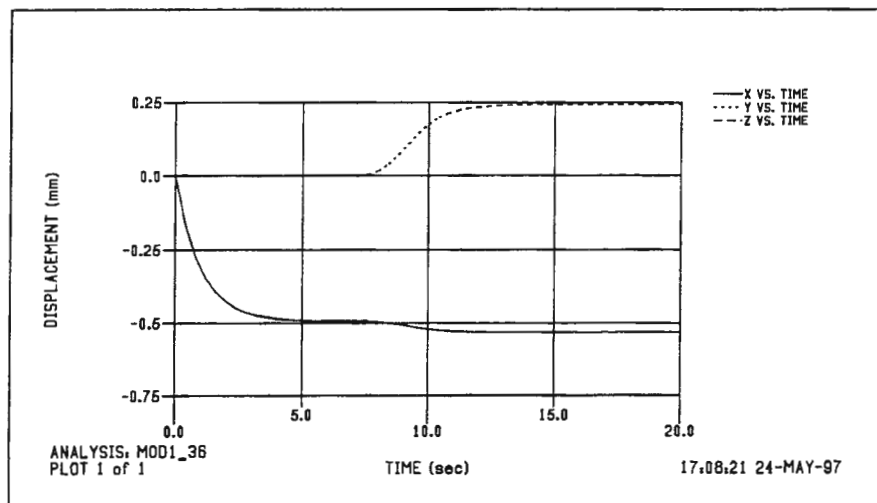
Figure 10. The displacement of the center of the plate.

In the last simulation we give first to one vector component a force step in vertical direction. After this similar steps are given to the other two vertical vector components. The gravity acts in the direction of negative y-axis.
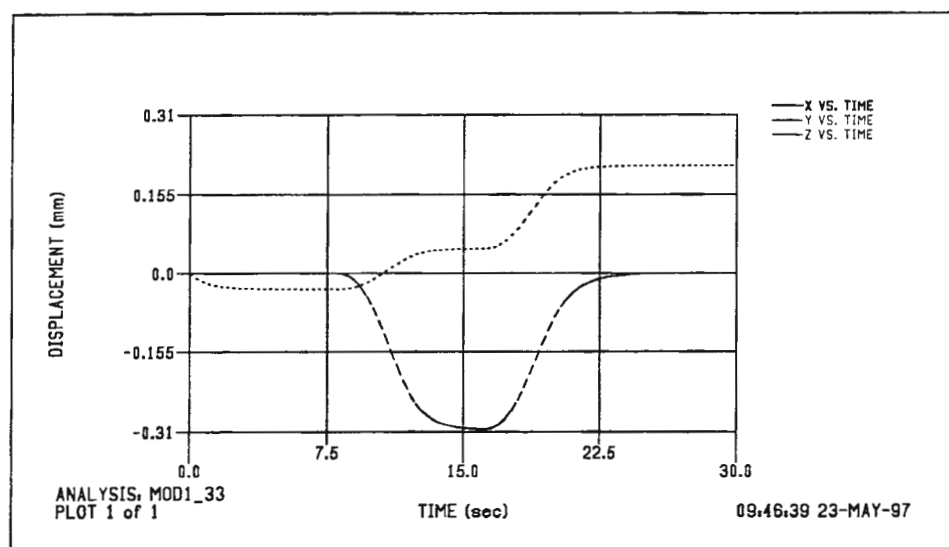
Figure 11. The displacement of the center of the plate.

518

From figure 11 is seen that the plate is first leaned about 30 micrometer in the direction of the negative z-axis and then the plate is orientated back to its horizontal position even if it has moved hundreds of micrometers in vertical direction.

Conclusions

The simulations show that the bending of the bellows should not cause any insuperable problems in controlling the movements of the manipulator. The created models of the manipulator can't be used as exact kinematic models because the used beam statement doesn't take into account that the diffraction line of the beam is actually curved. The errors could be corrected by using several short beams instead of one long beam. Furthermore we don't know exactly how good the approximation of the bellows as elastic beams is. In practice the non-linear pressure differences inside the bellows will cause some variations during the motion.

References

1   Kallio, P., Lind, M., Kojola, H., Zhou, Q., Koivo, H., An Actuation System for Parallel Link Micromanipulators, Proceedings of the Intelligent Robots and Systems, IROS'96, Osaka, Japan, November 1996, pp. 856 - 862.

2   Kallio, P., Lind, M., Zhou, Q. & Koivo, H. N. A parallel piezohydraulic micromanipulator - mechatronics aspects. CCAM - Control and Configuration Aspects of Mechatronics. Ilmenau, Germany 1997, Verlag ISLE, pp. 195-200.

3   Adams/View users reference manual, version 8.0. Mechanical Dynamics Inc., 1994.

4   Pennala, E. Lujuusopin perusteet (in Finnish), 7. p. Otatieto Oy, Helsinki 1995. 400 s.

5   Vesimäki, M., Saarinen, S.Adams Basic Training. Yliopistopaino, 1993.

MODELLING AND ANALYSIS OF LINEAR TIME-VARYING DIFFERENTIAL SYSTEMS BY LYAPUNOV-TRANSFORMATIONS

Kai Zenger , Jussi Orava

Helsinki University of Technology, Control Engineering Laboratory, P.O.Box 3000, FIN-02015 HUT, kai.zenger@hut.fi

Introduction

The model equations of linear time-varying systems cannot in general be solved analytically. Mathematically this means that the state transition matrix of a system of differential equations with time-varying coefficients can only in special cases be expressed by elementary functions. Because of that efforts have been made to study the properties of such systems without the need of actually solving the equations. From the point of view of system theory the main objective then is to study the structural properties of the system, i.e. stability, controllability, and observability. In the literature homogenous systems have been discussed extensively, but for the purposes of control engineering the theory for input-output systems and their representations would be needed.

The background of the work is related to the problem investigated by Lyapunov, in which the system matrix of an autonomous system was changed into a constant coefficient form by a suitable state transformation. If the transformation fulfils certain conditions, i.e. if it is a *Lyapunov-transformation*, the stability properties of the original and the target (transformed) system are the same. In this case the system is said to be *reducible*. Conditions for a system to be reducible can be found e.g. in [4] and [5]. In addition to reducibility the concept of *kinematically similar systems* has been presented in [2]. Two matrices are kinematically similar, if they are related by a time-varying state transformation, which prevails the stability properties. Additional results of the Lyapunov-transformations can be found e.g. in [7], [8], [9], and [10].

A drawback in the practical analysis of time-varying systems is that the transformation matrix depends on the state transition matrix of the homogenous original system, which means that it is not generally possible to solve for the transition matrix analytically nor to determine whether it is a Lyapunov-transformation or not. To study the theory further, one possibility is to carry out a simulation study to investigate the behaviour of the transformation. The original system representation is linear, which makes it possible to simulate it accurately enough by using a standard simulation package. It is then relatively straightforward to extend the simulation routine to the study of the transformation in the case of input-output systems. The final goal of the research is to develop tools to make it possible to construct control algorithms to control time-varying processes, which can with reasonable accuracy be described by linear time-varying system equations.

Transformations of the system equations

Consider the system representation

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0$$
$$y(t) = C(t)x(t) \tag{1}$$

in which $x(\cdot) \in \left(R^n\right)^R$, $u(\cdot) \in \left(R^m\right)^R$, $y(\cdot) \in \left(R^r\right)^R$ and $A(\cdot) \in \left(R^{n \times n}\right)^R$, $B(\cdot) \in \left(R^{n \times m}\right)^R$, $C(\cdot) \in \left(R^{r \times n}\right)^R$, respectively. Let us consider suitable transformations to change the system equations to the form

$$\dot{s}(t) = E(t)s(t) + F(t)v(t), \quad s(t_0) = s_0$$
$$z(t) = G(t)s(t) \tag{2}$$

in which the dimensions of the matrices E, F, and G are the same as the corresponding matrices in the original system.

The idea in the transformation is to find mappings to change the original system equations (1) into a form (2), in which the system matrices E, F, and G can be chosen arbitrarily. The purpose of doing this is to analyse the original system by means of the target system, which has been chosen to possess a simpler form; for example the coefficient matrices in (2) can be constant matrices.

Consider the linear but possibly time-varying transformations

$$x(t) = P(t)s(t), \quad u(t) = U(t)v(t), \quad y(t) = Z(t)z(t) \tag{3}$$

in which the continuously differentiable functions $P(\cdot) \in \left(R^{n \times n}\right)^R$, $U(\cdot) \in \left(R^{m \times m}\right)^R$, $Z(\cdot) \in \left(R^{r \times r}\right)^R$ are non-singular. The purpose of the transformations is to change the original system equations into a target system of the same dimension, in which the matrices can be chosen arbitrarily. Because the transformation matrices are invertible, they provide a one-to-one correspondence between the original and target systems.

It can be shown that in the case of a homogenous system a suitable transformation matrix P always exists, and that it preserves the stability properties if and only if it is a *Lyapunov-transformation*. No such results exist in the case of the input-output system, however, and the extension to such cases would provide an important contribution to the theory. It is relatively easy to show that an arbitrary system (2) can be obtained from (1) only by introducing suitable *augmentations* of the input and output variables. The analysis of the transformation then gets far more involved than in the case of the homogenous system alone. In what follows only the autonomous case is discussed in this paper.

Consider the special case $B(t)u(t) \equiv 0$ and $F(t)v(t) \equiv 0$. The aim is to find a state transformation, which forms a one-to-one correspondence between the solutions of the systems

$$\dot{x}(t) = A(t)x(t), \quad x(t_0) = x_0 \tag{4}$$

$$\dot{s}(t) = E(t)s(t), \quad s(t_0) = s_0 \tag{5}$$

It is easy to prove [8] that the transformation matrix always exists and it can be obtained from

$$P(t) = \Phi_A(t,t_0)P_0\Phi_E(t,t_0)^{-1} = \Phi_A(t,t_0)P_0\Phi_E(t_0,t) \tag{6}$$

in which $\Phi_A(\cdot,\cdot)$ and $\Phi_E(\cdot,\cdot)$ are the state transition matrices of the two systems, and P_0 is an arbitrary constant matrix of dimension $n \times n$, which fulfils the condition $x_0 = P_0s_0$.

The proof of the above result can be obtained by considering the solutions of (4) and (5), which are $x(t) = \Phi_A(t,t_0)x_0$ and $s(t) = \Phi_E(t,t_0)s_0$, $x_0 = P_0s_0$. It follows that $x(t) = \Phi_A(t,t_0)P_0s_0$ and $s_0 = \Phi_E(t,t_0)^{-1}s(t)$. By combining the two equations the result $x(t) = P(t)s(t)$, $P(t_0) = P_0$ follows, in which $P(t)$ is the transformation matrix (6). Since the state transition matrices are invertible and differentiable, the same holds for $P(\cdot)$ also.

The result proves the correspondence between the (solutions of) systems (4) and (5). Another form of the transformation can be obtained by differentiating (6), which leads to

$$\dot{P}(t) = \frac{\partial}{\partial t}(\Phi_A(t,t_0)) P_0 \Phi_E(t,t_0)^{-1} + \Phi_A(t,t_0) \frac{\partial}{\partial t}\left(P_0 \Phi_E(t,t_0)^{-1}\right)$$

$$= A(t)\Phi_A(t,t_0) P_0 \Phi_E(t,t_0)^{-1} - \Phi_A(t,t_0) P_0 \Phi_E(t,t_0)^{-1} E(t)\Phi_E(t,t_0)\Phi_E(t,t_0)^{-1} \quad (7)$$

$$= A(t)P(t) - P(t)E(t), \quad P(t_0) = P_0$$

The above system of differential equations is an initial value problem, which is in general as difficult to solve as the analytical expression for the state transition matrix $\Phi_A(t,t_0)$. The result is valuable in the sense that it proves the *existence* of a transformation matrix P, which relates the system matrices A and E.

Lyapunov-transformations

The transformation has practical value only if it preserves the stability properties of the original system and the target system. That leads to the concept of *Lyapunov-transformations* [3], which is defined as follows. Let the class $M_n(\cdot)$ be defined as $n \times n$-matrices $X(\cdot)$, which fulfil the conditions

-the elements of X are bounded and continuous functions,

-matrix X is differentialble and the elements of the derivative are bounded and continuous functions,

-there exists an $m > 0$ such that $\left|\det X(t)\right| > m$ for all time instants.

If the transformation (6) fulfils the above conditions i.e. for all t $P(t) \in M_n$, it is a Lyapunov-transformation, and in this case the stability properties of the two systems are the same [2]. In the literature alternative ways to study the stability properties of a system can be found, but the key issue is to determine, whether (6) is a Lyapunov-transformation or not. If an analytical solution for the state transition matrix cannot be found, there seems to be no general way to determine that, which emphasizes the complexity of time-varying systems when compared to time-invariant ones.

As an example, consider the autonomous system [6]

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 2-t^2 & 2t \end{bmatrix} x(t), \quad x(0) = x_0 \quad (8)$$

and study the possibility to find a transformation, which "changes" the system into the form

$$\dot{s}(t) = \begin{bmatrix} t & 1 \\ 1 & t \end{bmatrix} s(t), \quad s(0) = s_0 \quad (9)$$

The initial values of the states in both systems are chosen to be the same $x_0 = s_0 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$. A simulation program was constructed to simulate the systems and the transformation according to (6) and (7). The results which are presented in Fig. 1 suggest the analytical solution of the transformation to be

$$P(t) = \begin{bmatrix} 1 & 0 \\ t & 1 \end{bmatrix} \quad (10)$$

which is easily verified to be correct by using the equation (7). It should be pointed out that both the original and the target systems are unstable, and the transformation itself is not a Lyapunov-function (the first condition is violated, because all the elements of the matrix are not bounded).
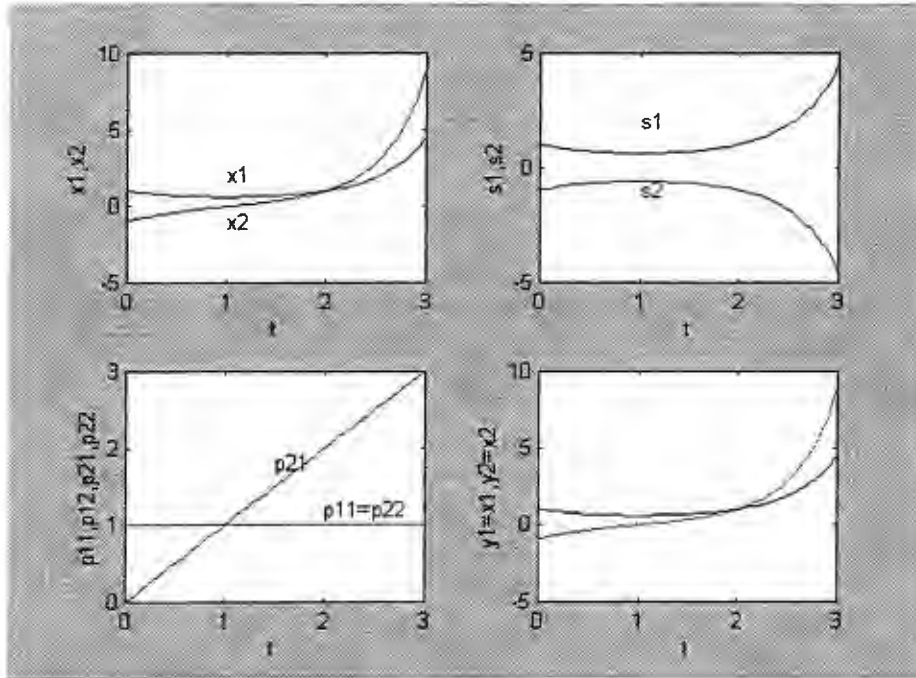
Fig. 1. States and their transformations (upper left: states of the original system; upper right: states of the target system; lower left: components of the transformation matrix (p12=0); lower right: states of the original system calculated based on the transformation).

Stabilization of an unstable system

Consider an unstable system (4) and introduce the problem of finding a feedback law $u(\cdot)$ such that the closed loop system

$$\dot{x}(t) = A(t)x(t) + u(t), \quad x(0) = x_0 \tag{11}$$

is stable. Assume further that the original system is reducible so that there exists a Lyapunov-transformation $P(\cdot)$, which changes the system into the form

$$\dot{s}(t) = Es(t) + P(t)^{-1}u(t), \quad s(0) = s_0 \tag{12}$$

where E is an unstable constant matrix. Applying a control law of the form $u(t) = -C(t)x(t)$ to (11) leads to

$$\dot{x}(t) = (A(t) - C(t))x(t), \quad x(0) = x_0 \tag{13}$$

By using the same Lyapunov-transformation $P(\cdot)$ as above to the system (13), the system matrix

$$F(t) = F = E - P(t)^{-1}C(t)P(t) \tag{14}$$

is obtained. (The matrix $C(t)$ is chosen in such a way the the matrix F is a stable constant matrix.) The state feedback matrix can then be determined from

$$C(t) = P(t)(E - F)P(t)^{-1} \tag{15}$$

The controller law is necessarily a stabilizing one, because F was stable and it was obtained by a Lyapunov-transformation from the matrix $A(t)$-$C(t)$.

As an example consider an example presented in [2], in which the autonomous system

$$\dot{x}(t) = \begin{bmatrix} -1 + \alpha \cos^2(t) & 1 - \alpha \sin(t)\cos(t) \\ -1 - \alpha \sin(t)\cos(t) & -1 + \alpha \sin^2(t) \end{bmatrix} x(t), \quad x(0) = x_0 \tag{16}$$

is considered. The eigenvalues of the system matrix are

$$\alpha_{1,2} = \frac{\alpha - 2 \pm \sqrt{\alpha^2 - 4}}{2} \tag{17}$$

which are independent of time and would indicate stability for $\alpha < 2$. However, it is well-known from linear system theory that the above way for determining the stability for time-varying systems does not generally hold. The situation can be analyzed as follows.

The system matrix (16) is periodic and hence reducible [1], [2]. In this case it is easy to check that a change of the state variable by the Lyapunov-transformation

$$P(t) = \begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix} \tag{18}$$

leads to the system with system matrix

$$E = \begin{bmatrix} \alpha - 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{19}$$

which is stable for $\alpha < 1$. Hence the system is unstable e.g. for $\alpha = 2$. By choosing the stable matrix

$$F = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \tag{20}$$

the stabilizing state feedback matrix in (13) becomes

$$C(t) = 3 \begin{bmatrix} \cos^2(t) & -\sin(t)\cos(t) \\ -\sin(t)\cos(t) & \sin^2(t) \end{bmatrix} \tag{21}$$

The responses of the autonomous and controlled systems have been presented in Fig. 2. The open-loop system gets unstable for $\alpha = 1$ as predicted; the time varying state feedback stabilizes the system.
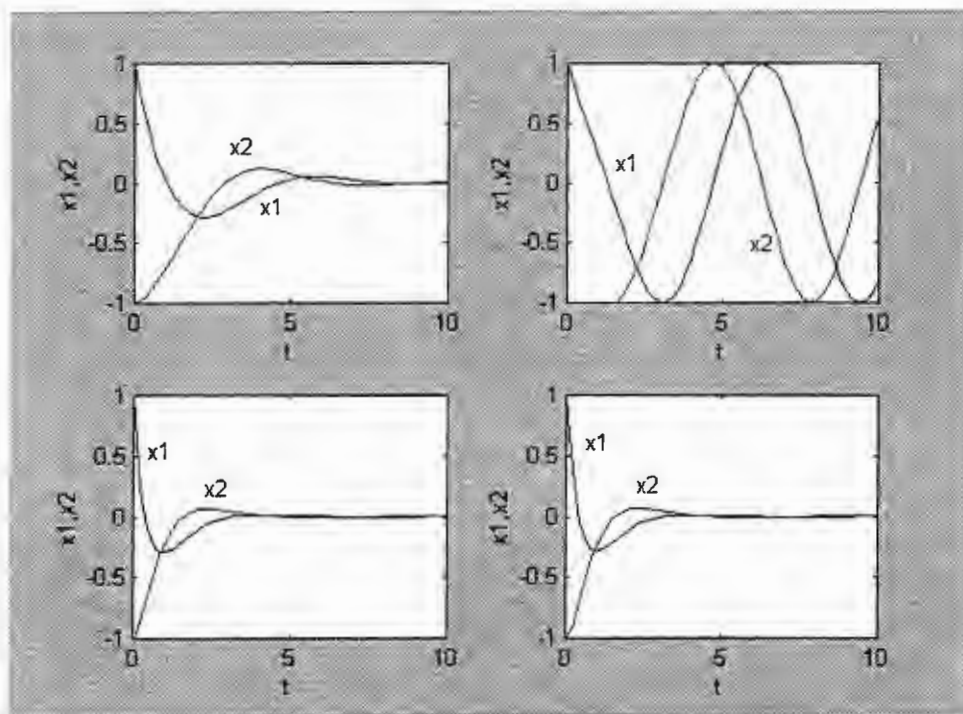
Fig. 2. States of the autonomous and controlled systems (upper pictures: autonomous system for parameter values $\alpha = 0.5$ and $\alpha = 1$; lower pictures: controlled system for the same parameter values).

References

1. Brockett, R. W. (1970). *Finite Dimensional Linear Systems.* John Wiley and Sons.

2. Harris, C. J. and J. F. Miles. (1980). *Stability of Linear Systems.* Academic Press Inc, London.

3. Lyapunov, A. M. (1966). *Stability of Motion.* Academic Press, New York (translated from the original text).

4. Markus, L. (1955). Continuous Matrices and the Stability of Differential Systems. *Math. Zeitschr,* **62**, pp. 310-319.

5. Nemytskii, V. V. and V. V. Stepanov. (1960). *Qualitative Theory of Differential Equations,* Princeton University Press, Princeton, NJ.

6. Rugh, W. J. (1993). *Linear System Theory,* Prentice-Hall.

7. Wu, M-Y. (1984). On Stability of Linear Time-Varying Systems. *Int. J. Systems Sci.,* **15**, No 2, pp. 137-150.

8. Zenger, K., and J. Orava. (1996). Time-Varying Transformations on Linear System Representations. *Proceedingss of the 13th IFAC World Congress,* San Francisco, USA.

9. Zhu, J. and C. D. Johnson. (1989a). New Results in the Reduction of Linear Time-Varying Dynamical Systems. *Siam J. Control and Optimization,* **27**, No 3, pp. 476-494.

10. Zhu, J. and C. D. Johnson. (1989b). *Invariants of Linear Time-Varying Dynamical Systems Under D-Similarity Transformation.* American Control Conference, **1**, pp. 827-833.

SIMULATION OF CONTROL SYSTEM FOR WATER TREATMENT

Yevgeny P. Zolotuchin, Victor V. Okol'nishnikov

Design Technological Institute of Digital Techniques of Siberian Branch of Russian Academy of Sciences

Institutskaja str. 6, 630090, Novosibirsk, Russia

zep@kti.nsc.ru

Abstract

Distributed process control systems for power engineering are developed at the Design Technological Institute of Digital Techniques. Models of control systems are designed in parallel with the development of actual control systems [1]. They are intended for implementing and testing automatic control programs and human operator interface. The models are constructed using Visual Interactive Manufacturing Simulation Environment CHIWIN [2].

In this paper the model of the developed control system for water treatment at the Surgut (West Siberian region) thermal power plant is described. The model is intended for testing automatic control programs for single-loop binary control and binary group control of filter systems. The model is integrated with the actual control system. It visualizes the simulation process by means of an actual mimic diagram and emulates an actual human operator interface.

CHIWIN automatically generates the model which provides means for control of process plant models individually (e.g., for one of a number of valves). In order that such a model provides means for the sequential control of the function group (e.g., regeneration cycle of filter system), it is additionally necessary to write a program in the simulation specification language. The CHIWIN input language is an extension of C++ language. The CHIWIN specification language is similar to the text form of Sequential Function Chart language.

Filter system description

The simplified mimic diagram of filter system is presented in Fig. 1. Such mimic diagram is observed by the human operator of control system. It consists of a filter and some remotely controlled valves V1 - V7. In operating mode of filter system the valves V1 and V3 are open, other ones are closed.

The process of cleaning of filter includes the following steps:

- *Water draining.* At this step the valves V2, V4, V5 are open, other ones are closed. The duration of the step is t1 minutes.

- *Air blowing.* High pressure air is added at the bottom of the filter during backwash. At this step the valves V2, V4, V6 are open, other ones are closed. The duration of the step is t2 minutes.

- *Air blowing and backwash.* Filter is cleaned by backwash process where water is reversed through the filter. At this step the valves V2, V4, V6, V7 are open, other ones are closed. The duration of the step is t3 minutes.

- *Backwash.* At this step the valves V4 and V7 are open, other ones are closed. The duration of the step is t4 minutes.

- *Reservation.* After this step all valves are closed. The filter system is ready for operation.
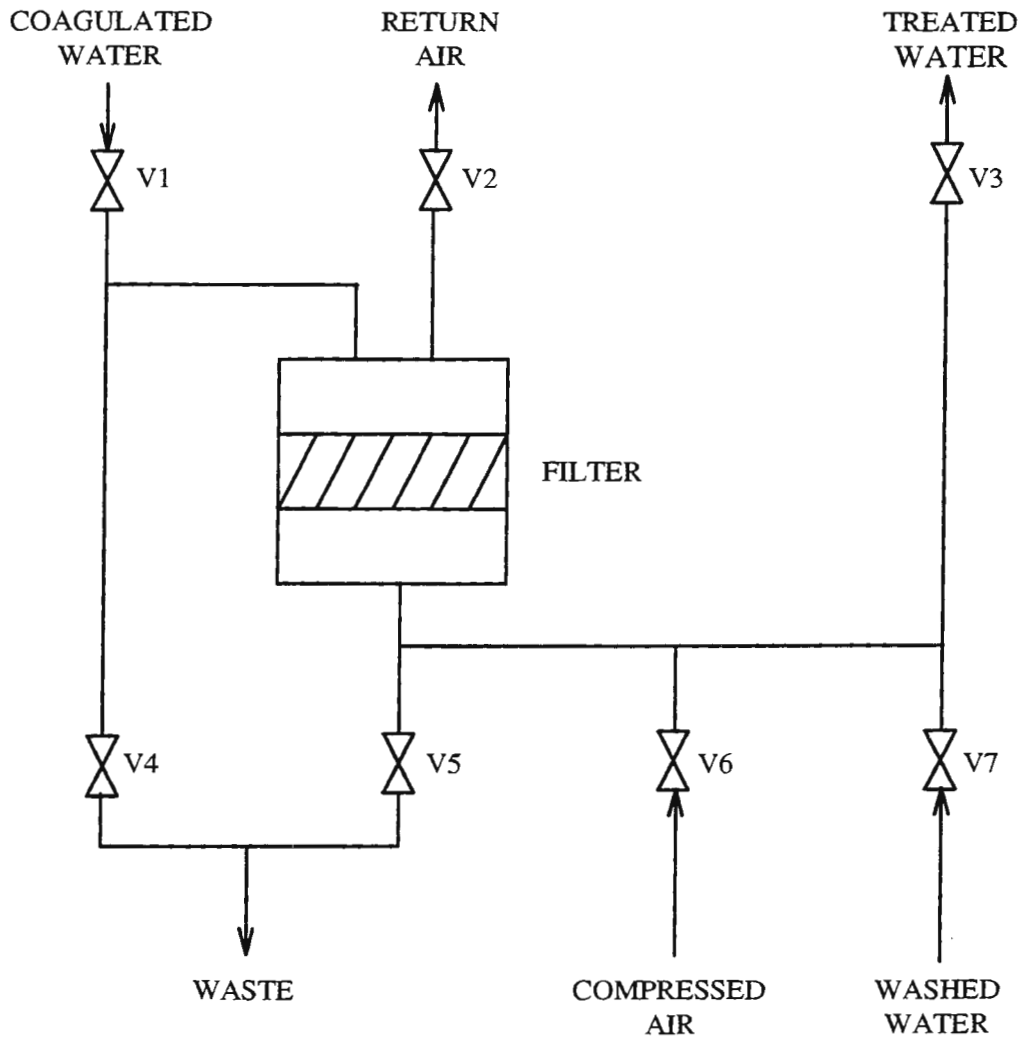
Fig. 1

Model description

The model can be programmed either in the input language (an extension of C ++ language) or in the specification language with simple syntax. CHIWIN has a specialized library for simulation of power plant distributed control system. This library contains classes corresponding to power plant equipment (e.g., a valve, a pump, and so on) and classes for typical automatic control programs (e.g., a function group, a step, and so on).

The model is created by the execution of the following program fragment in the specification language.

```
VALVE     V1
FG        EX
STEP      1
. . .
```

This model includes the instance V1 of the class VALVE and the instance EX (example) of the class FG (function group). A user of the model can exert control over the valve directly (OPEN command, CLOSE command) or indirectly (start-up instruction of function group). Function group can include several steps. In the simple case being considered the function group EX has only one step. The structure of the model is presented in Fig. 2.
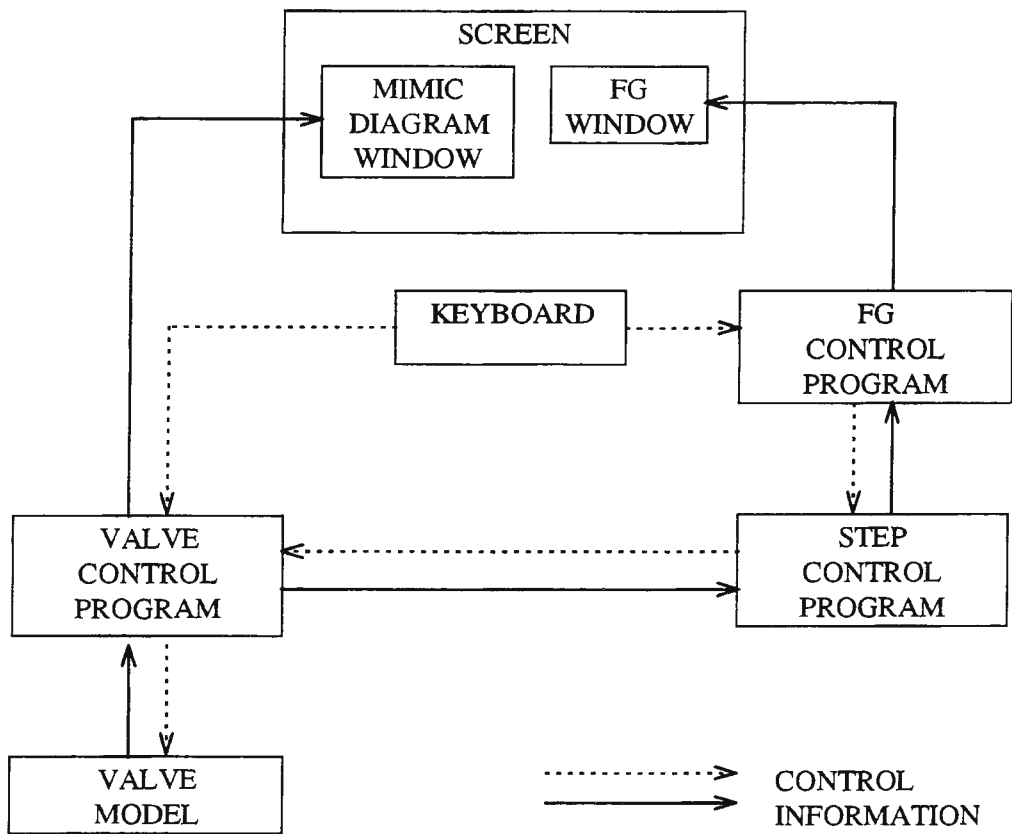
Fig. 2

The simplified model of the process of cleaning of the filter in the specification language is presented below. In an initial moment the filter is in operating mode (the valves V1 and V3 are open, other ones are closed).

VALVE	V1, V2, V3, V4, V5, V6, V7

FG	FILTER

STEP	1	// Water draining
CLOSE	V1	
CLOSE	V3	
OPEN	V2	
OPEN	V4	
OPEN	V5	

STEP	2
DELAY	t1

```
STEP        3                // Air blowing
CLOSE       V5
OPEN        V6

STEP        4
DELAY       t2

STEP        5                // Air blowing and backwash
OPEN        V7

STEP        6
DELAY       t3

STEP        7                // Backwash
CLOSE       V6
CLOSE       V2

STEP        8
DELAY       t4

STEP        9                // Reservation
CLOSE       V4
CLOSE       V7
```

After execution of the start-up instruction the function group is carried out sequentially step by step, since step 1. The step can consist of several operations, which are carried out in parallel. The end of the step occurs after all operations of the step are completed.

If the step was finished successfully, the following step (on number) is carried out. If the step was finished unsuccessfully (e.g., the critical time has elapsed), alternative step is carried out. There are no alternative steps in the program above.

During the execution of the model the valves state is visualized in the mimic diagram windows (see Fig. 2). There is the mimic diagram presented in Fig. 1 in the mimic diagram windows. The function group state (e.g., the number of step being executed and other information) is visualized in FG window.

The VALVE, FG, and STEP control programs are the actual programs, which work in the actual control system. The graphics images, the messages for the user are as in the actual system.

Conclusion

Control programs are debugged and tested on a model and then are transferred to the actual system. Thus, the CHIWIN simulation environment is also a prototyping environment. The model can be also used for operator training.

References

[1]   Nechepurenko M.I., Okol'nishnikov V.V., Iakimovitch D.A.: Control Systems Simulation for Power Engineering. Distributed Data Processing. Proc. of the Fifth international Workshop. Akademgorodok. Novosibirsk. Russia (1995) 221-226.
[2]   Okol'nishnikov V.V., Iakimovitch D.A.: Visual Interactive Industrial Simulation Environment. Proc. of 15th IMACS World Congress on Scientific Computation. Modelling and Applied Mathematics. Vol.5. Systems Engineering. Berlin (1997) 391-395.

Simulation of Two Adaptive Control Schemes of a Distributed Parameter Bioreactor

M.T. Nihtilä [†1], S. Julien [*], J.P. Babary [*], and J. Czeczot [‡]

† Department of Food Technology, University of Helsinki
P.O.Box 27, FIN-00014 University of Helsinki, Finland
* Laboratoire d'Analyse et d'Architecture des Systèmes
Centre National de la Recherche Scientifique (LAAS/CNRS)
7, avenue du Colonel Roche, F-31077 Toulouse Cedex 4, France
‡ Institute of Automatic Control, Technical University of Silesia
ul. Akademicka 16, PL-44100 Gliwice, Poland

e-mails: markku.nihtila@helsinki.fi, julien@laas.fr
babary@laas.fr, jczeczot@ia.polsl.gliwice.pl

1 INTRODUCTION

The distributed parameter system studied in this paper is described by a parabolic partial differential equation (PDE) system. The PDE system is nonlinear due to the nonlinear characteristics of the microbial growth, which is modelled by the Contois equation [2]. The objective is to design and study feedback control of this fixed-bed bioreactor aimed at cleaning waste water or improving the quality of drinking water. The reactor consists of a tube where microorganisms, which eat harmful nutrients, are on a fixed bed. The water to be treated flows through the reactor. The model includes dynamic equations for the biomass and for the substrate. Boundary conditions for the inlet and the outlet of the reactor tube complete the system model. The boundary conditions traditionally applied are so-called Danckwerts conditions [5]. These have been modified recently by Salmi & Romanainen [10] to obtain more realistic conditions, which are used here.

The basic technique to obtain a feasible simulation model for control design is the method of orthogonal collocation, see e.g. Villadsen & Michelsen [12] and Tali-Maamar et al. [11]. As a comparison Galerkin's finite element method was used in Nihtilä et al. [8] in the similar problem. Application of the orthogonal collocation method, which is a spatial discretization (semi-discretization) technique, results in a set of ordinary differential equations (ODE) which describe temporal behaviour of the biomass and substrate concentrations at certain discrete points (collocation points) along the reactor tube.

The boundary condition at the reactor outlet is a differential equation. In the controller design it is discretized in the time domain. Then a linear relationship is obtained between the unknown substrate consumption rate and the measured substrate concentration at the outlet. For the first adaptive controller this linear relationship serves as the measurement equation for estimating in real-time the unknown substrate consumption rate used in the control law. The least-squares technique with a forgetting factor is used in the substrate consumption rate estimation. This idea was originally developed and studied by Czeczot [3] & [4]. In the second adaptive controller the concentration and the specific growth rate of the biomass at the outlet are estimated by using the (real-time) observer technique of Bastin & Dochain [1] utilized also in Julien et al. [7]. The product of the biomass concentration, specific growth rate, a known semi-empirical function of Peclet number, the gradient, and the inverse of the yield coefficient gives the afore-mentioned substrate concentration rate, which is then used in the control law.

The controller design basically utilizes the idea of linearizing control of Isidori [6]. Then the desired closed-loop dynamics, i.e. the reference model, is the linear model

$$\frac{dS_{out}}{dt} = \lambda[S_{sp} - S_{out}(t)]; \quad \lambda > 0, \tag{1}$$

in which *sp* stands for the setpoint value and *out* for the outlet value of the substrate.

---

[1] Author to whom all the correspondence should be addressed

The basic equations of the bioreactor considered are given in Section 2. The two variants of the controller design are described next with some simulation results and with some concluding remarks.

2 SYSTEM MODELS

2.1 *Dynamics*

From mass balance considerations and under the assumption that the axial dispersion phenomenon is not negligible, we get the following dynamical model consisting in two partial differential equations, one for the biomass concentration $X[g/m^3]$ and the other for the substrate concentration $S[g/m^3]$:

$$\frac{\partial X}{\partial t}(z,t) = (\mu - k_d)X(z,t) \tag{2}$$

$$\frac{\partial S}{\partial t}(z,t) = \frac{D}{L^2}\frac{\partial^2 S}{\partial z^2}(z,t) - \frac{U}{L}\frac{\partial S}{\partial z}(z,t) - k_1\mu X(z,t) \tag{3}$$

with:

μ	:	specific growth rate $[h^{-1}]$	U	:	superficial velocity $[m/h]$ $(U = F/A)$
k_d	:	death rate coefficient $[h^{-1}]$	F	:	volumetric flow rate $[m^3/h]$
D	:	dispersion coefficient $[m^2/h]$	A	:	cross-section area of the reactor $[m^2]$
k_1	:	yield coefficient $[-]$	L	:	length of the reactor $[m]$

The state variables S and X depend on the time $t[h]$ and on the dimensionless space variable $0 \leq z \leq 1$. The value of the specific growth rate μ is described by the Contois model [2]:

$$\mu = \mu_{max}\frac{S(z,t)}{K_C X(z,t) + S(z,t)} \tag{4}$$

where $\mu_{max}[h^{-1}]$ is the maximum specific growth rate and $K_C[-]$ the saturation coefficient.

2.2 *Boundary conditions*

This mathematical model is of parabolic type and nonlinear. Boundary conditions of the space domain $[0,1]$ are necessary to get its solution. The classical Danckwerts boundary conditions [5] are defined by the space gradients of the substrate concentration at the inlet and the outlet of the reactor:

$$\frac{\partial S}{\partial z}(0,t) = Pe\left[S(0,t) - S_{in}(t)\right] \quad ; \quad \frac{\partial S}{\partial z}(1,t) = 0 \tag{5}$$

where $Pe = \frac{UL}{D}$ is the Peclet number and $S_{in}(t)$ the substrate concentration of the input feed.

This formulation means that the concentration gradient vanishes at the reactor output. Such an assumption is not always realistic for reactors operating close to plug flow conditions or intermediate conversions. In 1994, Salmi & Romanainen [9] [10] proposed an extension of the so-called Danckwerts boundary conditions implying that the output concentration gradient becomes zero in the case of complete backmixing and that it coincides with the gradient predicted by the plug flow model for low values of the dispersion coefficient D.

The Salmi & Romanainen boundary conditions are expressed as follows:

$$\frac{\partial S}{\partial z}(0,t) = Pe\left[S(0,t) - S_{in}(t)\right] \tag{6}$$

$$\frac{\partial S}{\partial t}(1,t) = -\frac{U}{L}\frac{\partial S}{\partial z}(1,t) - f(Pe)k_1\mu X(1,t) \tag{7}$$

$f(Pe)$ is a semi-empirical function given by $f(Pe) = (1 + 1/Pe)^{-r}$. The specific growth rate value is evaluated at the outlet, i.e. $\mu = \mu(X(1,t), S(1,t))$

Note that the boundary condition at the reactor outlet is a dynamical partial differential equation with respect to independent variables t and z. This is not the case for the Danckwerts boundary condition (5) at the outlet.

2.3 *Approximation of the PDE model*

The original PDE model (2)-(4) with the boundary conditions (6)-(7) is then approximated by a set of ODEs which is obtained via the approximation of the solutions by the finite series (c.f. [12] & [11]):

$$X(z,t) \approx \sum_{i=0}^{N+1} L_i(z)x_i(t) \quad ; \qquad S(z,t) \approx \sum_{i=0}^{N+1} L_i(z)s_i(t) \tag{8}$$

where $x_i(t)$ and $s_i(t)$ are the values of X and S at certain discrete spatial positions *(collocation points)* along the bioreactor. The basis functions $L_i(z)$ are chosen as orthogonal functions *(e.g. Lagrange polynomials)*. N is the number of internal collocation points. Specifically, the points z_0 and z_{N+1} correspond to the inlet ($z = 0$) and the outlet ($z = 1$) of the reactor.

By applying the collocation method together with the notations

$$l_{ij} = (dL_i(z)/dz)_{z=z_j} \quad ; \quad m_{ij} = \left(d^2 L_i(z)/dz^2\right)_{z=z_j}$$

we get the following ordinary differential system for $i = 1, 2, ..., N$

$$
\begin{aligned}
\dot{x}_i(t) &= (\mu_i - k_d)x_i(t) \\
\dot{s}_i(t) &= \tfrac{D}{L^2} \sum_{j=0}^{N+1} m_{ij}s_j(t) - \tfrac{U}{L} \sum_{j=0}^{N+1} l_{ij}s_j(t) - k_1 \mu_i x_i(t) ,
\end{aligned}
\tag{9}
$$

where the values of the specific growth rate are $\mu_i = \mu(x_i(t), s_i(t))$. The boundary conditions of Salmi & Romanainen are approximated correspondingly

$$
\begin{aligned}
\sum_{j=0}^{N+1} l_{0j}s_j(t) &= Pe\,[s_0(t) - S_{in}(t)] \\
\dot{s}_{N+1}(t) &= -\tfrac{U}{L} \sum_{j=0}^{N+1} l_{N+1,j}s_j - f(Pe)k_1\mu_{N+1}x_{N+1} .
\end{aligned}
\tag{10}
$$

3 CONTROLLER DESIGN AND SIMULATION

3.1 *Adaptive control law*

Controller design is based on the use of the boundary condition (7) of the substrate at the outlet. The substrate consumption rate at the outlet is given by

$$R_{out}(t) = f(Pe)k_1\mu_{out}X_{out}(t) \tag{11}$$

(Note: the subindex "out" is equivalent to the subindex "N+1".) Then the design equations are for our case of $L = 1$

$$\frac{dS_{out}}{dt}(t) = -U(t)\frac{\partial S}{\partial z}(1,t) - R_{out}(t) \tag{12}$$

$$\frac{dS_{out}}{dt}(t) = \lambda[S_{sp} - S_{out}(t)] \tag{13}$$

from which we obtain the theoretical control law

$$U(t) = -\frac{\lambda[S_{sp} - S_{out}(t)] + R_{out}(t)}{\frac{\partial S}{\partial z}(1,t)} \tag{14}$$

Due to the fact that the substrate consumption rate $R_{out}(t)$ depends on the Peclet number Pe, which in turn depends on the control $U(t)$, the control law (14) cannot be used in continuous-time formalism without iterations. In discrete-time framework the control law can be used as a discrete recursion for the subsequent values of the control signal. Consequently, in addition to spatial discretisation based on the orthogonal collocation, temporal discretisation and estimation-based calculations for obtaining the values of the substrate consumption rate are applied. Then the final control law is for the both adaptive methods to be described the following

$$U^{k+1} = -\frac{\lambda[S_{sp} - S_{out}^k] + \hat{R}_{out}^k}{\sum_{j=0}^{N+1} l_{N+1,j}s_j^k} \tag{15}$$

In applying the control law it is assumed that the substrate concentrations at the collocation points s_j^k are measured. Two different methods for obtaining estimates of the substrate consumption rate at the outlet are used.

The first method uses a direct procedure to obtain recursively the substrate consumption rate $\hat{R}_{out}^k$. It is based on the least-squares estimation with a forgetting factor (c.f. Czeczot [3] & [4]). The other method uses the asymptotic observer theory of Bastin & Dochain [1] for estimating values of the specific growth rate $\hat{\mu}_{N+1}^k$ and the biomass $\hat{X}_{N+1}^k$. Detailed estimator equations and the background were presented in Julien et al. [7] and are not repeated here. These estimated values are then used to calculate the estimate $\hat{R}_{out}^k$ by Eq. (11).

3.2 Simulation results

Results of simulation runs of the both adaptive controllers are depicted in Figs 1 and 2. The following parameter values were used (T is the discretisation time interval):

$$
\begin{array}{llllllll}
\mu_{max} & = & 0.35\ h^{-1}\ , & K_C & = & 0.4\ , & k_1 & = & 0.4\ , & k_d & = & 0.05\ h^{-1}\ . \\
L & = & 1\ m\ , & A & = & 0.02\ m^2\ . & D & = & 0.1\ m^2/h\ , \\
N & = & 5\ , & T & = & 0.25\ h & \lambda & = & 0.25\ h^{-1}
\end{array}
$$

The adaptive controllers were able to keep the process stable in the presence of disturbances (the inlet substrate concentration S_{in} increases from 5 g/l to 6 g/l at time $t = 10\ h$ and the maximum growth rate μ_{max} increases from 0.35 h^{-1} to 0.40 h^{-1} at time $t = 40\ h$). Moreover, the controller successfully drives the system to a new steady state after the setpoint change S_{sp} from 1.17 g/m^3 to 1.40 g/m^3 at time $t = 70\ h$.

The both adaptive controllers gave smooth output transients from one steady state to another. On the other hand changes in the input concentration S_{in} caused larger disturbances of the control and of the output concentration when the asymptotic observer technique was applied as it is seen in Fig. 1. Furthermore, the computational burden was smaller when the substrate consumption rate R_{out} was directly estimated. The traditional PI-controller could not well compensate transient disturbances caused by S_{in}.

REFERENCES

[1] G. Bastin, and D. Dochain, *On-Line Estimation and Adaptive Control of Bioreactors*. Amsterdam: Elsevier, 1990.

[2] D. Contois, "Kinetics of bacterial growth relationship between population density and specific growth rate of continuous cultures," *Journal of General Microbiology* **21** (1959), p.40.

[3] J. Czeczot, "Substrate consumption rate - New concept of measuring and monitoring in the activated sludge process," *Proc. 6th IFAC Int. Conference on Computer Applications in Biotechnology*. Garmisch-Partenkirchen, Germany, 1995.

[4] J. Czeczot, "Substrate consumption rate application to the minimal-cost model based adaptive control of the activated sludge process," *Proc. 7th IAWQ Workshop on Instrumentation, Control & Automation of Water & Wastewater Treatment & Transport Systems*, Brighton, U.K., 1997.

[5] P.V. Danckwerts, "Continuous flow systems," *Chem. Engng Sci.* **2** (1953), pp. 1-13.

[6] A. Isidori, *Nonlinear Control Systems*, 2nd Ed. Berlin: Springer, 1989.

[7] S. Julien, J.P. Babary, and M.T. Nihtilä, "On modelling of boundary conditions for fixed-bed reactors," *Proc. 1995 EUROSIM Conference*, Vienna, Austria, 1995, pp. 451-456. 231-237.

[8] M.T. Nihtilä, J. Tervo, and J.P. Kaipio, "Simulation of a distributed parameter bioreactor by FEM approach," *Simulation Practice and Theory* **5** (1997), pp. 199-216.

[9] Romanainen, J., "Numerical strategies in solving gas-liquid reactor models," Academic Dissertation, Abo Academy, Finland, 1994.

[10] T. Salmi and J. Romanainen, "A novel exit boundary condition for the axial dispersion model," *Chem. Eng. & Processing*, **34** (1995), pp. 359-366.

[11] Tali-Maamar, N., Damak, T.,Babary, J.P., and Nihtilä, M.T., "Application of a collocation method for simulation of distributed parameter bioreactors," *Math. Comput. Simul.* **35** (1993), pp 303-319.

[12] J.V. Villadsen and M.L. Michelsen, *Solution of Differential Equation Models by Polynomial Approximation*. Prentice-Hall, 1978.
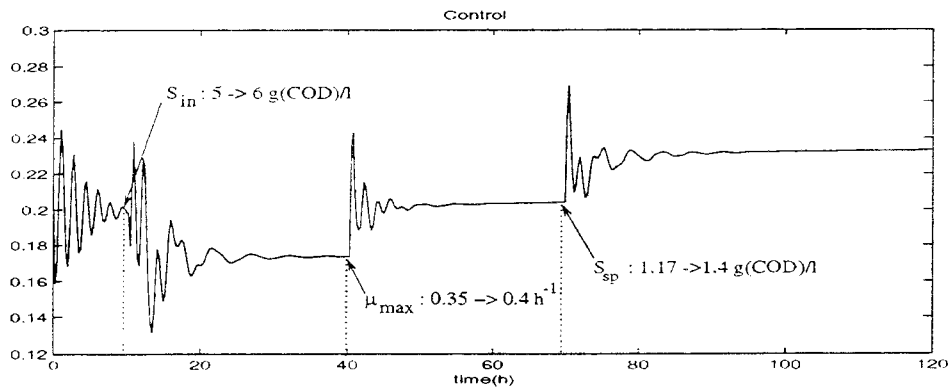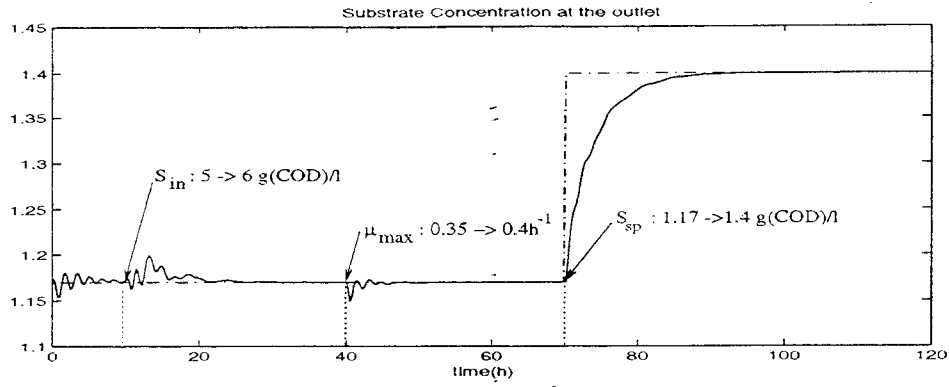
Fig.1. Simulation of the adaptive controller based on the estimation of the specific growth rate μ_{N+1} and biomass X_{N+1}
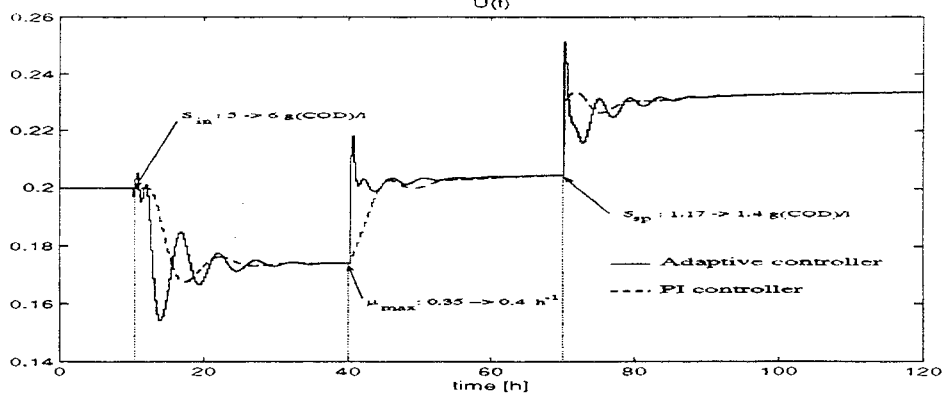

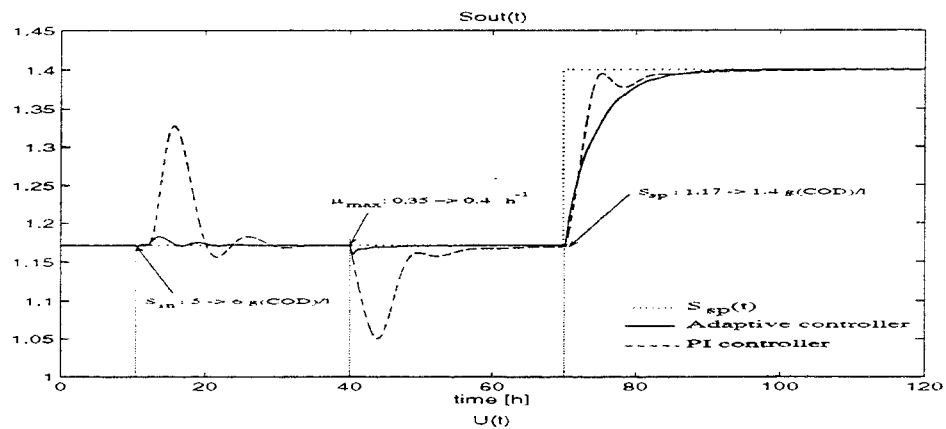Fig.2. Simulation of the adaptive controller based on direct estimation of the substrate consumtion rate R_{out} and of the traditional PI-controller.

Steam Temperature Control in Thermal Power Plant by Means of Fuzzy Logic - A Simulation Study

Dražen Lončar, Nikola Šerman

University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Luciceva 5, HR-10000 CROATIA
dloncar@fsb.hr, nserman@fsb.hr

Abstract
Performance of the fuzzy logic controller has been investigated in the control of the steam temperature in the utility thermal power plant and compared with the performance of the state controller. Various operating conditions have been simulated using detailed nonlinear model of the entire line of the steam superheaters.

INTRODUCTION

There are many reports on application of the fuzzy logic control technology in different industrial areas. Described examples are ranging from low-level to supervisory control. Fuzzy logic controllers (FLC's) are usually derived from a knowledge acquisition process or are automatically synthesized from self organizing control architectures defining a nonlinear mapping from the system's state space to the control space. The key advantages of FLC's are their effectiveness in synthesizing nonlinear controllers for dynamic systems and robustness with respect to internal and external disturbances [1].

Considering low level control, advantages and excellent performance of the FLC are often limited on the low order processes. To deal with higher order system dynamics more complex internal structures are usually required, but this fact reduces attractiveness of the FLC [3, 9]. Some authors suggest compromised solutions that conserve simplicity of the controller and are still capable to perform satisfactory control of higher order system. One of these compromised solutions, the fuzzy $PID^{(n)}$ structure proposed in [2], has been investigated in this work in the control of the steam temperature in the utility thermal power plant.

The investigation has been performed by simulation using nonlinear control-volume based model of the thermohydraulic processes in the entire line of the steam superheaters. The performance of the FLC in the control of the live steam temperature has been compared with the performance of the chosen advanced control concept.

CONTROL LOOP DESCRIPTION

The Superheater
The superheater under consideration is positioned in the flue gas duct of the 320 MW thermal power plant and comprises 588 tubes connected in parallel between unheated manifolds. Geometry of the representative tube is as follows: overall tube length 22,6 m, outer tube diameter 0,0318 m, wall thickness 0,0063 m. Heat transfer from flue gas to the superheater tube wall is dominantly convective. The temperature dynamics is dominated by a non-trivial combination of transport phenomena and energy storage phenomena depending on the thermohydraulic conditions of the steam flow. In particular, the interaction between steam and metal walls causes the occurrence of a time delay larger than pure transport delay which basically depends on the ratio between global heat transfer coefficient and the flow heat transfer capacity.

Control system scheme
The steam temperature control is one of the most difficult problems that a control engineer can meet working on a thermal power plant because of the several reasons [7]. There are significant interactions between superheated and reheated steam temperature control, and possibly with steam generation when recirculation gas flow or burner tilt are used as control variables. The heat exchange process is highly nonlinear and the involved process transfer functions are intrinsically transcendental. These problems are usually tackled by combination of feedforward and feedback actions. Control parameter adaptation is often introduced letting parameters be varied in dependence on the main steam flow. Due to limited performance of traditional concepts, which cannot fulfill the recently increasing requirements caused by the use of high steam parameters, advanced control concepts are often used.

The steam temperature control scheme is usually divided into main and subordinated loop. This approach is known as cascaded control. Various concepts have been implemented in the main loop while continuous or three-point PI controller have been used for driving actuator in the subordinated loop [5].

The state controller with observer has been chosen for comparison with the FLC in this paper. It is extensively described in [6] and schematically shown in Fig. 1. The controller is commercially available as a standard function module within the distributed process control system together with the tables for the Instead of setting some 2*N feedback gains the number of tuning factors can be reduced to two: time constant of observer model and inverse loop gain. Both factors are load - dependent.

Fig.1 State controller with observer
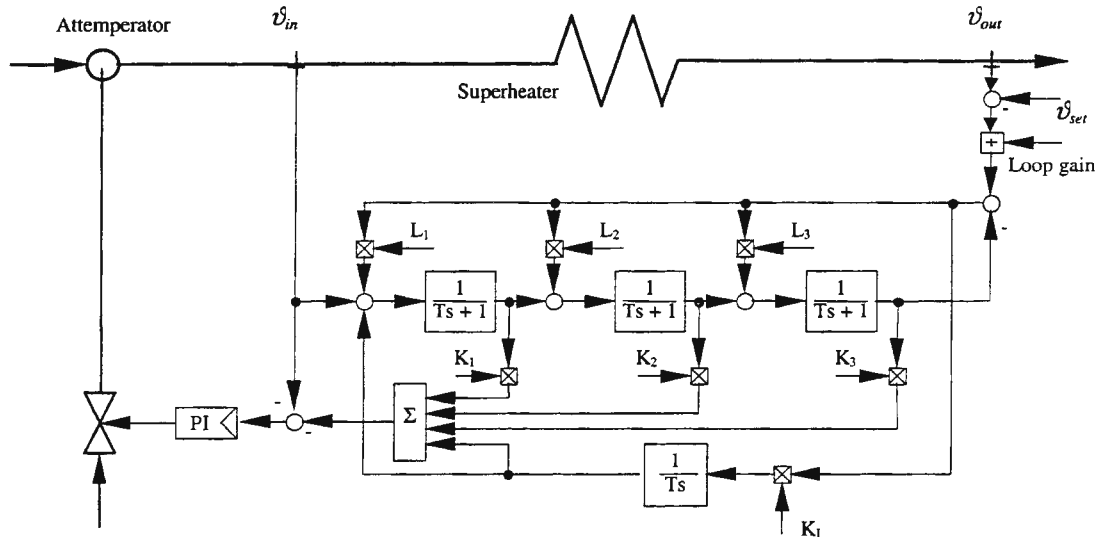
The scheme of the FLC structure is shown Fig 2. In this work parameter tuning of the FLC has been performed according to following recommendations [4]:

- The output scaling factors KP and KI have a direct impact on the overall gain, and consequently on the stability. This scaling should play the first role and be tuned at first.
- The input scaling factors have an effect for basic sensitivity of the controller with respect to an optimal choice of the input signal operating area and therefore play the second role.
- The tuning of shape and location of membership functions is the matter of fine-tuning. It is therefore almost impossible to drive system into instability by changing the membership functions.
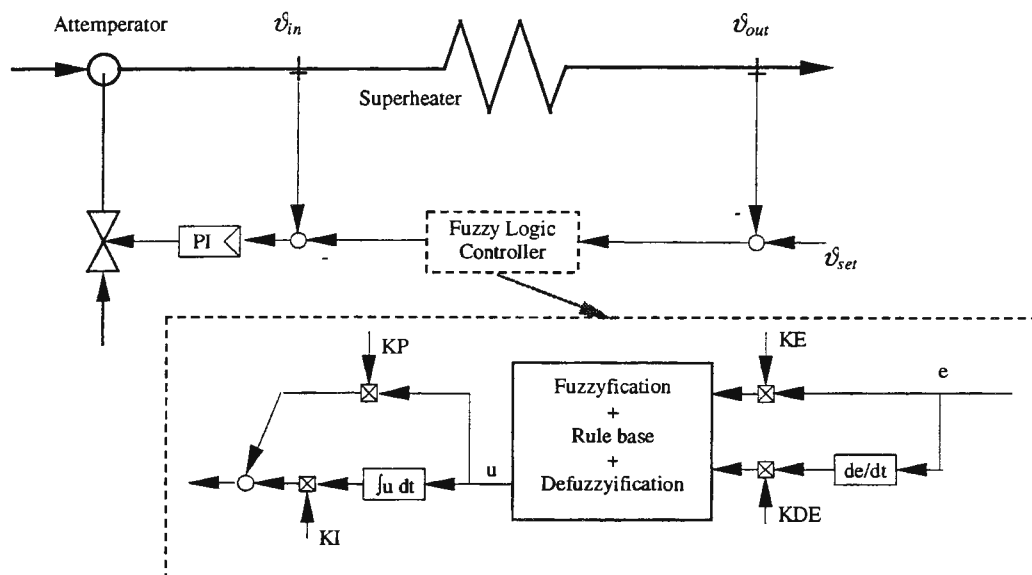
Fig 2. Implementation and dynamic structure of the FLC

Tuning of parameters of the FLC has been performed with respect to rise time and overshoot. Both input gain KE and output gain KP are load - dependent. Derivative and integral gains KDE and KI are constant as well as the static structure of FLC. The membership function distribution and rule base definition correspond to well known *two input - one output* Mamdani structure shown in Fig. 3.

| Membership functions | Control Surface u = f(e,de) |
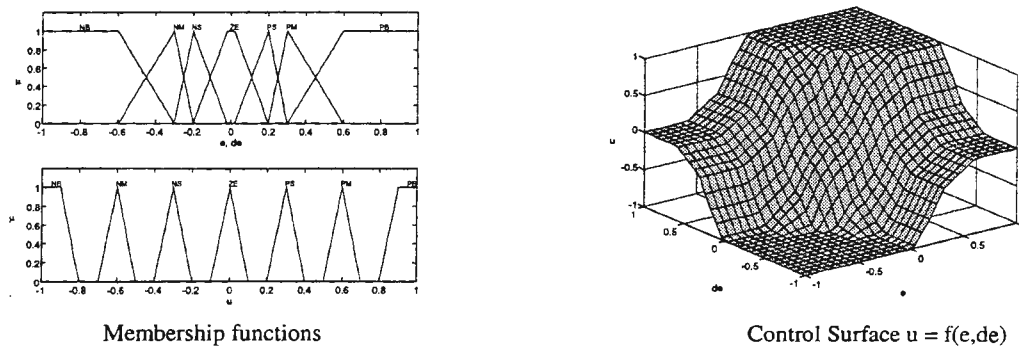
Fig. 3. Realization of the FLC static structure

MATHEMATICAL MODEL

Mathematical model of the superheated line has been obtained considering two separate (yet interacting) flows: the water steam path and flue gas path. The mass, energy and momentum equations have been applied to control volumes into which both paths are divided. Interaction through metal walls has been taken into account by the metal energy equation. The system of the conservation equations has been discretized and implicit scheme applied. Resulting system of nonlinear algebraic equations has been solved by iteration in each time step using modified Newton-Raphson method. The basic concepts underlying modeling and numerical solutions were described in more details in [9]. For the purpose of this work discretized mathematical model has been programmed in the MATLAB® environment enabling excellent compression of the code.

SIMULATION RESULTS

A model validation has been performed by comparing the open loop responses of the superheater outlet temperature to a stepwise change of the attemperator water flow. The results are shown in Fig. 4. Some disagreements between measured and simulated curves are caused by sudden steam flow changes in the period between 350th s and 450th s. Illustration of the varying dinamic behavior of the process is given in Fig. 5. Due to inherent nonlinearities outlet temperature responds quite differently at low and high load levels.
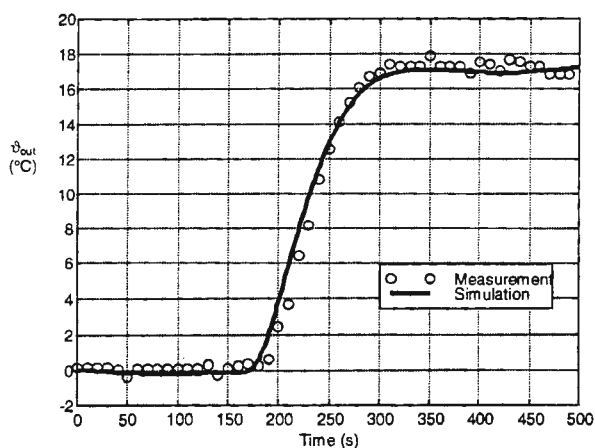
| Fig. 4 | Normalized response to a stepwise change of the attemperator water flow in 160th s. | Fig. 5. | Normalized open loop responses to stepwise change of the attemperator water flow at different load levels |
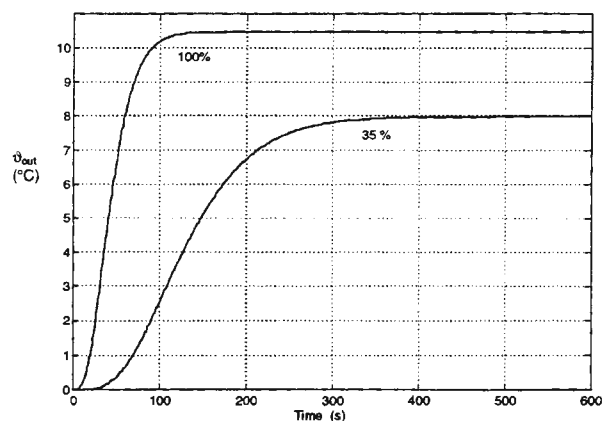
In order to compare the performance of the chosen control concepts various operating conditions have been simulated. Control variable responses (ϑ_{out}) to the set point changes are shown in Fig. 6 and Fig. 8., while simulation results in case of heat rate disturbance are shown in Fig. 7. and Fig. 9. Both situations are simulated for low and high plant load with control parameters tuned appropriately for superheater dynamics. Response of the inlet temperature (ϑ_{in}) illustrates the intensity of water flow changes i.e. intensity of control valve action.
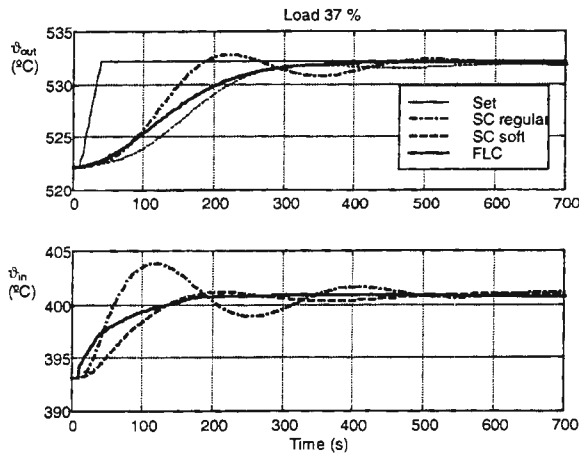
Fig 6.   Closed loop responses to a ramp change of the set point at 37 % load.
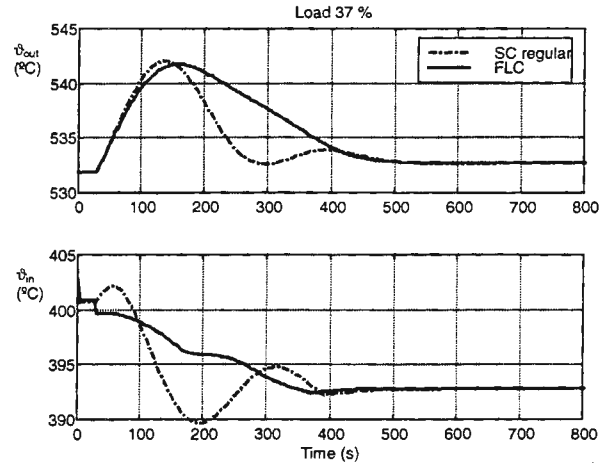
Fig 7.   Closed loop responses to a stepwise change of the heat rate at 37 % load.
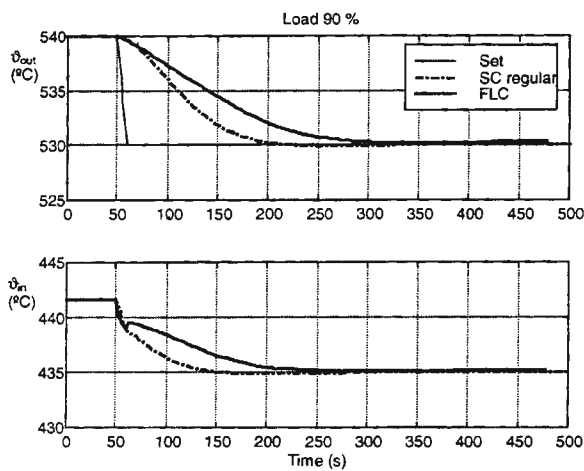
Fig 8.   Closed loop responses to a ramp change of the set point at 90 % load.
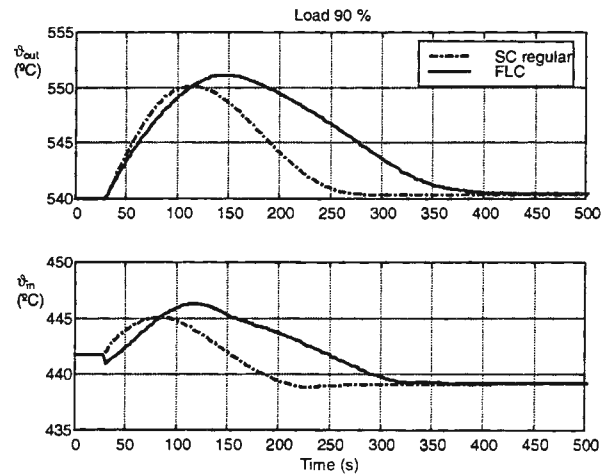
Fig 9.   Closed loop responses to a stepwise change of the heat rate at 90% load

Performance analysis of the simulation results in fig 6. to fig. 9. shows that responses of the fuzzy logic controller are slower than responses of the state controller with regular parameter setting. All fuzzy responses are well damped. The inlet temperature responses show faster and more intensive changes in case of state controller. These responses are in direct connection with the occurrence of thermal stresses in the inlet manifold material.

The robustness of the chosen concepts has been tested in off-design operation. The plant operating point has been increased from 37% to 60% of the nominal load while the control parameters have been left unchanged. The closed loop responses to the characteristic disturbances shown in Fig. 10 and Fig. 11. confirm excellent characteristics of the fuzzy logic controller.
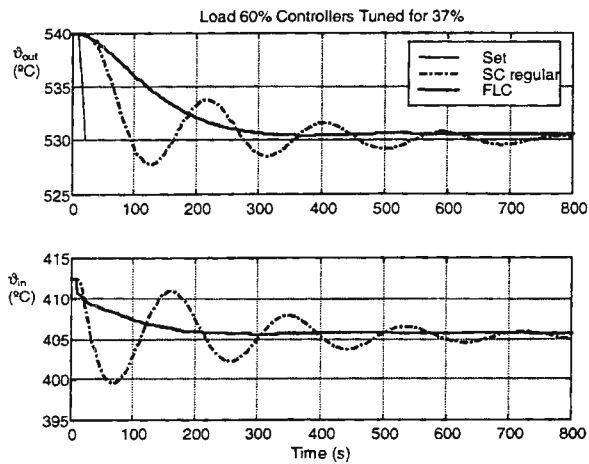
Fig 10. Closed loop responses to a ramp change of the set point at 60% of nominal load, control parameters tuned for 37% load
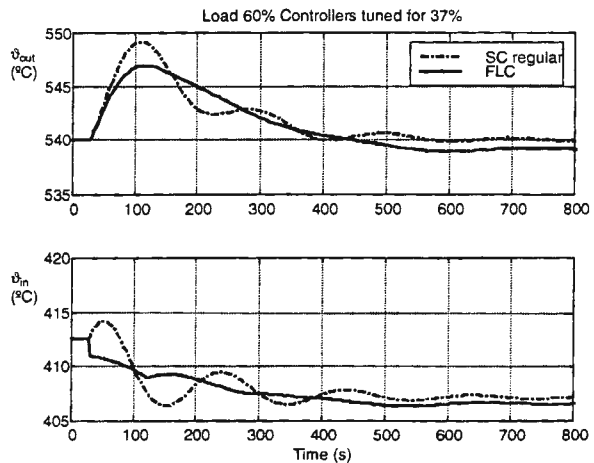
Fig 11. Closed loop responses to a stepwise change of the heat rate at 60% of nominal load, control parameters tuned for 37% load

CONCLUSION

In this simulation study has been shown that the fuzzy logic controller of the relatively simple structure can be competitive to state controller with observer with respect to steam temperature control performance. The well known robustness of the FLC has been proven in this investigation as well.

REFERENCES

1. Bonissone et al, Industrial Applications of Fuzzy Logic at General Electric, *Proceed. IEEE*, pp. 450-465, Vol. 83, No. 3, 1995.
2. Boverie, S., Demaya, B., Lequellec, J.M., Titli, A., Fuzzy Logic Control for High Order Systems, *Proceed. IEEE '93 Int. Conf. Fuzzy Systems*, pp. 117-121, San Francisco, California
3. Chen, C.L., Jong, M.J., Fuzzy Predictive Control for the Time Delayed System, *Proceed. IEEE '93 Int. Conf. Fuzzy Systems*, pp. 236-240, San Francisco, California
4. Demaya, B., Palm, R., Boverie, S., Titli, A., Multilevel Qualitative and Numerical Optimization of Fuzzy Controller, *Proceedings of FUZZ - IEEE '95*, Yokohama, pp. 1149-1154, IEEE 1995.
5. Klefenz, G., *Die Regelung von Dampfkraftwerken*, Bibliographisches Institut Manheim, Wien, Zürich, 1991
6. Lausterer, G.K., Mann, J., Temperature Control State Feeedback in a Fossil Fired Power Plant, Proc. IFAC Symposium on Control of Power Plants and Power Systems, pp. 37-42, Munich, 1992.
7. Maffezzoni, C., Concepts, Practice and Trends in Fossil-Fired Power-Plant Control, Proc. IFAC Symposium on Control of Power Plants and Power Systems, pp. 1-9, Beijing 1986.
8. Sugeno, M., Aoki, S., Kawachi, S., Application of Fuzzy Control logic for Dead-Time Processes in a Glass Melting Furnace, *Fuzzy Set and Systems 38*, pp. 251-265, Nort Holland 1990.
9. Šerman, N., Mavračić, K.: Flexible program package for simulation of once-through boiler steady states and dynamics, ICHMT International Forum On Mathematical Modeling and Computer Simulation of Processes in Energy Systems, Sarajevo, 1989.

KNOWLEDGE REPRESENTATION IN SIMULATION MODELS

FOR ENVIRONMENTAL CONTROL *

András JÁVOR, Gábor SZŰCS

Széchenyi István College
H-9026 Győr, Hédervári u. 3. HUNGARY
javor@goliat.eik.bme.hu

Introduction

The solution of environmental problems has become increasingly important in recent years all over the World since the quality of life is strongly dependent on their treatment. In finding optimal solutions it is unavoidable to investigate highly complex and interdisciplinary systems. This can be solved by using simulation methods.

In this paper an artificial intelligence controlled simulation methodology is introduced for representing and optimizing the models [1]. In the approach presented knowledge bases are assigned to the mobile entities within the models enabling the representation of sophisticated behaviours. This is based on the application of Knowledge Attributed Petri Nets (KAPN) [2]. The heuristic search for determining optimal solutions is undertaken by the application of intelligent demons monitoring the dynamic behaviour of the simulation models and modifying their structure and/or parameters. The methodology is illustrated on two examples, where it was used for solving large scale problems [3][4]. Our simulation experiments were performed by the CASSANDRA (Cognizant Adaptive Simulation System for Applications in Numerous Different Relevant Areas) 3.0 AI controlled simulation system that we have developed based on the results of our research [1][2].

Investigation of air pollution caused by urban traffic

In our first example we have dealt with the air pollution caused by urban traffic. We have built the traffic model of Budapest represented internally with KAPNs and externally with a user friendly representation. The road segments were represented with emission line sources (see Fig. 1.). The numbers, types and speed of the vehicles on these road segments generated the amount of the various pollutants CH, CO and NO_x and these elements provided the emission. The representation of the vehicles by Knowledge Attributed Tokens internally enabled a close approximation of reality by assigning adaptive path finding capabilities to them in a similar way as a human driver finds his or her way in traffic. In Fig. 2. an example of such a knowledge base assigned to a vehicle is shown. According to the traffic conditions various routes can be chosen in accordance with the knowledge base structure in frame form. The line sources taking into consideration the road segment parameters as speed limits e.g. as well as the complex nature of the actual traffic on them provided compact emission sources for the large scale model of 512 road segments in our case.

In order to find optimal solutions traffic light controls, speed limits and traffic rules have been changed by the demons monitoring the simulation of the model i.e. its dynamic behaviour [3].

a) External representation of the road element line source with vehicles

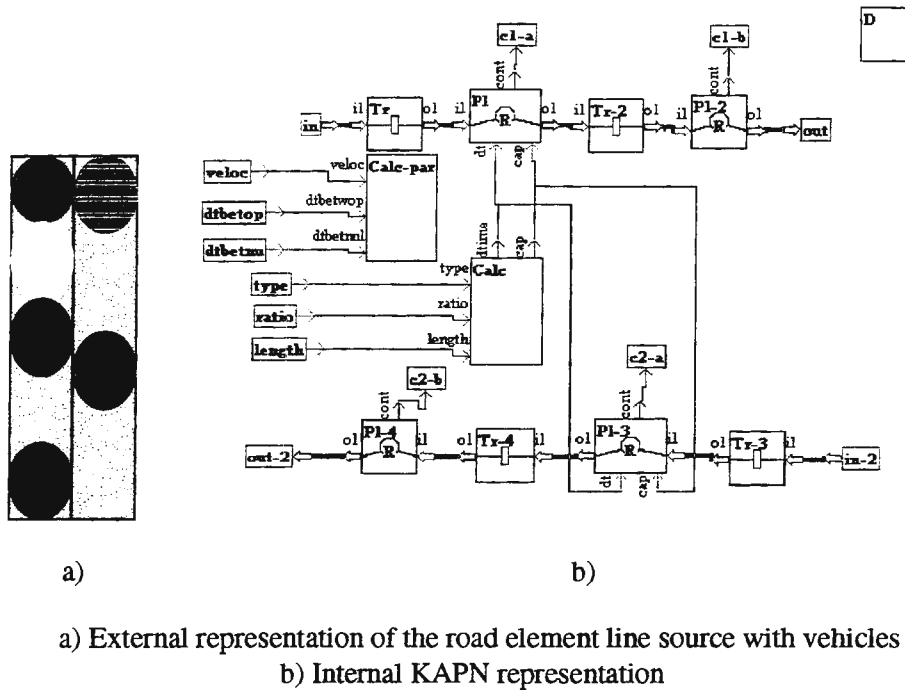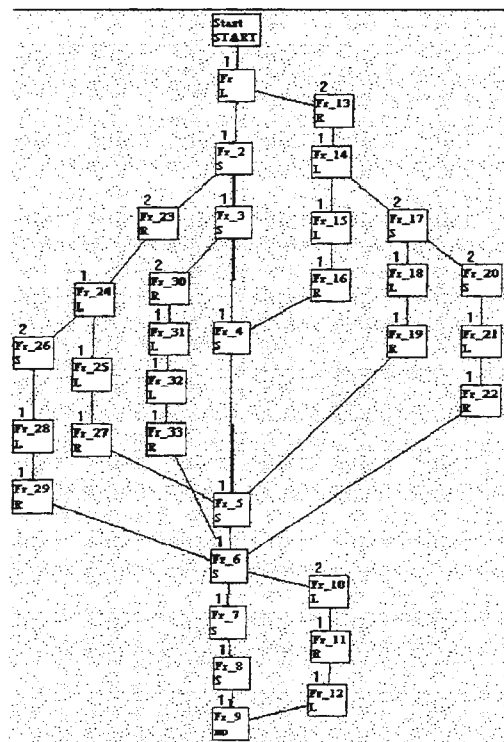b) Internal KAPN representation

Figure 1.

Figure 2. Knowledge base of a vehicle

Simulation of a quality controlled manufacturing plant with recycling

Our second example has been the investigation of a manufacturing plant with high quality assurance. In the factory various thermocontrollers (see Fig. 3.) have been produced, sampling quality control was undertaken in numerous intermediate stages of the production beyond the final control performed on each product. Beyond the main task - executed within the framework of an EUREKA project - i.e. determining the optimal allocation of the intermediate quality controls to achieve minimal cost of the production we have investigated the possibility of recycling.
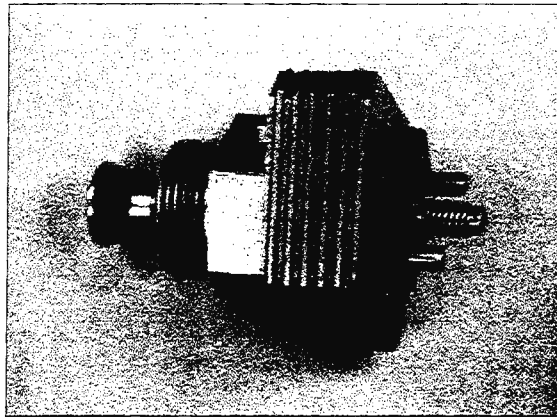
Figure 3. One of the products in the plant investigated
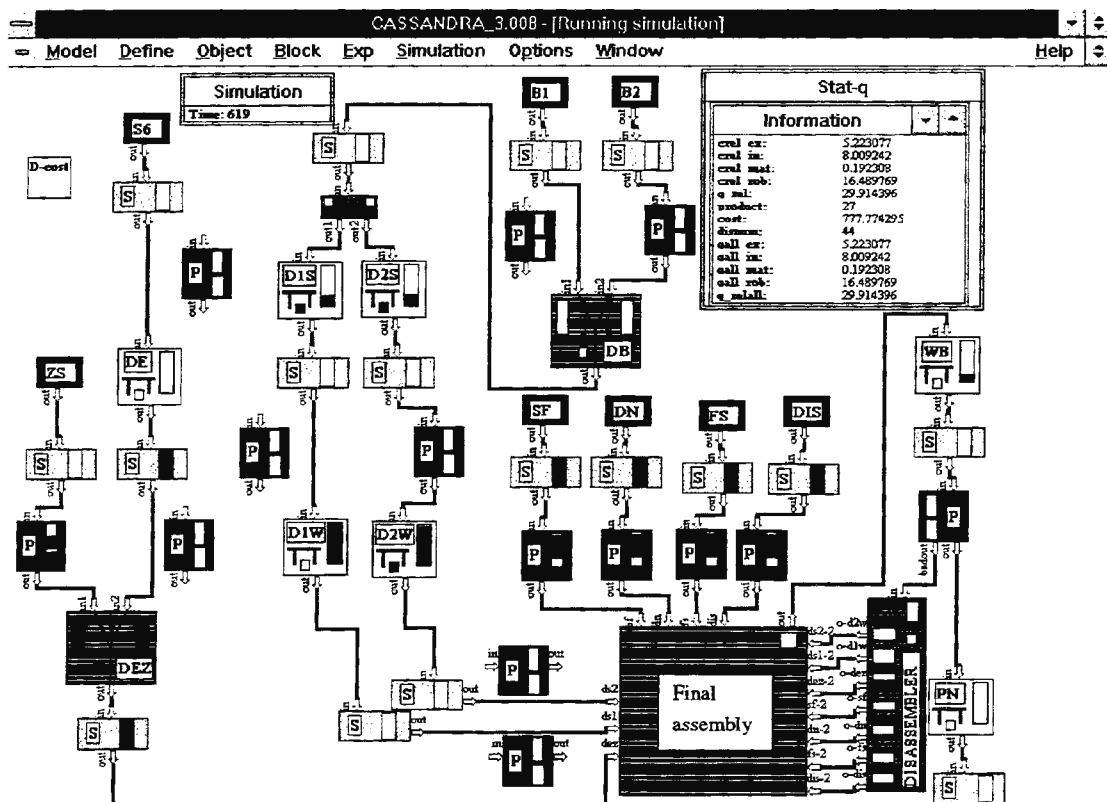
Figure 4. Model with disassembler for recycling

The thermocontrollers consist of a large number of different components. Some of them are supplied from various other firms, while the rest is produced in the factory and their manufacturing process is included in the model (see Fig. 4.). The technological prescriptions determining the route of the products during the manufacturing process has been described by knowledge bases assigned to them; represented internally by tokens within our model. The faults caused during various phases of the process have also been added to the corresponding parts (in our case to the appropriate slots of the frame) of the knowledge bases of the products. In Fig . 4. the block diagram of the simulated model of the manufacturing plant can be seen.

A number of experiments have been performed for determining possible strategies of optimization. The investigations for finding optimal solutions were performed by means of automating the procedure of approximation by a series of simulation experiments. This has been executed by artificial intelligence controlled demons able to modify the model structure and parameters. After having found the optimal solution for the production quality control we have assigned the task for finding the rate for disassembling faulty products after the final quality control and re-entering the faultless parts into the final assembly process. The precondition was to find a solution where this process still did not decrease production costs and at the same time provided the advantage of decreasing the environmental pollution produced by the garbage created by the faulty products that had to be thrown away during the manufacturing process. This resulted in a reduction of environment pollution without excessive costs.

Conclusions

The solution of environmental problems is usually a compromise between the contradictory demands of financial means, services and production on the one hand and reducing the demage of the environment on the other. We have been convinced that instead of arguments based on biased feelings it is advantages to find optimal and feasible solutions by using simulation. The two examples presented in this paper were aimed at providing such approaches.

References

[1] Jávor, A.: Demon Controlled Simulation
        Mathematics and Computers in Simulation 34(1992)283-296.
[2] Jávor, A.: Knowledge Attributed Petri Nets
        Systems Analysis, Modelling, Simulation, 13(1993)1/2, 5-12.
[3] Jávor, A., Szűcs, G.: Traffic Simulation using AI
        IMACS European Simulation Meeting on Simulation Tools and Applications,
        28-30 August, 1995, Győr, Hungary, 76-81.
[4] Szűcs, G., Vigh, Á.: Simulation of a Quality Controlled Manufacturing and Assembly System
        IMACS Symposium on Systems Analysis and Simulation,
        26-30 June, 1995, Berlin, Germany, 279-282.

SIF: an Agent-Based Simulation Toolbox

using the EMS Paradigm

P. Funk, C. Gerber[1], J. Lind, M. Schillo

German Research Center For Artificial Intelligence (DFKI),

Im Stadtwald, Building 36, 66123 Saarbrücken, Germany,

{funk, gerber, lind, schillo}@dfki.de

Abstract

There are many different approaches to the task of realising a simulation, ranging from using an off-the-shelf simulation tool to implementing a simulation from scratch. In this paper, we propose an approach to simulation based on the use of *multi-agent systems* developed in the discipline of Artificial Intelligence (AI). The field of multi-agent systems is concerned with computational systems composed of several autonomous agents capable of mutual interaction in an artificial or real environment. We present the Social Interaction Framework *SIF* [Funk et al. 1998] and demonstrate its suitability for social simulation as it naturally allows for modelling of processes at both, the individual (micro) and society (macro) level. SIF implements the EMS (Effector-Medium-Sensor) paradigm, which provides a generic agent-world interface.

Motivation: Why Use Multi-Agent Systems?

According to K. Troitsch [Doran et al. 1995] simulation tools seem to be used mainly for two rather conflicting purposes: On the one hand, for the better understanding of some observed real-world phenomenon and on the other hand, for advice and decision support. In order to satisfy these diverging demands, a whole variety of simulation approaches has been developed since the early 1960s.

Pre-built products such as the classic flow diagram system from the 1960s, GPSS (General Purpose Simulation System) of [Gordon 1969] or more recent graphic-oriented Visual Interactive Modelling Systems (VIMS) provide built-in functionalities leading to little modelling effort. Furthermore, these systems require only little programming skills. However, limitations of such tools are the rather low execution speed, the restricted expressivity leading to insufficient scalability as only few entities can be modelled. On the other extreme, the most flexible approach is do-it-yourself-coding, since the developer has all freedom to model entities. But such an approach requires detailed programming skills and rather time consuming implementation and debugging.

Specialised simulation programming languages and pre-written simulation libraries for general-purpose languages are located somewhere between the extremes. The simulation designer still has a high degree of flexibility and can make use of features pre-specified in the system. Such systems allow easy access to simulation modelling for novice users as well as more complex simulation experiments for experts.

A recent development in Artificial Intelligence is *Agent-Oriented Programming* (AOP) [Shoham 1991], derived from Object-Oriented Programming (OOP): the entity in focus is a so-called *agent*, a unit able to autonomously plan and perform actions. Some approaches of AOP see communication between agents as a special case of acting, whereas others regard acting and communication as different forms of interaction. Tasks are performed in a distributed fashion through cooperation between agents. With SIF, we present a JAVA™ library which enables agent-oriented programming for quick and platform-independent simulation modelling. SIF provides a multi-threaded framework where basic agent functionalities (such as sensing,

---

1

performing actions, communication, etc.) are already provided. The simulation designer only needs to specify the agents' decision making procedures. This can be done from scratch or by evoking of reference procedures.

Simulations based on multi-agent systems can be used to model a broad variety of domains because multi-agent systems support simulations of any granularity:

- In *Micro-Level Simulations* agents are used to represent individuals. An agent's decision making procedure can be defined in a primitive way, only depending on a certain state of the agent and on states of *neighbour agents*. In such a case, SIF basically works as a cellular automaton [Axelrod 1984]. However, more complex cognitive processes, such as logic-based inference capabilities or even "illogical" emotions can be modelled leading to a much more realistic simulation of human behaviour.

- In *Macro-Level Simulations* agents represent whole individual groups or societies. Hence, macro-level properties can be modelled effectively. Through combination of micro- and macro-agents the behaviour of individuals in societies can be simulated easily leading to a natural bridge of the micro-macro-gap.

The remainder of the paper is structured in the following way: in the next section we introduce the EMS paradigm. This model is the underlying theory of the SIF simulation system, which is described in the third section. Finally, we give a conclusion.

The EMS Model of Interaction

A very general and broadly accepted definition of an agent is provided by [Russell and Norvig 1996]:

An *agent* is an entity that can be viewed as *perceiving* its environment through *sensors* and *acting* upon that environment through *effectors*.

The so called *Effector/Medium/Sensor paradigm* (for short EMS) reflects this definition. This paradigm provides an appropriate abstraction of a (human) agent acting and interacting with its environment and other (human) agents. This idea is related to Malsch's work on *generalised media of interaction* in sociology [Malsch and Schulz-Schäfer 1997]. Here action and interaction are transmitted via appropriate media.

According to EMS, verbal communication – as an instance of action and interaction – is modelled in the following manner: each agent (human) has a *speech effector* (voice, speech apparatus) and an *audio sensor* (ear). The spoken language, respectively the sound, is transmitted through the air. We call this transmitter the *medium*. Figure 1 shows this abstraction.

The EMS paradigm reflects this definition. In general, an agent according to the EMS paradigm is equipped with a number of effectors through which it can act upon its virtual environment and with a number of sensors through which it can perceive the state of the environment (see Figure 2).
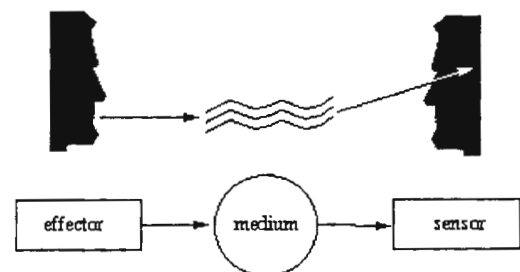

Figure 1 The EMS model of interaction

The EMS allows to model groups of media and sensors/ effectors belonging to those media. For example, we formalise communication by using an air-like medium comprising message sending effectors and receiving sensors. To simulate e.g. agents moving around in a virtual world, the library provides media which reflect the topology of the environment.

2

The Simulation Toolbox SIF

In this section we describe the Social Interaction Framework (SIF) that implements the EMS paradigm. SIF is a JAVA™ library which supports agent-oriented programming. It consists of a scenario-independent simulation engine and a number of ready-to-use components. SIF contains a mechanism to specify simulations in a simple script language. These scripts can be loaded, modified and reloaded by using the graphical user interface (GUI) without need to shut down the SIF system. In the following paragraphs we first describe the architecture of the SIF library and then we explain the control and information flow in the system during run-time.
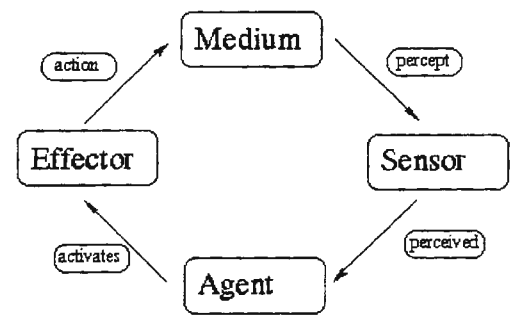
Figure 2 The EMS model of interaction

The SIF Architecture

The EMS-based implementation consists of the following modules: the *world server*, a variety of *agents* situated in the simulation environment and the *graphical user interface*. These parts run independently from each other in a JAVA™ thread of their own.

World Server: The world server updates the world representation and organises the flow of information during the simulation. It starts and stops the simulation and provides debugging and data collection facilities. For creating individual scenarios, it is necessary to specify two parts of the world server: the *action manager* and the *sensor manager*. The *action manager* operationally defines the state transition of the world for every action. The *sensor manager* processes the data in the world representation to compute the input of the sensors according to the specifications of the scenario.

Agents: An agent can possess different types and any number of sensors and effectors. The toolbox offers agents that can move around, pick up resources, communicate, etc. Due to the generic design, any type of agent can be connected to the world server ranging from simple JAVA™ objects to sophisticated InterRRaP [Müller 1996] agents.

Graphical User Interface: The GUI is linked to the core of SIF just like an agent, i.e. it communicates with the core via sensors and effectors. The percepts that it receives tell the GUI e.g. to update certain parts of the visualisation or to display agent information. The GUI offers a control pad which is capable of forcing agents to perform certain actions. It also provides information windows for easy access to internal data of agents.

Control and Information Flow in SIF

The control and information flow in a SIF simulation is implemented according to the EMS paradigm (see Figure 3). To illustrate the information flow we assume that the simulation has been running for a while. We start the description of the control cycle at the point where a certain agent (upper left box in Figure 3) is about to choose its next action to make. As soon as the decision making process of the agent has produced a result, the agent activates the corresponding effector. For instance, if an agent has decided to move forward, it activates the *move effector*.

In the simulation, the effector wraps the information of what is intended to happen into an *action object*. The world server stores all incoming actions in a first-in-first-out queue. In each step of the simulation, the action manager (as part of the world server) picks the next action from the queue and accesses the world representation to find out whether the action is legal or not. If it is legal the action manager will update the world representation accordingly. Furthermore, the action manager commits the action and sends a percept to the GUI, informing it that a part of the world has to be redrawn before it passes control to the sensor manager. If the action is not legal in a particular situation the action manager will directly pass control to the sensor manager.

The sensor manager then computes the input for all sensors that are affected by the change of the world. The input is sent in form of percepts (reflecting the local point of view of the agents) to the sensors in question.

Now one cycle for the agent is completed and it can perform the next action. As a side effect of the asynchronous implementation of SIF the agent does not have to wait for this feedback from the world.

The connection of the GUI to the queue of actions illustrates an information flow that occurs when either the experimenter uses a control pad (which is part of the GUI) to force an agent to execute a particular action or when the experimenter wants to start or stop the experiment. In the first case, an action containing the command is posted to the queue. These commands are passed to the agent and force it to execute the respective action. In the second case a command action is sent to the action manager to start or stop the control flow in the simulation.
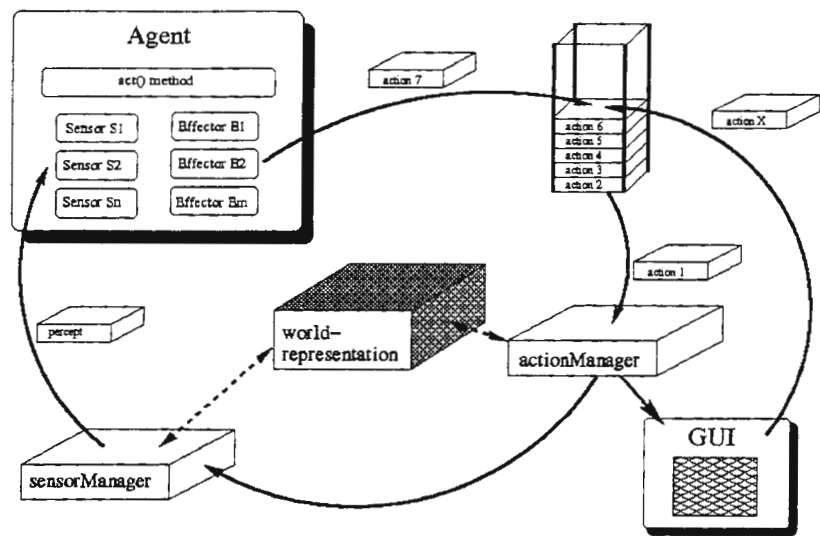
Figure 3 The information flow in SIF

Conclusion

In this paper we have outlined a general model for agent interaction (EMS paradigm) and presented a simulation library (SIF) based on this model. The library supports agent-oriented programming (AOP) which facilitates rapid prototyping of asynchronous multi-threaded simulations. Further work will focus on extending SIF to a distributed testbed using TCP/IP and enhancing the implementation of time restrictions of actions.

References

[Axelrod 1984]   Axelrod, R. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[Doran et al. 1995]   Doran, J. et al. *Social Science Microsimulation: A Challenge to Computer Science*. Dagstuhl-Seminar-Report 112. D-66687 Wadern, Germany, 1995.

[Funk et al. 1998]   Funk, P., Gerber, C., Lind, J., Schillo, M. *Social Interaction Framework – A Generic Testbed for Social Agents*. Research Report, German Research Center for Artificial Intelligence. To appear, 1998.

[Gordon 1969]   Gordon, G. *System Simulation*. Prentice-Hall, Englewoods Cliffs, NJ, 1969

[Malsch and Schulz-Schäfer 1997]   Malsch, T. and Schulz-Schäfer, I. Generalized Media of Interaction and Interagent Coordination. In: *Socially Intelligent Agents: Papers from the 1997 AAAI Fall Symposium*. Technical Report FS-97-02, Cambridge, Massachusetts, 1997.

[Müller 1996]   Müller, J. P. *An Architecture for Dynamically Interacting Agents*. PhD thesis, Universität des Saarlandes, Saarbrücken, 1996.

[Russell and Norvig 1996]   Russell, S. and Norvig, P. *Artificial Intelligence, A Modern Approach*. Prentice-Hall, 1996.

[Shoham 1991]   Shoham, Y. Agent-oriented programming. In *Proceedings of the 11th International Workshop on DAI*, pages 345-353, 1991.

4

The use of genetic algorithms in modelling the multivariable control problem

Maja Atanasijević-Kunc, Rihard Karba, Borut Zupančič, Aleš Belič
Faculty of Electrical Engineering
Tržaška 25, 1000 Ljubljana, Slovenia
e-mail: maja.atanasijevic@.fe.uni-lj.si

Abstract

In the paper the laboratory three - tanks system is presented for which nonlinear and linearised models were built and evaluated through the whole modelling procedure. The main goal of modelling was to obtain the models with which good prediction of the closed - loop behaviour could be expected. It has been found out that nonlinear properties of the system components involve nonanalytical description, which is of course important in the controller design step. The definition of approximate decoupling of multivariable system is then given to be used in the proposed design approach. This approach is a combination of different design methods where also genetic algorithms have proven to have good potential. The main point in solving the problem represents the fact that information of partial design can be used in the next steps to help in selection between different design algorithms and/or structures or as the input data to the chosen design procedure.

1. System description

The process (made by AMIRA [1]) is schematically shown in Fig. 1. The activators are two pumps with which the levels in two of three tanks can be controlled, but level measurement is realised in all three tanks.
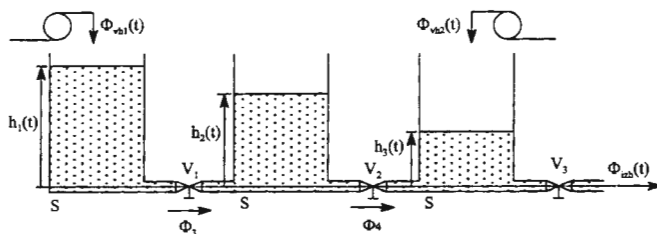
Figure 1: Schematic representation of the process.

In the first step some frequently used assumptions have been introduced, like: the water flows through the valves are square root functions of pressure difference and activators and sensors have linear characteristics in the whole operating range. The obtained nonlinear and linearised models were tested in open and closed - loop, where only two SISO - PI controllers were used. Comparison with the measurement results have proven mismatching which was very transparent in the closed - loop behaviour especially regarding the control signals. Controlled outputs (the levels in the first and the third tank) were in fact similar due to the controller action, but control signals had to act differently due to the mentioned differences between the models and the real plant. This had stimulated additional modelling efforts which have shown that both pumps and output valve V3 exhibit nonlinearities as it is shown in Fig. 2 to 4, for which good analytical descriptions do not exist in the whole operating range. The resulting nonlinear model was obtained in the following form:

$$\Phi_{vh1}(t) - K_1\sqrt{h_1(t) - h_2(t)} = S\frac{d[h_1(t)]}{dt} \tag{1}$$

$$K_1\sqrt{h_1(t) - h_2(t)} - K_2\sqrt{h_2(t) - h_3(t)} = S\frac{d[h_2(t)]}{dt} \tag{2}$$

$$K_2\sqrt{h_2(t) - h_3(t)} + \Phi_{vh2}(t) - \Phi_{izh}(t) = S\frac{d[h_3(t)]}{dt} \qquad (3)$$

where for both input flows $\Phi_{vh1}(t), \Phi_{vh2}(t)$ and for the output flow through valve V_3 the characteristics in Fig. 2 to 4 were used. For sensors the following linear description was accepted:

$$x_1(t) = y_1(t) = K_{s1}\, h_1(t) + K_{os1} \qquad (4)$$

$$x_2(t) = K_{s2}\, h_2(t) + K_{os2} \qquad (5)$$

$$x_3(t) = y_2(t) = K_{s3}\, h_3(t) + K_{os3} \qquad (6)$$

Parameters in the given equations were defined with: $S = 0.0154m^2$, $K_1 = 1.05\ 10^{-4}m^{1.5}/sec$, $K_2 = 1.03\ 10^{-4}m^{1.5}/sec$, $K_{s1} = -27.6937V/m$, $K_{s2} = -27.5827V/m$, $K_{s3} = -27.5087V/m$, $K_{so1} = 9.1500V$, $K_{so2} = 9.6000V$, $K_{so3} = 9.0000V$.
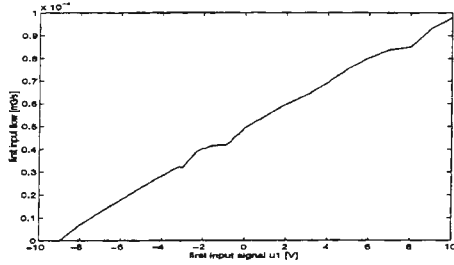
Figure 2: Characteristic of the first pump.   Figure 3: Characteristic of the second pump.
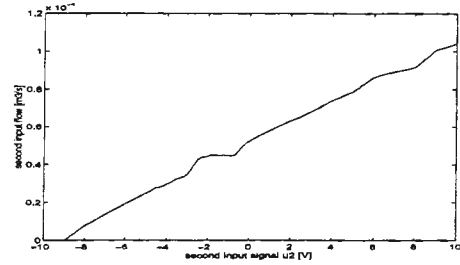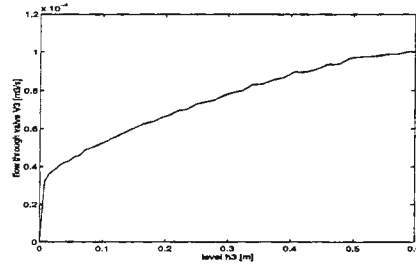
Figure 4: Characteristic of the output valve V_3.

From described nonlinear model it was found out that linearised approximation is justified around the working point which is defined with input signals: $\bar{u}_1 = -4V$, $\bar{u}_2 = -3.5V$. Obtained linearised description is in state space description is given with:

$$\Delta\underline{\dot{x}} = \begin{bmatrix} -0.0125 & 0.0126 & 0 \\ 0.0125 & -0.0246 & 0.0121 \\ 0 & 0.0120 & -0.0212 \end{bmatrix} \Delta\underline{x} + \begin{bmatrix} -0.0091 & 0 \\ 0 & 0 \\ 0 & -0.0092 \end{bmatrix} \Delta\underline{u}$$

$$\Delta\underline{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta\underline{x} + \underline{0}\,\Delta\underline{u} \qquad (7)$$

where Δ denotes the deviations of variables from working point.

2. Control design

We have mentioned that some acceptable control action like stability and zero steady state to step changes can in this case be obtained with only two SISO PI controllers for each control loop. For

testing the model accuracy in the closed - loop these controllers were chosen to have the following parameters:

$$\underline{\underline{G}}_{PI}(s) = \begin{bmatrix} -3.6230 + \frac{-0.0181}{s} & 0. \\ 0. & -10.9200 + \frac{-0.1092}{s} \end{bmatrix} \tag{8}$$

Further we wanted to examine the possibilities to improve obtained control action but primarily in the sense of reduction the existing cross couplings. In the same time the control structure should be as simple as possible, which means that its order and the degree of freedom should be optimised. A great number of decoupling techniques and algorithms can be found in existing literature. From the structural point of view they can be separated into two main groups. Compensation approach (like [5]) usually accepts one degree of freedom structures. Very often this results in high order control structures where complexity is comparable with the system inversion. In many other cases two step design can be recognised [2-4] where the first is mainly devoted to decoupling while with the second some final design goals of the overall process are satisfied. In this case more then one degree of freedom controller is usually needed. From the design procedure point of view some relative advantages can be found in the second approach as the goals to be satisfied can also be in some degree separated into both steps.

It is not difficult to found out that some good results can be expected also with the so called interactive design techniques. If obtained closed - loop time constants are small then also transient responses in cross - couplings can be satisfactory small. But it is important to notice that this doesn't mean also small interactions as it can be found out from the frequency response. On the contrary. Very effective multivariable controllers usually involve also strong couplings of the overall process. If interactions are in this case still too large, design procedures usually don't give direct answer to this problem. It is of course always true that models are not exact but only approximate description of process properties. That means that also reduction of cross couplings with the exact decoupling controller is not exact but depends on the model quality. In this context the definition of approximate decoupling can be described in the following manner: The method or design algorithm can be called approximate decoupling if design includes also some explicate mechanisms to reduce the connections between chosen input - output variables. In our case this so called approximate decoupling was realised in the following manner. As nonlinear analytical description of the process is in this case not known and decoupling in the whole operating range is not sensible we started the design with the decoupling controller which was obtained on the basis of linearised description (7). Free parameters of the realised control law [2,3]:

$$\Delta \underline{u} = \underline{\underline{F}} \, \Delta \underline{x} + \underline{\underline{F}} \, \Delta \underline{\omega} \tag{9}$$

where chosen in such manner that time constants of the decoupled system were similar to the situation, where only two SISO PI controllers were used. Matrices $\underline{\underline{F}}$ and $\underline{\underline{G}}$ where so chosen to be:

$$\underline{\underline{F}} = \begin{bmatrix} 0.8242 & 1.3846 & 0 \\ 0 & 1.3043 & -0.1304 \end{bmatrix}, \quad \underline{\underline{G}} = \begin{bmatrix} -2.1978 & 0 \\ 0 & -2.1739 \end{bmatrix} \tag{10}$$

With this controller the closed - loop linear system was completely decoupled and also simplified into two equal subsystems of the first order (the shortest path through the system was achieved) due to the fact that the process possesses one stable transmission zero. To improve steady state accuracy two univariable PI controllers were further added in such manner that transfer function matrix remained unchanged as:

$$\underline{\underline{G}}_z(s) = \begin{bmatrix} \frac{1}{50s+1} & 0 \\ 0 & \frac{1}{50s+1} \end{bmatrix} \tag{11}$$

Simulation results of this control structure in comparison with measurement data are shown in Fig. 5 to 8, where start up was first realised and the behaviour around the working point is illustrated with

large reference changes. Comparison with the situation where only SISO PI controllers were used have shown that both closed - loop systems have approximately the same settling time with the important difference that interactions were in the decoupled case reduced to the level of measurement noise in very wide range around the operating point. The attenuation of measurement noise was better in the decoupled case and control signals needed for the decoupled process were smaller due to the multivariable nature of the controller which means that this structure enables also faster control action. Design procedure to obtain desired properties is in this case very straight forward and the goals can be directly connected to input parameters of the design algorithm. The important difference is of course also the fact that in the decoupled case the measurement of all system states is needed which is in general, especially in process industry not possible. So our further goal was directed in exploring the possibilities to transform obtained control action into the one degree of freedom controller of as simple structure as possible. For this it is important to notice that exact equivalence doesn't exist as the decoupling compensator has made closed - loop process partly unobservable. This property could be changed by introduction of additional dynamic structure to the original system in such manner that all controllability subspaces would be independent, but in this case the shortest path from each input to corresponding output would be lost. Additional dynamic structure would be needed for transformation these results so, that only output feedback would be needed.

To avoid unnegligible enlargement of the controller order the optimisation approach with genetic algorithms (GA) was involved in the following manner. In the first step pole placement approach was used to establish the structure which enables exact pole locations to be specified. In our case this is possible with multivariable PI structure. Then GA were used in the optimisation procedure were parameters of the desired control structure have to be found. The most important factor (in spite of the great number of input parameters) which influences the success of optimisation procedure is the definition of fitness function. This was in our case realised on the basis of comparison of control signals from the previous (desired control activity) solution and the solution with desired multivariable PI structure. It is important to point out that the starting set of parameters were randomly generated that means that no a priori knowledge of stabilisable controller was needed as it is usually the case. On the basis of this procedure very good matching was obtained with the following controller:

$$\underline{G}_{PI1}(s) \quad = \quad \begin{bmatrix} -46.7457 & -44.0255 \\ 0.5363 & -38.9919 \end{bmatrix} + \frac{1}{s} \begin{bmatrix} -0.9758 & -0.7860 \\ -0.0047 & -0.7668 \end{bmatrix} \tag{12}$$

The results are illustrated in Fig. 9 to 12 for linearised model on the basis of which optimisation was realised.

3. Conclusions

In the paper modelling and control design is presented for the three tanks system. During modelling procedure control signal comparison was first used to improve the nonlinear model which further led to more suitable linearisation conditions. In the closed - loop design special attention was devoted to approximate decoupling where control signal comparison was again used in the optimisation procedure with genetic algorithms. In this step desired control action was first defined with decoupling and SISO controllers and further transformed into very similar control action where less measurement points and only one degree of freedom controller were needed. The structure of the controller was defined with the pole placement approach which can guarantee that stability is reachable but no initial parameter values were needed in this case. Obtained approach has also the advantage that desired range of control action is incorporated directly into optimisation procedure.

References

[1] AMIRA, DTS 200 Laboratory Setup Three - Tanks - System, Manual, 1995.

[2] Peter L. Falb and William A. Wolovich: Decoupling in the Design and Synthesis of Multivariable Control Systems, IEEE Transactions on Automatic Control, Vol. AC-12, No. 6, December 1967, pp. 651-659.

[3] Elmer G. Gilbert: The Decoupling of Multivariable Systems by State Feedback, SIAM J. Control, Vol. 7., No. 1, February 1969, pp. 50-63.

[4] J. W. Howze and J. B. Pearson: Decoupling and Arbitrary Pole Placement in Linear Systems Using Output Feedback, IEEE Transactions on Automatic Control, December 1970, pp. 660-663.

[5] A. I. G. Vardulakis: Decoupling of Linear Multivariable Systems by Unity Output Feedback Compensation, Int. J. Control, 1989, Vol. 50, No. 4, pp. 1079-1088.
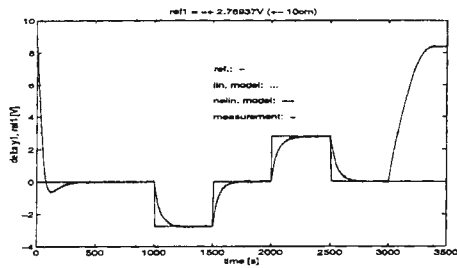
Figure 5: First output.
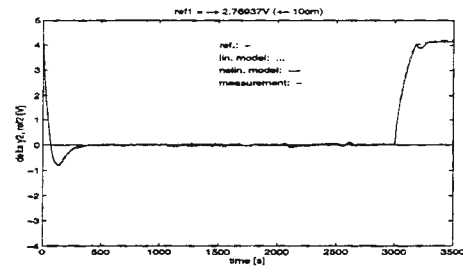
Figure 6: Second output.
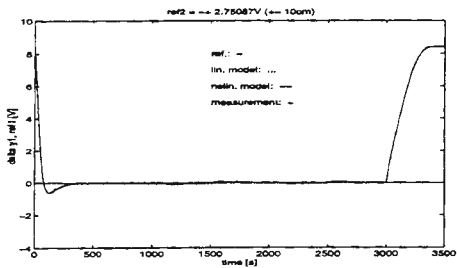
Figure 7: First output.
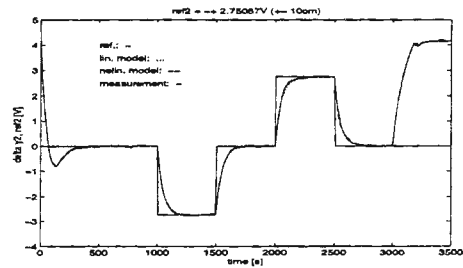
Figure 8: Second output.
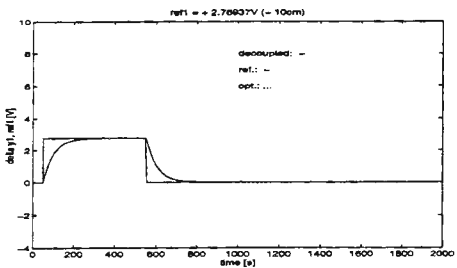
Figure 9: First output.
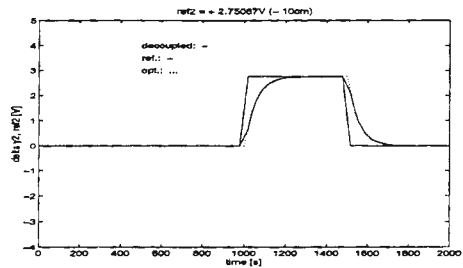
Figure 10: Second output.
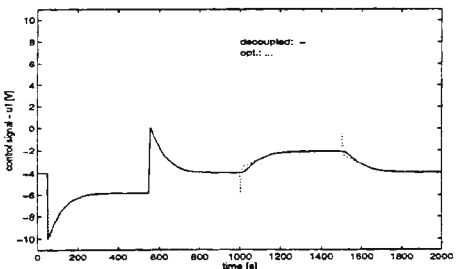
Figure 11: First control signal.
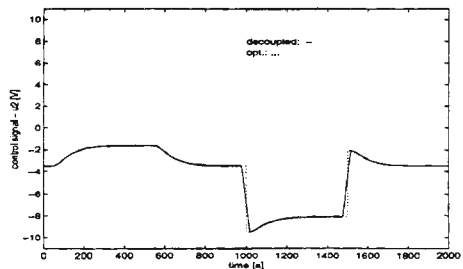
Figure 12: Second control signal.

Modeling Intelligent Decision Making on Electronic Mail Distribution Systems

Alcino Cunha
alcino@di.uminho.pt

Orlando Belo
obelo@di.uminho.pt

Alexandre Santos
alex@di.uminho.pt

Departamento de Informática
Escola de Engenharia
Universidade do Minho
Largo do Paço, 4709 Braga Codex, PORTUGAL

Keywords: Electronic Mail Distribution Systems Modeling, Economic Decision Making, Multi-Agent Systems, Concurrent Engineering, Negotiation.

Extended Abstract

Actual *Electronic Mail Distribution Systems* (EMDS) assume a relevant role in most organizations. Activities related to information exchange, product ordering, enterprises' personnel communication, or thematic discussion zones use them as a primer form of communication. Their simplicity of use, low communication costs, easy installation, and the ability to deal with almost every kind of system's information (e.g. text, images, sound) make them powerful and useful tools in human daily activities. There are cases where commercial viability of enterprise's products are affected by the availability, qualification and effectiveness of the EMDS installed, making them critical resources.

EMDS's activities are supported by a very diversified group of computational resources. Host computers, routers, bridges, gateways, and high sophisticated communication lines are examples of some of those resources. Their permanent availability, twenty four hours a day, ensures EMDS's normal operation and guarantees correct message deliverance inside organizations and

worldwide. Several EMDS models were designed and developed. However, it is possible to find a generic (sub-)model that all of them have included in their functional structures. Such model is basically made of two subsystems: [1]:

- a *Message Transfer System* (MTS), which integrates one or more *Message Transfer Agents* (MTA) responsible for message transportation;

- the *User Agents* (UA), that act on behalf of the users and ensure interface to the local MTA.

Additionally, we may add a third subsystem constituted by the *Administrative Agents* (AA), that controls and supervise MTA's activities.
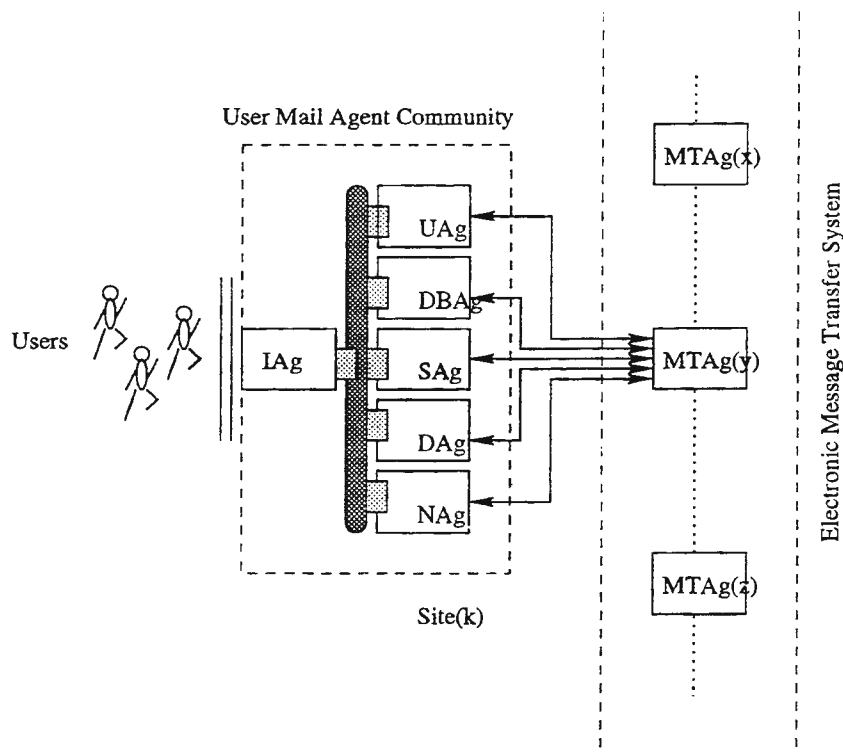
Figure 1: An Electronic Mail Distribution System Model.

In our case we designed a EMDS model (Figure 1) based on the *MH* system - a "Multifarious" User Agent [6] - to which we added some new features. MH is a

little more than an UA, it is an electronic mail message handler with several user interface facilities. MH's model considers a group of small programs, designed as agents, that have associated well determined functions. Beyond the user interface agent, basically it is organized into four components:

- a *Database Agent* (DBAg)- that manages the mail messages that the user received;

- an *User Agent* (UAg) - that provides submission and delivery services;

- a *Directory Agent* (DAg) - that provides for name to address mapping;

- a *Security Agent* (SAG) - that implements authentication and privacy services over mail messages if required.

All these agents were considered in our model. However, they are now included in a user mail agent community sharing a common environment with two other agents:

- a *Negotiation Agent* (NAg) - responsible to develop negotiation processes with the MTA in order to obtain better transportation conditions at lower prices, according to the user mail agent community status and needs of transportation;

- a transformed *Interface Agent* (IAg) - that basically acts as the MH's one but now having conditions to operate as an autonomous agent integrated in a cooperative agent community.

The agents communicate with each other through messages. They act according their own competences, cooperating with each other every time they have inter-related tasks. The reasons that lead to the design of an user mail agent community, or an user mail *Multi-Agent System*(MAS) [7] [5], are related to the high levels of modularity, decentralized control, scalability and opportunistic behavior that any conventional MAS allows.

In this paper we do not intend to analyze and discuss the advantages or disadvantages of the presented model. We have imported concepts, adopted parts of a model, designed new components and proposed an EMDS model based on MAS technology with a simple idea in mind: to apply economic decision approaches, conciliated with the ones related to contract net protocols, in order to get an user mail agent community made of intelligent agents with

the aptness to reach lower mail transportation costs. Thus, this paper will focus on the characterization of the negotiation processes that NAg is able to develop with its local MTA on the MTS. The remaining agents are necessary to support ordinary mail processes and to deliver to NAg the information it needs.

The MTS usually involves several *Service Providers* (SPs). Although the connection between a MH system and a MTA is controlled by just one SP, we will assume that the NAg can contact any of the other SPs in order to determine dynamically the best provider to deliver a particular mail message. The protocol that is used by the NAgs when choosing the best SPs is the *Contract Net Protocol* (CNP) [10, 4]. This protocol tries to reproduce the negotiations that occur in real markets when a particular entity wants to find the best partner to execute a task. In this protocol we may find two kinds of agents:

- *Managers* - the agents that announce the tasks to be executed (in the *announcing* phase) and afterwards select the best partners according to the bids received (in the *awarding* phase);

- *Contractors* - the agents that execute the tasks and that answer to the announces with a bid that reflects their suitability to their execution (in the *bidding* phase).

In this particular instance of the protocol the *manager* roles will be assumed by the NAgs and the *contractor* roles by the SPs.

The economic principles are applied in the decisions that must be made in the *announcing*, *bidding* and *awarding* phases of the CNP. The use of economic principles in this problem can be justified by the fulfillment of the following conditions [11]:

- the fundamental problem to be solved is one of resource allocation;

- all the involved agents act rationally in order to achieve their most preferred outcomes;

- the decision making is inherently decentralized.

We will not address the *bidding* phase of the protocol, i.e., this paper will only focus on how the MH system is designed. The approach followed in modeling

the *announcing* and *awarding* phases is based in our previous work in the area of resource allocation in multi-enterprise environments [3, 2], which in turn was based in Sandholm work in the area of automated contracting using extensions of the CNP [8, 9].

The NAg is modeled as an utility-based agent. The definition of an utility-function provides the ability to quantify the monetary compensation that the NAg is willing to tradeoff for the delivering of a particular message. Let $C(t)$ be a function that represents the available cash at instant t. Let $P_i(t)$ be a function that assesses the value of a specific parameter i that influences the level of satisfaction of the NAg's owner (a possible parameter is, for example, the average time of message delivery). Let $U(C(t), P_1(t), ..., P_{Np}(t))$ be the utility function of a particular NAg. This function is specified by the NAg's owner and should be defined according to its preferences.

When the NAg's owner submits a new message to be delivered, it is necessary to define precisely to which SPs the announce should be sent , or, in other terms, it is necessary to define the extent the message's market. This decision depends mainly on two factors: the communication costs of sending the announces and the results of past interactions. The market of a set of messages T is defined as $M(T)$. This set results from the maximization of the expected return of an announcement and is determined as

$$\max_{M(T) \subseteq SPs} \mathcal{E}(\Delta U^{M(T)}(T)),$$

where SPs is the set of involved service providers and $\mathcal{E}(\Delta U^{M(T)}(T))$ is the expected change in utility if the messages in T are announced to the set $M(T)$.

Let $\mathcal{E}(\Delta U^O_{with\_off}(T))$ be the expected change in utility if the set of SPs O make a valid offer and $\Delta U^{M(T)}_{comm}(T)$ be the change in utility caused by the communication costs of sending announces to all the SPs in $M(T)$. Then, for any message set T and for any set of entities $M(T)$, $\mathcal{E}(\Delta U^{M(T)}(T))$ is bounded below by

$$\mathcal{E}^-(\Delta U^{M(T)}(T)) = \min_{O \subseteq M(T), O \neq \emptyset} (\mathcal{E}(\Delta U^O_{with\_off}(T)) + \Delta U^{M(T)}_{comm}(T)),$$

and above by

$$\mathcal{E}^+(\Delta U^{M(T)}(T)) = \max_{O \subseteq M(T), O \neq \emptyset} (\mathcal{E}(\Delta U^O_{with\_off}(T)) + \Delta U^{M(T)}_{comm}(T)).$$

The value of $\Delta U^{M(T)}_{comm}(T)$ is easily assessed by the NAg, since that the cost of sending an announce to a SP is previously fixed.

The final value of $\mathcal{E}(\Delta U^{M(T)}(T))$ is determined by

$$\mathcal{E}(\Delta U^{M(T)}(T)) = \mathcal{E}^{+}(\Delta U^{M(T)}(T)) - \alpha \times (\mathcal{E}^{+}(\Delta U^{M(T)}(T)) - \mathcal{E}^{-}(\Delta U^{M(T)}(T))).$$

where α $(0 < \alpha < 1)$ stands for the NAg attitude. A low value for α leads to a optimistic NAg, that always expects high returns from the announces. High α values will cause the NAg to act in more pessimistic fashion. This parameter should be tuned by the NAg's owner according to its own preferred attitude.

The expected change in utility if the set of SPs O make a valid offer is determined by

$$\mathcal{E}(\Delta U^{O}_{with\_off}(T)) = \max_{Winner \in O} \mathcal{E}(\Delta U^{Winner}_{ex}(T)) \tag{1}$$

where $\mathcal{E}(\Delta U^{Winner}_{ex}(T))$ is the expected change in utility if the message is delivered by $Winner$ (i.e., the negative value that corresponds to its cost), which is estimated based on past values stored in the database of the MH system. Equation 1 gives a hint about how the NAg should proceed in the *awarding* phase: just choose the SP that offers the lower cost.

The application of a economic decision making approach to the previous model brought the ability to determine intelligently and dynamically the best SP to deliver a particular electronic mail message. The possibility to define the utility function and the level of optimism of the NAgs gives to this model the necessary flexibility to adapt him to the requirements of almost every user or organization.

References

[1] A.Tanenbaum. *Computer Networks*. Prentice-Hall, Inc., iii edition, 1996.

[2] Alcino Cunha and Orlando Belo. An electronic commerce framework for resource allocation amongmulti-agent entreprises. In *Proceedings of The 10th International FLAIRS Conference (FLAIRS-97)*, pages 362–366, Florida, USA, May 1997.

[3] Alcino Cunha and Orlando Belo. Resource allocation on agent meta-societies. In *Progress in Artificiall Intelligence, 8th Portuguese Conference on Artificial Intelligence (EPIA-97)*, volume 1323 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1997.

[4] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. In A. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 333–356. Morgan Kaufmann, 1988.

[5] J.Muller. Control Architectures for Autonomous and Interactig Agents: A Survey. In L.Cavedon, A.Rao, and W.Wobcke, editors, *Lecture Notes in Artificial Intelligence (1209). Intelligent Agent Systems - Theoretical and Practical Issues, Based on a Workshop Held at PRICAI'96, Cairns, Australia, August 1996*, pages 1–26. Springer-Verlag, 1996.

[6] M.Rose. *The Internet Message, Closing the Book with Electronic Mail.* Prentice-Hall, Inc., 1993.

[7] M.Wooldridge and N.Jennings, editors. *Intelligent Agents, Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages.* Lecture Notes in Artificial Intelligence (890), Springer-Verlag, 1995.

[8] Tuomas W. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, Washington D.C., 1993.

[9] Tuomas W. Sandholm and Victor R. Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *First International Conference on Multiagent Systems (ICMAS - 95)*, pages 328–335, San Francisco, 1995.

[10] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[11] Michael P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 27(3), 1995.

MODEL-BASED VERIFICATION OF REAL-TIME SOFTWARE FOR AUTOMATIC LOGISTIC SYSTEMS

Christian Kreiner[1], Christian Steger[1], Reinhold Weiss[1]

[1]Institute for Technical Informatics, Technical University, A-8010 Graz, {kreiner,steger,rweiss}@iti.tu-graz.ac.at

Abstract: Control computer systems of automatic logistic systems are complex and highly project-specific software systems. Its reactive real-time parts are usually tested and verified in conjunction with the real · environment. Online coupled models provide an execution environment for the control system, in this way allowing earlier, in-house tests and verification. A framework for the generation of logistic environment models and their integration into the control system's lifecycle is described; it is based on a layered model architecture and domain separation. Implications for the model generation process and model verification issues are discussed.

Introduction

Control computers for automatic logistic systems fulfil a variety of different tasks, including basic control of underlying transport facilities, their synchronization (interchange positions), collision prevention and conflict resolution of transport vehicles (stacker cranes, AGVs), global throughput optimization and reaction to equipment failures, implementation of custom warehouse management strategies, batch scheduling and in-time delivery to truck docks. Additional functionality provides user interface, process visualization and interfacing to enterprise computing systems. Most of the controlling functionality has reactive and real-time characteristics, real-time in a "soft" sense: failing to schedule transports and quickly recover from overload situations can cause high follow-on costs (penalties from missing just-in-time delivery windows, production line stalls, etc.).

Software for Automatic Logistic Systems

The control structure of an automatic logistic system is shown in Figure 1: it is embedded into the enterprises flow of goods and data (E,F). Automatic transport facilities (typically controlled by a local programmable logic controller - PLC) are subordinated to the control computer system (interfaces at fig. 1, C).
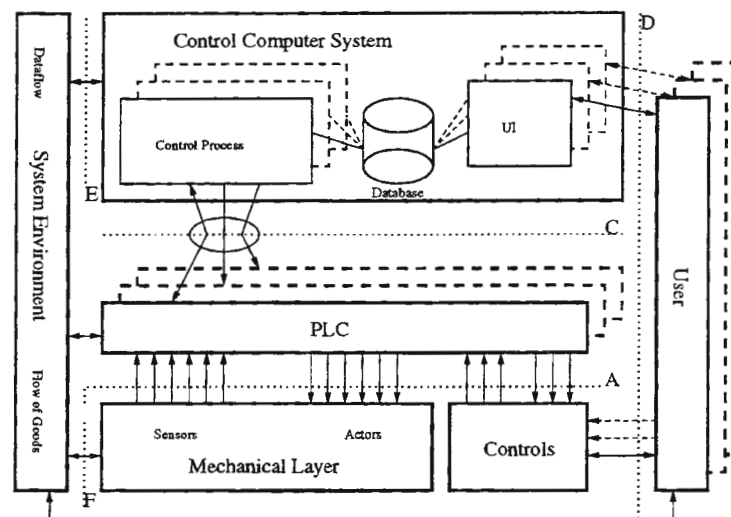

Figure 1: Control Structure of Automatic Logistic Systems

Software for control computer systems is typically developed according to a defined software lifecycle model and tends to be highly project specific (20-30% software reuse [1]). It turns out that test and installation phases of the control computer development raise a large share of the total software project costs (up to 50% [2]), because many tests and verification of reactive components can only be done in conjunction with the real environment. Employing environment simulation in the software lifecycle - providing an executable model of the process to control for the control computer under development - promises a shift from expensive out-house activities towards earlier in-house testing and verification, yielding lower test and installation costs as well as a better software quality. Similar techniques have been used punctually in the logistic area for control system test and optimization [3], control of multi-AGV systems [4], high-level interface design of an AS/RS (automatic storage/retrieval system) [5].

Modeling Process and System Architecture

An execution environment for test and verification of control computer software has to capture the interface behavior towards this system accurately and provide the necessary insight into and control over the corresponding environment activities. Due to the wide range of reactive functionality to check (communication protocol, basic control, synchronization of transport facilities, optimizations, strategies), the model must implement the used communication protocol(s), model the local PLC behavior (transport order scheduling, operation mode control), transport behavior (movements, hand-over of transport units) and storage system (maintaining inventory of goods). In order to allow a broad application of this method, model generation - directly contributing to project software costs - has to be efficient and flexible. Starting from these requirements, a model architecture based on 3 layers (Figure 2) and model domain separation has been developed. Implications for the model generation process and model verification issues are discussed.
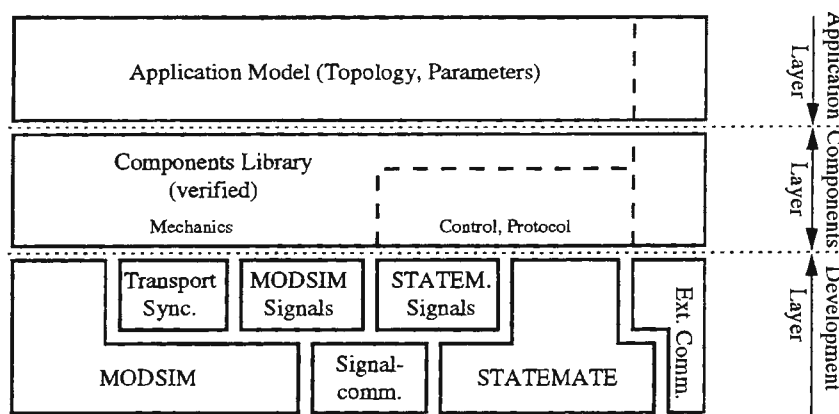

Figure 2: Model Layers

Application Layer

Automatic logistic systems are designed by composing transport lines, racks, cranes, etc. The imitation of this process - building the model from (previously verified) components - provides a familiar environment and procedure for the "logistic domain expert" and assures the application of the plant design in a straightforward manner. The components used are high-level, domain-specific and parameterized objects. In this way model generation on this level - triggered from the control system's design phase - by plugging together component instances should be efficient enough to contribute only marginally to project costs. The executable custom model has to be delivered for the (now in-house) test phase of the control system. It is worth mentioning that there is a potential synergy in using the model in site planning and operation phase too [3].

Components Layer

Activities in the components layer, carried out by a "simulation expert", aim at the population of the above mentioned, reusable components library, which is based on existing transport and component specifications. The components are object-oriented storage/transport models of real-world logistic elements including visualization, animation and parameter management (physical and structural). If applicable, these objects include a model representation of their local controller (PLC) behavior, typically communication protocols (peering the control computer), transport job scheduling and operation mode control. Predefined communication objects are used for outside communication (transport unit hand-over, basic control computer interface) in order to isolate component internals and guarantee correct interaction with other components constructed in this way (Fig. 3, notation from [6]). The library is designed to be "open", i.e. new component objects can be developed and added on demand.

Development Layer

This layer provides basic tool support, verified interface libraries for components, tools and external coupling as well as design rules to apply to component development. In order to exploit their advantages, different tools and methods are used for different modeling domains. This results in a cosimulation environment, where online coupling mechanisms have to be provided, connecting tools [7] as well as coupling the model with the control computer. Due to the nature of our problem it is practical to use asynchronous coupling (no clock synchronization). This allows speeding up one model domain (e.g. mechanical transports) while preserving real-time protocol behavior. Inter-simulator communication is limited to a client-server scheme. Together with a loop-free client-server digraph structure (satisfied in hierarchically controlled logistic systems), internal deadlock-freedom of the model can be achieved [8]. A similar communication mechanism for transport unit hand-over is provided through coupling port objects.
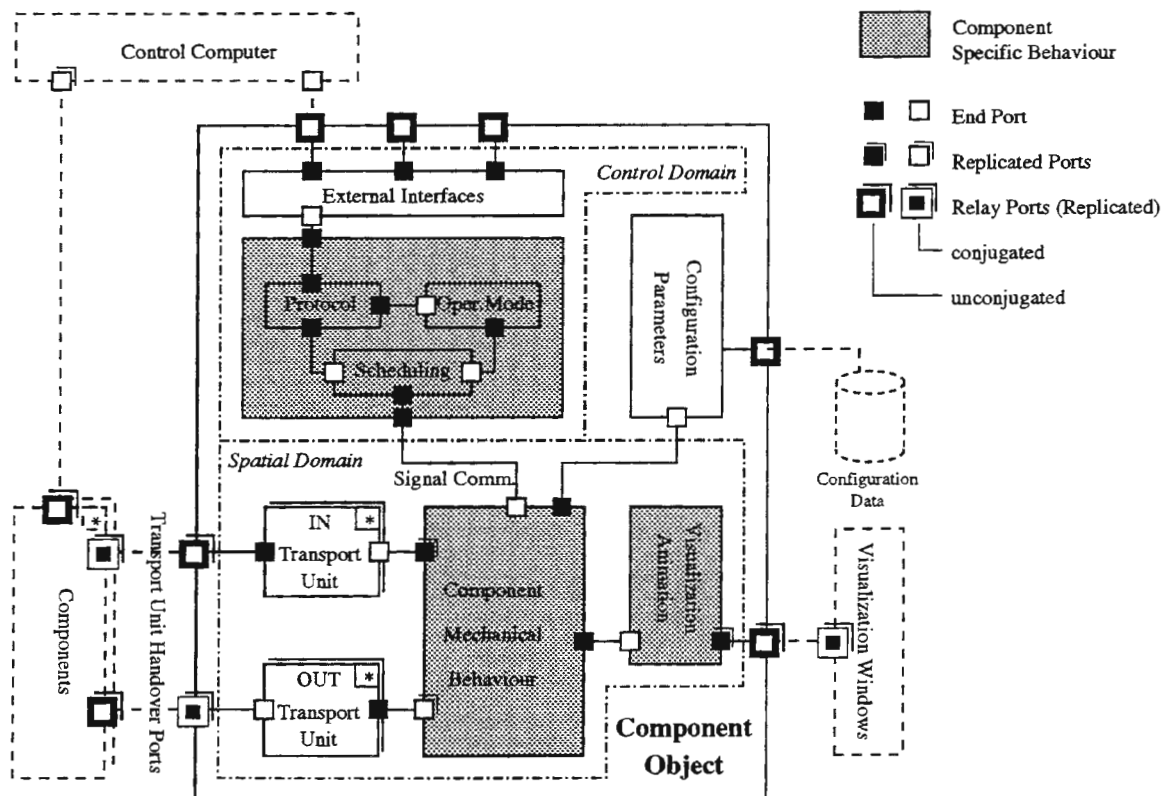

Figure 3: Component Anatomy

Model Domains and Tools

To support efficient modeling, different methods are used for modeling in different domains. Methods, languages and tools are chosen according to their adaptability to the respective modeling domain, i.e. their affinity of language paradigms and constructs to domain objects and dynamics.

Spatial Domain

The spatial domain covers the topographical aspect and transport movements of the site. A process oriented approach following the object-oriented paradigm is used for modeling in this domain. It is desirable that tools should include integrated visualization and animation. This is not only of importance for the application itself but for the dynamic verification of the components as well. Object-orientation supports structuring and reuse (e.g. connected instances of a conveyor segment object). As a system requirement, external communication has to be possible. The discrete event simulation language MODSIM II/III [9] has been chosen for spatial domain modeling.

Control Domain

Control domain functionality like communication protocols and transport job scheduling must have an accurate model representation. Specification is typically given in a free-text form and must be transformed into an executable model, having the valuable effect of improving and clarifying these imperfect specifications. Variants of finite state machines have been used successfully for the description and analysis of reactive systems and communication protocols. Statecharts, a practical and powerful extension, add hierarchical structuring and parallelism to flat finite state machines [10]. An available tool, STATEMATE, complements statecharts with activitycharts (hierarchical data-flow diagrams) and modulecharts as graphical specification languages [11]. Partial and overall verification is supported via static syntax and consistency checking (type checking, matched flow lines, etc.). A powerful simulation tool allows dynamical verification through execution with animating charts, variable tracing, racing condition detection etc.

Model and Component Verification and Validation

Usually the real logistic system and the control computer system are built and developed in parallel; as this applies to the model as well, model validation, i.e. the correctness of the modeled behavior can only be shown after it has been employed in the test and verification phase of the control system. However, model composition of verified, application-specific high-level components and interaction mechanisms leads to internal deadlock freedom and reasonable realistic behavior of the particular model. In the course of first modeling experiments some serious inconsistencies and missing items in an existing, heavily used free-text specification of transport order scheduling and communication protocol for an automatic stacker crane have been detected.

Component verification is supported by the debugging/simulation/visualization facilities of the used tools, allowing glass-box execution of modules and their integration. Component acceptance testing (validation) is done by a coupling experiment with a productive control computer and checking protocol traffic and reactions. Feeding real-life transport history logs into the simulator and checking for adequate transport results and matching final states of the real and the modeled systems validates spatial domain behavior. The validated components are then collected in the components library for reuse. It can serve as a reference communication partner providing and execution environment for in-house testing and verification of future control computer system implementations.

Conclusion

A framework for the efficient generation of executable models of a control computer system's automatic logistic environment has been presented. By providing an execution environment for the control system, it enables early, in-house test and verification procedures. The framework is designed to be easily extensible, to support efficient and flexible model generation, and to be suitable for regular application to the control system's development cycle.

References

[1] G. Fellrieser. *Improving Software Quality using CASE Methods ; ESSI Project No. 21358 (ISUC).* In 8. Austrian ENCRESS Workshop, December 1996.

[2] H. Klumpp. *Informations- und Datenverarbeitung in einem Logistikzentrum für Elektrogeräte.* In Kommissionieren in Industrie und Handel, VDI Berichte ; 1131, S. 63-71, 1994.

[3] J. Prinz and P. Liebe. *Integration von Simulationssoftware mit Leitsystemen.* In 8. ASIM-Fachtagung, Simulation in Produktion und Logistik, pp. 63-71, Berlin, 1998.

[4] Jaap A. Ottjes and Frank P. A. Hogedoorn. *Design And Control Of Multi-Agv Systems. Reuse Of Simulation Software.* In Simulation in Industry, 8. European Simulation Symposium ESS'96, pp. 461-5. SCS, Ghent, Belgium, 1996.

[5] G. Unthank and E. J. Fletcher. *Using a simulation model to test the functionality of a Decision Support System which designs the interface between a logistic centre and other company systems.* In Proceedings of the 1995 EUROSIM Conference, EUROSIM '95, pp. 1083-8. Elsevier, The Netherlands, 1995.

[6] Bran Selic, Garth Gullekson and Paul T. Ward. Real-time object oriented modeling. Wiley, New York, 1994.

[7] Christian Kreiner and Christian Steger. *Modellkopplung STATEMATE-MODSIM.* In STATEMATE User Forum IV. Berner & Mattner, Germany, 1997.

[8] Jeremy M. R. Martin. *The Design and Construction of Deadlock-Free Concurrent Systems.* PhD.-Thesis, School of Sciences in the University of Buckingham, 1996.

[9] CACI Products Company. *MODSIM II. The Language for Object-Oriented Programming.* Reference Manual. CACI Products Company, 3333 North Torrey Pines Court, La Jolla, CA 92037, 1995.

[10] David Harel and Amnon Naamad. *The statemate semantics of state charts.* Technical Report CS95-31, Weizmann Institute of Science, Department Of Mathematics, Rehovot 76100, Israel, October 1995. revised July, 1996.

[11] David Harel and Michal Politi. *Modeling Reactive Systems with Statecharts: The Statemate Approach.* I-Logix, 1996.

HARDWARE–IN–THE–LOOP SIMULATION USING THE PLANAR MANIPULATORS TOOLBOX

Leon Zlajpah

The Josef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia, leon.zlajpah@ijs.si

Abstract

In the paper a MATLAB/SIMULINK Planar Manipulators Toolbox for dynamic simulation of n-degree of freedom planar manipulator is presented. The objective of the toolbox is to serve as an environment for the design of the robotic manipulators control algorithms and the testing of these algorithms on a real system. Beside the toolbox the experimental system includes a planar robot manipulator with four links and low gear actuators. The paper gives a short description of the Planar Manipulators Toolbox and the experimental system. The emphasis is given to the integration of a simulation system and a real robotic manipulator into an integrated environment which enables the simulation of different control schemes and immediate testing on a real system. Finally, some simulations and experimental results are given showing the capabilities of the described system.

Introduction

In general, in the process of controller design different steps have to be performed. First of all, the system has to be modelled. In the next step, the control algorithm is developed. The first results are then obtained by the simulation. If the results are satisfactory, then in the final stage the control algorithms are tested on a real system. For this a real-time code should be generated and implemented on the real system. The integration of all these steps, although essential, is very difficult. Namely, the different steps in the development of the controller require the use of different methods for which different tools are needed. Hence, the results from one step to another have to be transferred often by hand. This bottleneck can be overcome with a system which provides an integrated tool for the control design and testing.

For robotic systems, several simulation systems have been developed for robot analysis and control design: "A Robotic Toolbox" [1] and "A Toolbox for Simulation of Robotics Systems" [3] are implemented in MATLAB, and "Robotica" [2] is based on Mathematica. All these packages support general manipulator structures and as such are rather complex. As such they are not appropriate for the real-time simulation. On the other hand, testing of different control algorithms on a real robotic systems is in general not very user friendly: the algorithms have to be rewritten for the real-time execution and the implementation details have to be considered. Therefore, we have developed a special system for analysis, design and testing of control algorithms for robotic manipulators. The system consists of a simulation software and a real redundant robotic manipulator. The software package for the dynamic simulation of n-degree of freedom planar manipulators is based on MATLAB [4]. This environment has been selected mainly due to its capabilities of solving problems with matrix formulations, easy extensibility, and because of possibility to simulate in real-time, and to automatically generate real-time code.

In the following we give first some background information on kinematic and dynamic model of redundant manipulators and a basic description of the simulation package Planar Manipulators Toolbox. Then, the modelling and building of a robotic manipulator using MATLAB/SIMULINK and the Planar Manipulator Toolbox is described. In the next section, our experimental system, i.e. a four DOF planar manipulator, which has been developed specially for testing of different control algorithms, is presented. Finally, we explain the "manipulator-in-the-loop" simulation where the emphasis is given to the connection of both subsystems, which is done by using the corresponding hardware interface and software drivers. In the last section we give an example of control design an testing by using this environment.

Planar Manipulators Toolbox

The Planar Manipulator Toolbox is a toolbox for simulation of n-R planar manipulators in MATLAB and SIMULINK. It is based on a kinematic and dynamic model of a planar manipulator with revolute joints. Let the configuration of the manipulator be represented by the vector q of n joint positions, and the end-effector position by m-dimensional vector x. Then, the joint and task position are related by the following relation

$$x=p(q) , \qquad \dot{x}=\mathbf{J}(q)\,\dot{q} , \qquad \ddot{x}=\mathbf{J}(q)\,\ddot{q} + \dot{\mathbf{J}}(q)\,q \qquad (1)$$

where p is m-dimensional vector function representing the manipulator forward kinematics and $\mathbf{J}$ is the $m{\times}n$ manipulator Jacobian matrix. The overall dynamic behaviour of the manipulator is described by the following equation

$$\tau = \mathbf{H}(q)\ddot{q} + h(q,\dot{q}) + g(q) \qquad (2)$$

where τ is the n-dimensional vector of control torques, $\mathbf{H}$ is $n{\times}n$ symmetric positive-definite inertia matrix, h is n-dimensional vector of Coriolis and centrifugal forces, $\mathbf{B}$ is $n{\times}n$ matrix of viscous friction coefficients, and g is n-dimensional vector of gravity forces. As the manipulator under study is a planar manipulator with revolute joints, we have been able to simplify significantly the calculation of the model, i.e. the vectors and matrices x, $\mathbf{J}$, $\dot{\mathbf{J}}$, $\mathbf{H}$, h and g, and other model based functions. The detailed derivation of the manipulator models is given in [5] and [4].

As the complexity of the model for this particular type of manipulators increases slower with the number of DOF than in the case of general manipulators, the derived toolbox permits simulation of manipulators with many DOF within reasonable simulation times. The toolbox consists of several M-files for the calculation of the model and other functions for planar manipulators which can not be created using the standard ones. In Table 1 the main MATLAB functions are listed (for a detailed description see [5]). The M-file functions are written in a straightforward manner to gain the understanding of the functions. For a better computational efficiency some functions have been converted to MEX-files using the MATLAB Compiler.

Table 1. Planar Manipulator Toolbox: Main MATLAB functions

Manipulator models		Performance measures	
kinmodel	Kinematic model of n-R manipulator	idxman	Manipulability measure
kinlink	Link positions of n-R manipulator	idxcn	Condition number of $\mathbf{J}$
dynmodel	Dynamic model of n-R manipulator	idxicn	Inverse of condition number of $\mathbf{J}$
modpar	GUI for model parameters	idxgrav	Joint torques norm due to gravity
Graphics		**Inputs**	
linkanim	GUI for manipulator animation	refpoint	Task trajectory: point
linkplot	Graphic representation of manipulator	reflin	Task trajectory: line
Control		refsqr	Task trajectory: square
gotox	Inverse kinematic transformation using kinematic control	refcirc	Task trajectory: circle

As an extension to MATLAB, SIMULINK adds many features for the easier simulation of the dynamic systems, e.q. graphical model building and selection of the integration method and parameters. To exploit additional features, we have developed several blocks and functions needed to create kinematic and dynamic

models and to simulate the motion of n-R planar manipulators in SIMULINK. Figs. 1-5 show some SIMULINK block libraries included in the Planar Manipulator Toolbox.
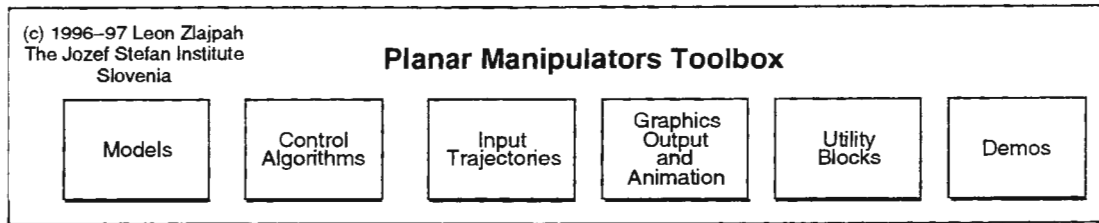

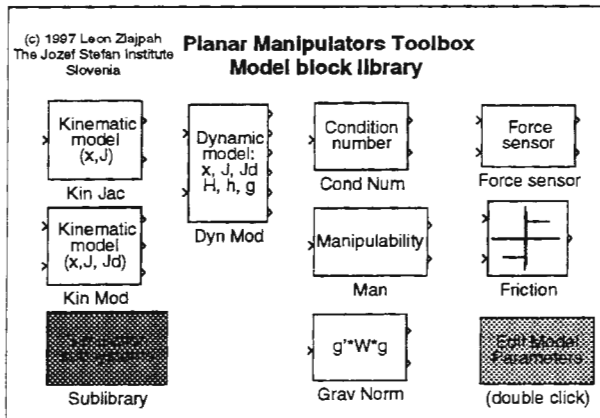Figure 1. Planar Manipulator Toolbox: SIMULINK block libraries
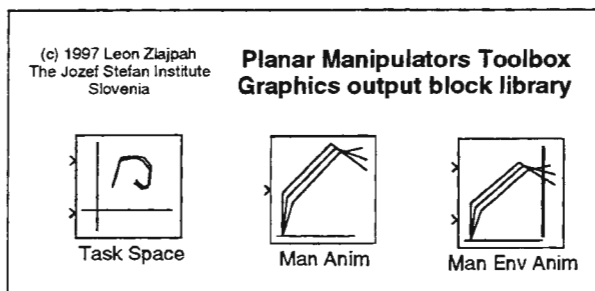

Figure 2. Model block library
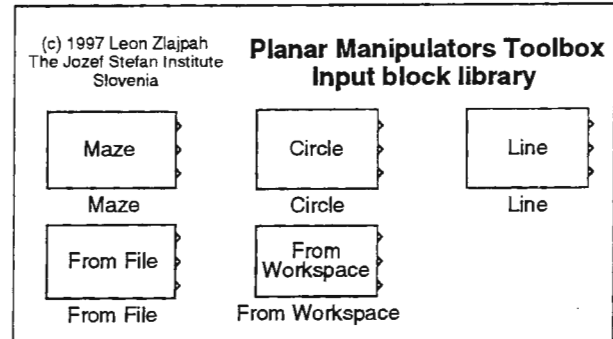

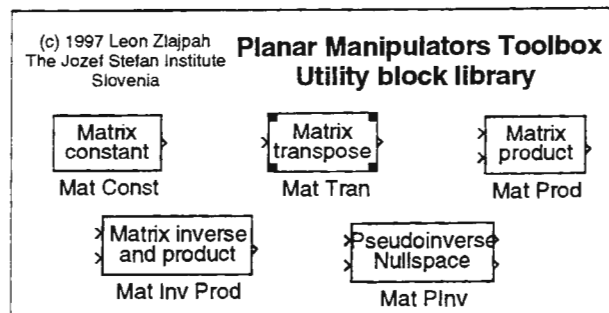Figure 3. Graphics output block library

Figure 4. Input block library

Figure 5. Utility block library

In the Model block library there are blocks needed for modelling of n-R planar manipulators and sensors (like force sensors). The basic blocks are used to calculate the elements of the manipulator model. Combining them with other blocks subsystems representing kinematic or dynamic models are obtained. In the library most common manipulator subsystems are prepared as simulation subsystem blocks. The Input block library includes some predefined task space trajectory generation blocks and the Graphics output block library includes some blocks for the animation of the manipulator motion during the simulation run. As this toolbox has been written for SIMULINK version 1.3 additional utility blocks for matrix computation in SIMULINK have been written and are included in the Utility block library.

All model based blocks which can not be realised with the standard SIMULINK blocks have been realised as S-functions. All S-functions which are necessary for the real-time simulation with the Real-Time Workshop have been rewritten into CMEX versions. Hence, the Planar Manipulators Toolbox enables the real-time simulation of manipulators with real manipulators in the loop, i.e. *Hardware-in-the-loop* simulation.

Model building in SIMULINK

To gain the transparency the system is represented by the block structure with several hierarchical levels. The top level represents the close loop model. The corresponding block diagram is given in Fig. 6.
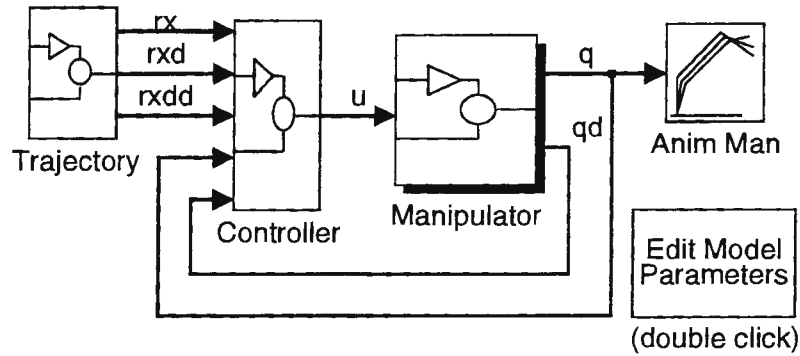
Figure 6. SIMULINK block diagram for a general close loop system

This is a general block scheme. Under the **Manipulator** block the kinematic or dynamic model of the manipulator are modelled. For example, using Eq. (2) the dynamic model of the manipulator can be built straight forward as it is shown in Fig. 7. In this block diagram the block **Model** includes the calculation of model matrices and vectors x, $\mathbf{J}$, $\dot{\mathbf{J}}$, $\mathbf{H}$, h and g. Additionally, the model includes the **Friction** block representing Coloumbic friction.
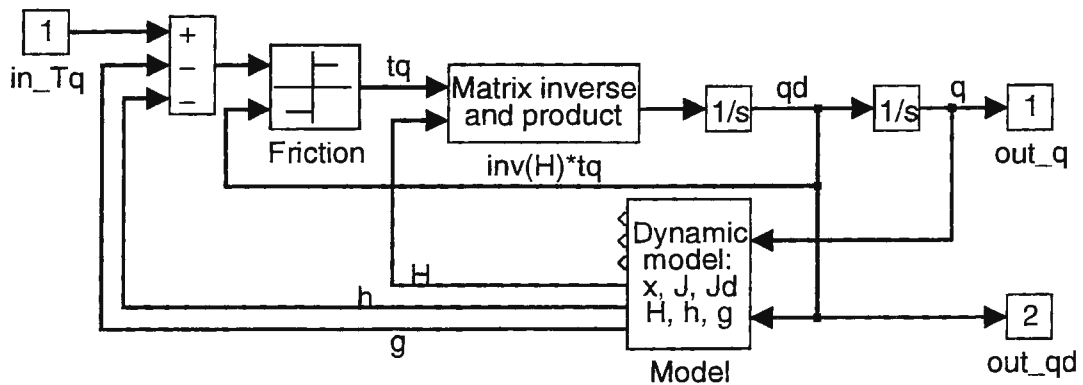
Figure 7. SIMULINK blocks for a dynamic model of a manipulator

The next main subsystem is the **Controller** block. The contents of the **Controller** block depends on the particular control algorithm the user wants to simulate. In most cases the controller for the manipulator includes part of the manipulator model. As an example, Fig. 8 shows a controller for position tracking control of an redundant manipulator in the task space

$$\tau = \mathbf{H}\mathbf{J}^{+}(\ddot{x}_d + \mathbf{K}_D(\dot{x}_d - \dot{x}) + \mathbf{K}_P(x_d - x) - \dot{\mathbf{J}}\dot{q}) + h + g$$

$$\mathbf{J}^{+} = \mathbf{H}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^T)^{-1}$$

(3)

The controller includes the calculation of the complete dynamic manipulator model. As the reference trajectory input a block from the Input block library can be used or the user can use standard SIMULINK blocks to generate the input trajectories.
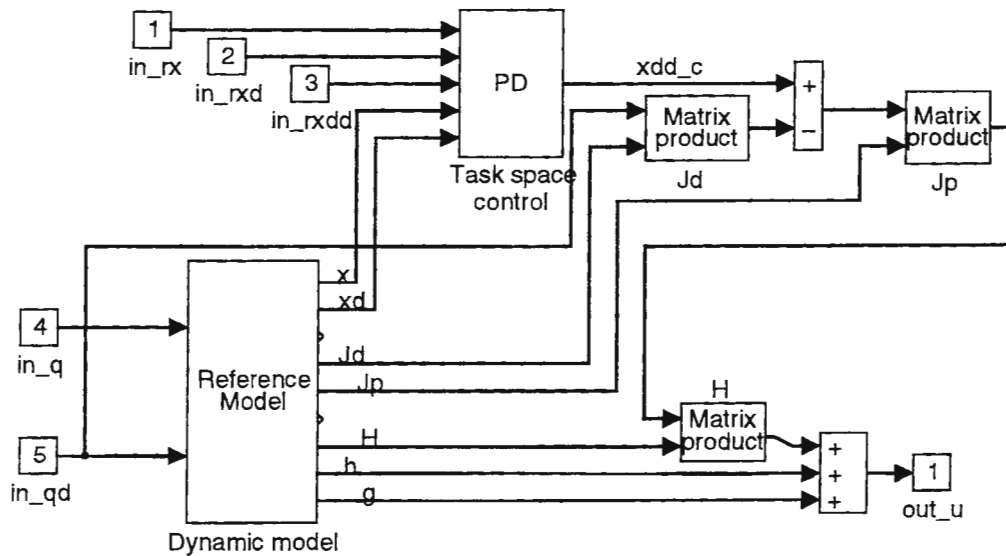
Figure 8. SIMULINK blocks for the control algorithm (3)

Manipulator in the loop

Simulation systems as previously described represent an essential tool for development of complex systems like control algorithms for redundant manipulators or path planning. Unfortunately, for the final tests the developed algorithms should be implemented on a real system. This step is usually done separately using another system. Hence, rewriting the algorithms into another language and/or for a different platform may be needed. For the implementation on the real system it is necessary also to consider different implementation details like interfaces, sampling frequencies, etc. As a consequence, the development time can be significantly prolonged, especially if the design process has to be repeated. A promising concept to overcome these problems and to integrate the design phase and the testing on a real system in one environment is the hardware-in-the-loop simulation. The main idea of this concept is to substitute one part of the simulation model by a real system during testing. Hence, the testing of the algorithm can be done instantly without the need to rewrite or change them in any way. Prerequisites for the "hardware-in-the-loop" simulation is the ability of real-time simulation and additional hardware components which are needed for linking the real system and the simulation system together.

In our case hardware-in-the-loop means manipulator-in-the-loop, i.e. the model of the manipulator is replaced by a real manipulator. The interface between the simulation systems and the manipulator consists of I/O boards with the corresponding software drivers at the control computer and servo drives at he manipulator side. Such system can be modelled as shown in Fig. 9. Depending on the particular servo drives used in the real systems and the available control modes of these drives the control signals can be either torque, speed or position command inputs. Actually, the flexibility of the servo drives determines how the real manipulator can be included into the simulation system. For example, if a controller includes compensators for manipulator dynamics, then the control signals should represent joint torques and servo drives must be able to work in torque control mode (see Fig. 10). On the other hand, when control signals correspond to joint velocities servo drives must be in speed control mode. In this case, the simulation manipulator model has to include also the internal speed controller as shown in Fig. 11. The other part of the interface is dedicated to the measurement of the state of the manipulator. In most cases the state is obtained by measuring joint positions (and velocities) as it is shown in Fig. 9.

Note also that when using the hardware-in-the-loop additional gains are included in the system. They aim is to obtain correct signal values for/from the interfaces and to make available in the simulation model the signals which can be measured on the real systems used. In this way the transparency of the system is very good. We want to point out that using the block scheme as shown in Fig. 9 the user can select whether he will use in the

simulation run the model of the manipulator or the real system by simply setting Access_Hardware flag which determines if the signals pass through the hardware or not. For example, if the real-time code is generated with Access_Hardware flag set, the signals go from Output interface block via the real manipulator to the Input interface block, i.e. the Manipulator block is not used. However, for higher sampling rates during the real-time hardware-in-the-loop simulation it is recommended to exclude the blocks between the Input interface and Output interface blocks because they are not active during simulation.
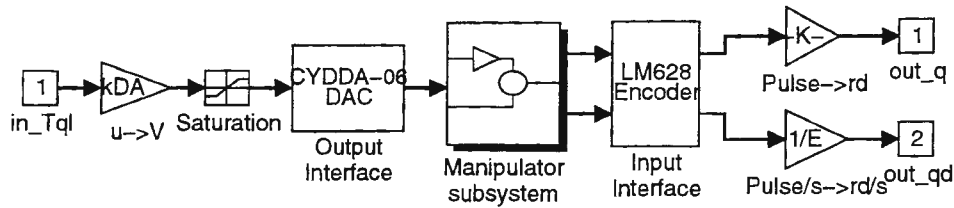
Figure 9. Block diagram of the part of the "manipulator-in-the-loop" system with hardware interfaces
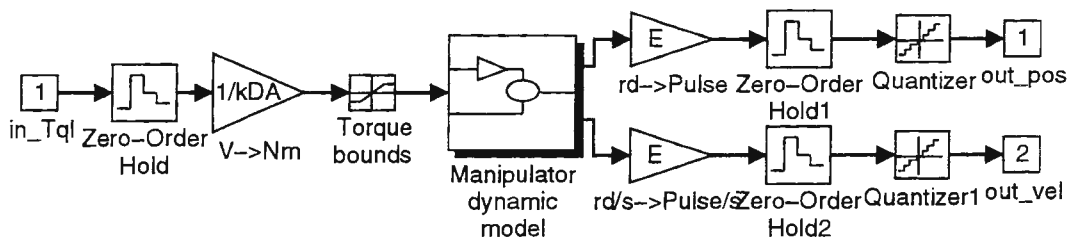
Figure 10. Manipulator subsystem for "Manipulator-in-the-loop" with torque control mode
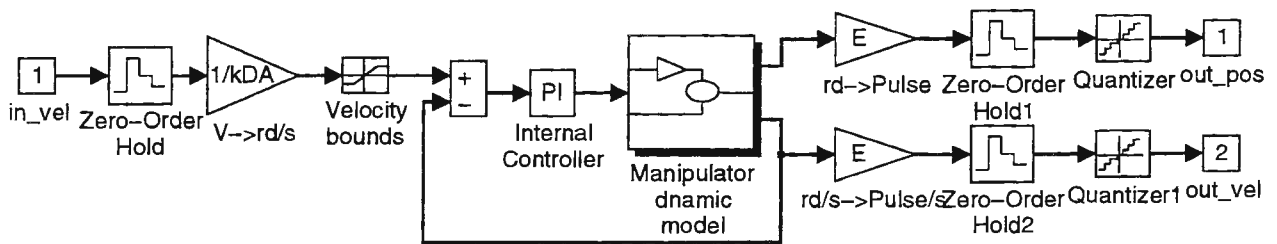
Figure 11. Manipulator subsystem for "Manipulator-in-the-loop" with velocity control mode

Experimental system

The main part of our experimental system is a laboratory manipulator developed specially for testing of different control algorithms for robotic manipulators. The manipulator has four revolute DOF acting in a plane (see Fig. 12). The link lengths are approximately 0.25m each. The configuration of the manipulator enables also the testing of algorithms for redundant systems.

As the task space is two-dimensional (x-y plane) the manipulator has two redundant DOF. We have selected the configuration with two redundant DOF because in our opinion almost all characteristics of redundant systems can be tested on a system with a least two redundant DOF. On the other hand, the planar structure with revolute joints was selected because it enables real-time simulation of advanced control algorithms considering dynamic properties of the manipulator. To be able to control the influence of the gravity forces, the slope of the plane where the manipulator is moving can be adjusted to any angle between horizontal and vertical position. Additionally, the construction of the manipulator poses no limitation for the joint positions. The manipulator is driven by AC motors and gears with low gear ratios (12 and 6). The servo
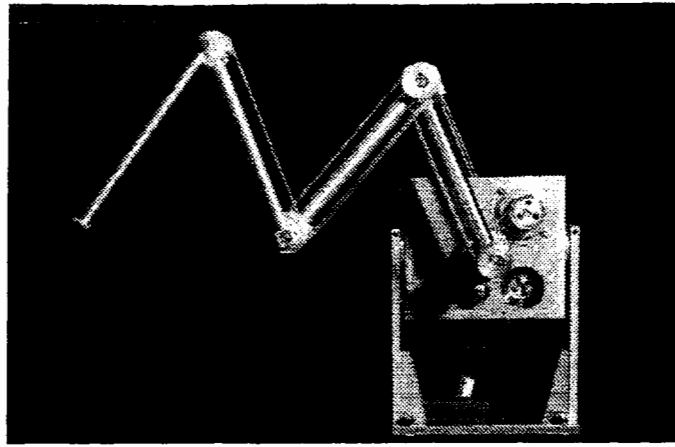
Figure 12. Experimental 4-R planar manipulator

drives have different control modes: the torque mode (input is the desired torque of the motor) and the speed mode (input is the desired speed of the motor). The motors are equipped with incremental encoders with 2500 pulses/rev.

For the control we use in a first place a PC computer (Pentium/166MHz). For linking the manipulator with the computer we have Lab-PC+ board (National Instruments), CYDDA06 board (CyberResearch) and for interfacing the incremental encoders our own PC-board based on LM628 (National Semiconductor). For all these board software drivers for SIMULINK have been written. With this equipment we can achieve up to 25Hz sampling frequency within SIMULINK (in the interpreter mode). On the other hand, when from the SIMULINK block diagrams a C code is generated and then compiled for the real-time application the sampling frequencies more than 1kHz can be achieved depending on the complexity of the control algorithms.

Simulation and experimental results

To illustrate the robot control design and testing we have selected the following example: tracking of the circle in the task space with a four DOF planar manipulator. To control the task position the control algorithm as given by Eq. (3) and shown in Fig. 8 has been used. The reference trajectory has been selected as a circle with radius 0.4m and period time 5s.

In the first step we have made a simulation using the complete model of the manipulator including also the Coulombic and viscous friction (see Fig. 7). The practical usefulness of the simulation results depends mainly on the accuracy of the model parameters (assuming that the structure of the manipulator model is known). In our case all model parameters except the friction have been determined using the construction data. The friction parameters have been identified by measurements using the Planar Manipulators Toolbox.

In the next step the control algorithm has been tested on our experimental manipulator. As the interface D/A converters on CYDDA06 board and encoder interfaces on LM628 board have been used (see Fig. 9). The sampling frequency has been 500Hz. The simulation results and the experimental results are shown in Fig. 13 and 14. Fig. 13 shows the task positions and configurations of the manipulator during the motion for every 0.5s. In Fig. 14 the joint velocities versus time are presented. From both figures we can see that the simulation results and the results on a real manipulator are almost equal. The little difference is due to some uncertainties in the parameters of the friction model. Note that the controller does not controls the internal motion of the manipulator and that the internal motion depends only on the friction. Note also that after the SIMULINK block structure has been defined, all further steps for the simulation using the "manipulator-in-the-loop" are practically equal to those of the "pure" simulation as all the implementation details are hidden for the user.
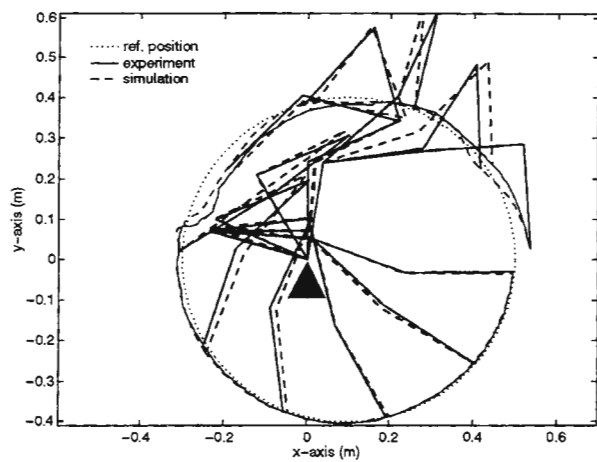
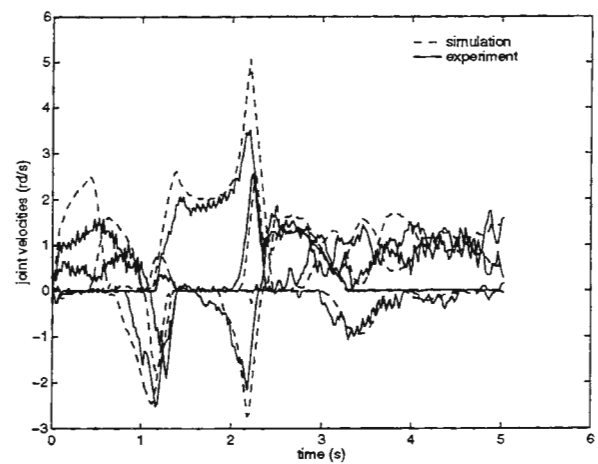Figure 13. Task positions and manipulator configurations

Figure 14. Joint velocities

Conclusion

In the paper we present an integrated environment for controller design for a robot manipulator. The system consists of two subsystems. The first one is a simulation package Planar Manipulators Toolbox for dynamic simulation of n-degree of freedom planar manipulator based on MATLAB/SIMULINK and the other subsystem is a experimental planar robot manipulator with four links, revolute joints and low gear actuators. The paper gives a short description of the Planar Manipulators Toolbox and the experimental system. The emphasis in the paper is given to the connection of both subsystems into an integrated environment for the design and the testing. We show that the "manipulator-in-the-loop" simulation (where the model of the manipulator in the simulation scheme is substituted with the real manipulator by using the corresponding hardware interface and software drivers) is a promising concept in the process of controller design for robotic manipulators as it enables the simulation of different control schemes and immediate testing on a real system. Additionally, the derived system has proved to be useful and effective tool because the user can design and test different control schemes without considering the implementation details. The user can switch between the "model" and "real manipulator" simply by setting one flag. In both cases the input and output signals and the rest of the system are identical.

Summarising, the Planar Manipulators Toolbox is very useful and effective tool for many purposes: kinematic simulation, dynamic simulation, analysis and synthesis of control systems, trajectory generation, etc. It is very easy to extend and to adapt the simulation package to different requirements. Thus, it should be of interest to the researchers involved in the development of advanced robot control systems.

References

[1] P. I. Corke. A Robotics Toolbox for MATLAB. IEEE Robotics & Automation Magazine, 3(1), pp24-32, 1996.

[2] J.F. Nethery and M.W. Spong. Robotica: a Mathematica package for robot analysis. IEEE Robotics & Automation Magazine, 1(1), 1994.

[3] D. Surdilovic, E. Lizama, and J. Kirchhof. A Toolbox for Simulation of Robotic Systems. In F. Breitenecker and I. Husinsky, editors, EUROSIM '95 Simulation Congress, pp693-698, Vienna, 1995. Elsevier Science.

[4] L. lajpah. Dynamic Simulation of n-R Planar Manipulators. n F. Breitenecker and I. Husinsky, editors, EUROSIM '95 Simulation Congress, pp699'704, Vienna, 1995. Elsevier Science.

[5] L. lajpah. Planar Manipulators Toolbox: User's Guide. Technical Report DP-7791, Jo ef Stefan Institute, 1997. URL: http://www2.ijs.si/~leon/planman.html.

MODELING AND SIMULATION OF LINEAR QUADRATIC REGULATORS FOR MIMO AIR-CONDITIONING SYSTEMS USING A SIMULINK SOFTWARE PACKAGE

B. Šumilin1, V. Zanchi2

Faculty of Electrical Engineering, Machinery and Naval Architecture Split, R. Boskovica bb, 21000 Split, Croatia
Boris.Sumilin@fesb.hr
Faculty of Electrical Engineering, Machinery and Naval Architecture Split, R. Boskovica bb, 21000 Split, Croatia
Vlasta.Zanchi@fesb.hr

INTRODUCTION

The paper is dealing with design of Linear Quadratic Regulator (LQR) for optimal regulation of temperature in a complex building consisting of few subsystems. After syntheses of LQR regulator and mathematical modeling of heat regulation, the numerical validity checking of proposed project has been carried out. All data concerning the simulated system are based on the analyses of existing buildings and surrounding meteorological conditions. The analyses have been performed on simulation model using a SIMULINK Dynamic System Simulation Software.
Keywords: Multi Input - Multi Output system (MIMO), LQR regulation, Quadratic Performance Criteria (QPC), Simulation.

LINEAR QUADRATIC REGULATION (LQR)

The design of linear optimal regulation is based on description of MIMO systems given by differential state equation [L.5],

$$dx/dt = A(t)x + B(t)u, \; x(0) = X_0 \tag{1.}$$
$$y = C(t)x + D(t)u$$

Essential task of optimal regulation is definition of feedback matrix K, so that chosen performance criterion has minimal value. Performance criterion, accepted at designing of regulation system, depends upon characteristics of the system to be optimized, its economy, reliability, applicability, usefulness, etc. One way to determine optimal feedback matrix K is by selection of Quadratic Performance Criteria in a form of [L.5],

$$J = \frac{1}{2}x^T(t)S(T)x(t) + \frac{1}{2}\int_0^T \left(x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) \right) dt \tag{2.}$$

where (0,T) is the observed time interval and S(t), Q(t), R(t) are symmetrical weight matrixes, selected as requested by system performance. These matrixes should be chosen in a way that they meet the following requests $x^T(t)S(T)x(t) \geq 0$ and $x^T(t)Q(t)x(t) \geq 0$, for all $x(t)$ and $u^T(t)R(t)u(t) > 0$, for all $u(t)$.

The selection of matrixes S(T), Q and R is based on a compromise and on a principle that variables of higher weights aim at decrement in a closed loop. For example, if elements of matrix Q are chosen much smaller than those from S(T) and R, the corresponding components of x(t) shall be, by means of optimal regulation, kept smaller. On the other hand, if R is chosen high, u(t) shall be kept smaller, so that less energy shall be consumed during regulation. Commonly, system poles of closed loop are "faster" (require more energy during regulation) for smaller R. At last, if the elements of final state of weight matrix S(T) are chosen higher, then final states of x(t) in a closed loop shall be smaller.

If the system and matrixes A, B, Q and R are time-invariant, the Riccati equation has constant solution Π_∞. Then, $d\Pi/dt = 0$, so that it transforms to Riccati algebraic equation (ARE) resulting to constant Kalman steady state gain.

In such case the LQR design equations are [L.5],

$$0 = Q + A^T\Pi + \Pi A - \Pi BR^{-1}B^T\Pi \qquad \text{- ARE} \tag{3.}$$
$$K = R^{-1}B^T\Pi \qquad \text{- Kalman gain}$$

The different programming packages, as MATLAB, MATRIX and SIMULINK, contain routines for solving these equations, using system of (A, B, C) constant matrixes and performance criteria weight matrixes (Q,R).

Temperature regulation of a closed space.

Air-conditioning and air-handling systems supply or remove heat and humidity in air-treatment areas. Therefore, air temperature and humidity are the basic computation factors of each air-conditioning systems. System load should be also computed. The selection and sizing of equipment are such that it optimally performs the required functions, in accordance with computed load, main basic factors and initial conditions. Basic aim of the investigation was mathematical modeling of air-conditioning system and air handling process, as well as control and simulation of such modeled process.

The functional temperature dependence inside room, as the basic computing element, is given by differential equation (4.). Physical scheme is shown on Fig.1, [L.9],

$$\dot{T}_R = - \frac{1}{R_{eq}Mc} T_R + \frac{1}{R_{eq}Mc} T_{OA} + \frac{1}{Mc} Q \qquad (4.)$$

where, Q - heat quantity coming from heat exchanger, R_{eq} - room equivalent thermal resistance, M - total air mass inside room, c - air thermal constant ($1005.4 \ Jkg^{-1}K^{-1}$), T_R - temperature inside room and T_{OA} - outside air temperature.
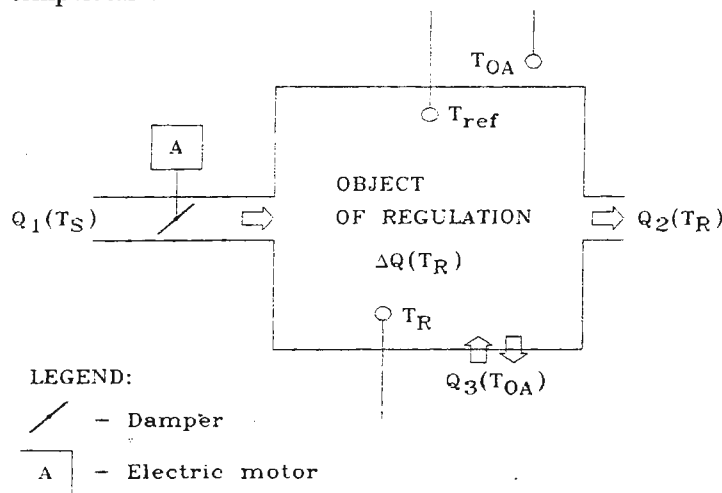
Fig. 1. The object of regulation

The point of our investigation was temperature only, while air humidity at this moment was ignored. The functionality of the system was not imperiled by such approach.

All parameters of the object of regulation (room) are computed according to the above quoted equations, [L.9].

Equivalent room thermal resistance is $R_{eq} = \dfrac{R_z R_p}{R_z + R_p}$, where total room wall thermal resistance is $R_z = \dfrac{l_z}{k_z A_z}$,

l_z - wall thickness, A_z - surface of all walls limiting the room, k_z - wall constant, depending of the kind of used insulation (in our case glass insulation k = 0.038).

The total window thermal resistance is $R_p = \dfrac{l_p}{k_p A_p}$, where l_p - window thickness, A_p - all window surface,

k_z - window constant depending of material used and window type (in our case k = 0.78).

The modeling of the object of regulation (rooms) is performed by program language SIMULINK and by application of differential equation (1.). The work proceeds with design and implementation of temperature regulator in a previously modeled object. Two models of regulations are proposed: LQR regulation (Fig. 2.a) and thermostat regulation (Fig. 2.b). Room temperature in a winter period is controlled by LQ regulator and thermostat. System responses are given on figures 3.a) and 3.b). At the same time, energy consumption is monitored, both with LQ regulation (Fig. 4.a) and thermostat regulation (Fig. 4.b), by means of formula
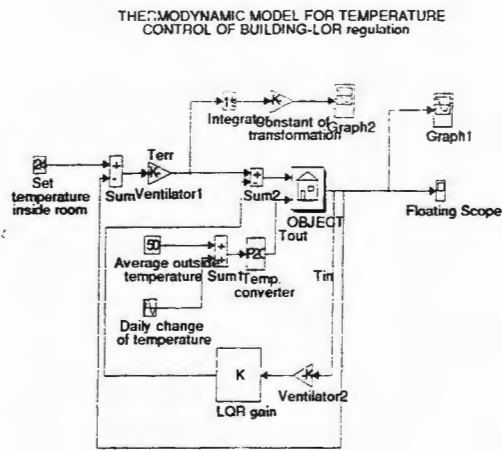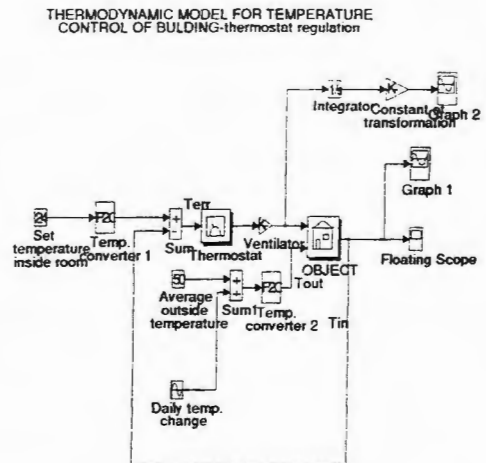
$$Q = mc \int dT.$$

Fig. 2.a

Fig. 2.b

Fig. 3.a) System response -LQR regulation
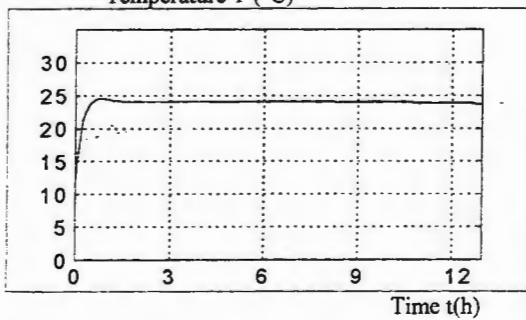(winter period - object 1)

Temperature T (^{o}C)

Time t(h)

Fig. 3.b) System response - Regulation with thermostat
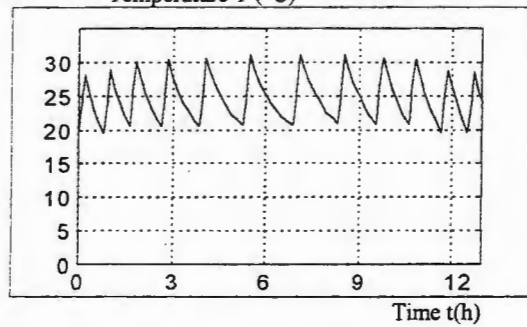(winter period - object 1)

Temperature T (^{o}C)

Time t(h)
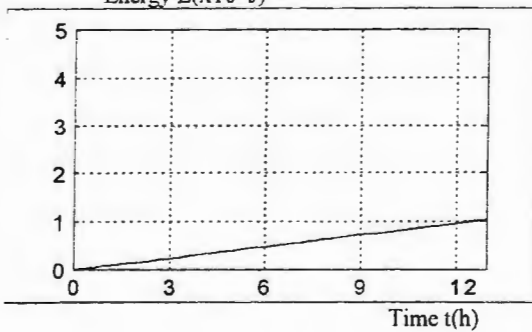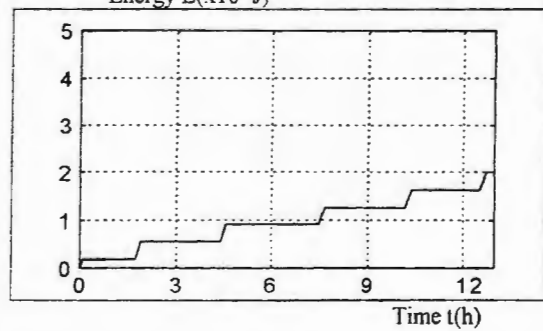
Fig. 4a) Energy consumption - LQR regulation

Energy E($\times 10^5$J)

Time t(h)

Fig. 4b) Energy consumption - Regulation with thermostat
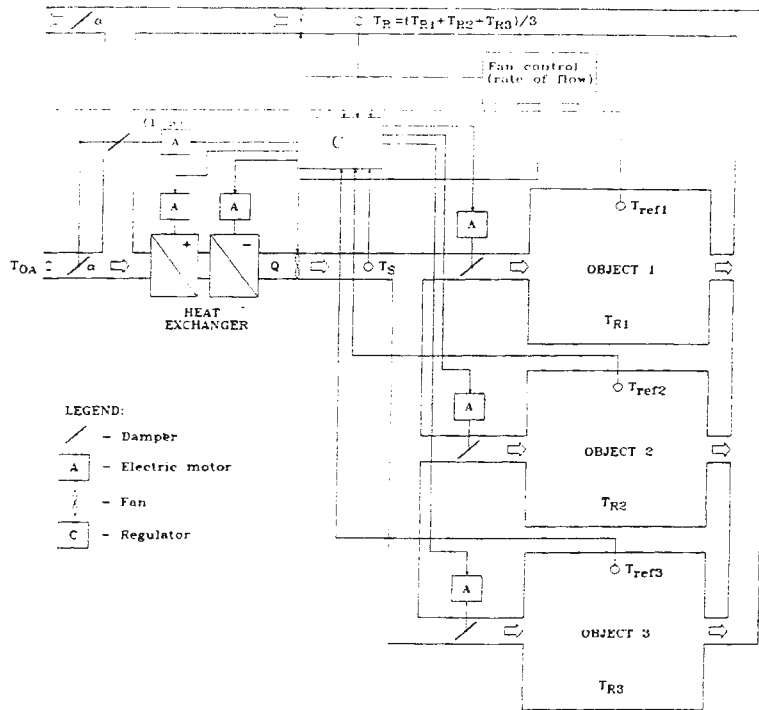

Energy E($\times 10^5$J)

Time t(h)

Fig. 5. Compound LQ temperature regulation system with utilization of return air

The system shown on Fig. 5. controls temperature in a complex building, by LQ regulator extended with control of used return air. Mixture of fresh outside air and drawn out room air is introduced into system, depending of their temperatures.

The dampers control the quantity of return air coming back into rooms. They are placed inside ventilation pipes to control air flow. The damper turning regulates air mixture temperature T_M coming to heat exchanger. The elementary damper role is energy saving, as main energy part is consumed by air heating or cooling. By gradual damper rotation the air temperature T_M is approximating the set value of temperature T_{ref}. In this way the consumed energy for heating or cooling decreases.

The obtained air mixture temperature is [L.9],

$$T_M = \alpha\, T_{OA} - (1 - \alpha)\, T_R, \qquad\qquad (5.)$$

where α is the part of outside air in the mixture, ranging from 0.2 to 1.

Air-conditioning system for temperature regulation in a complex building object is MIMO system. According to the physical model on Fig. 5. it contains mixing room, heat exchangers, air-conditioning rooms and fan air distribution system. As per initial model of the object of regulation, the output variables are temperatures T_{R1}, T_{R2} and T_{R3} of the air which is drawn out from the conditioning rooms. The main input variables are outside air temperature T_{OA} and set values of temperatures inside rooms - T_{ref1}, T_{ref2} and T_{ref3}. The auxiliary input variable is outside and return air mixture temperature T_M. The controlled variable is the quantity of heat Q, coming into system from heat exchanger. In this case the simulation and testing are performed on simulating model shown on Fig. 6. The results are given on Figs. 6-1, 6-2 and 6-3.
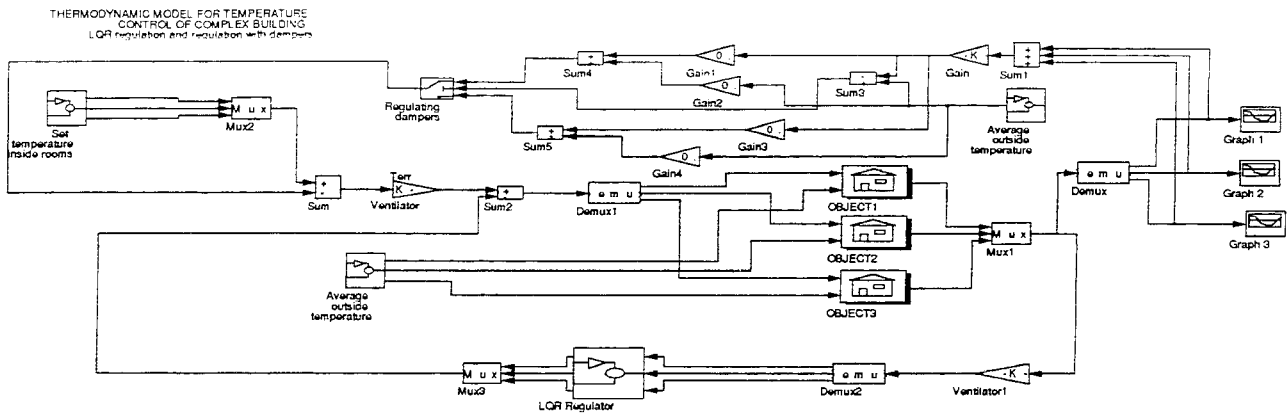
Fig. 6) Simulation model

THERMODYNAMIC MODEL FOR TEMPERATURE
CONTROL OF COMPLEX BUILDING
LQR regulation and regulation with dampers

Fig. 6-1) System response-LQR regulation (winter-object 1)
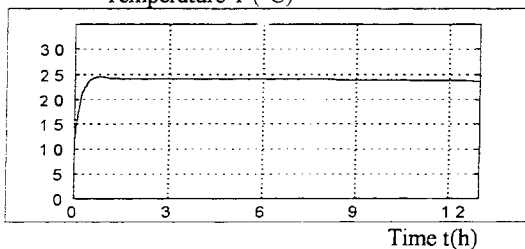

Temperature T (°C)

Time t(h)

Fig. 6-2) System response-LQR regulation (winter-object 2)
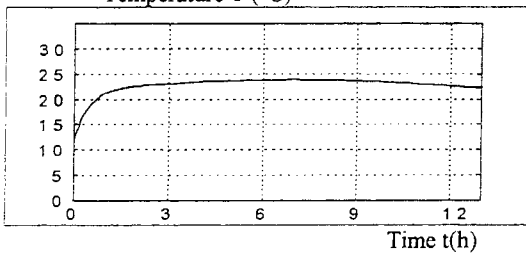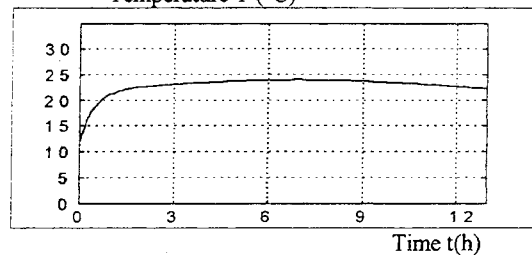

Temperature T (°C)

Time t(h)

Fig. 6-3) System response-LQR regulation (winter-object 3)

Temperature T (°C)

Time t(h)

CONCLUSION

The main purpose of this paper was introduction to mathematical modeling and designing of temperature control in a complex building. For such model the structure of LQR optimal regulator has been accepted, what is novelty in relation to so far used temperature regulators.

After syntheses and implementation of LQR regulator for heat system, the numerical validity of proposed project has been carried out. Simulation system data are based on the analyses of the existing building objects and meteorological conditions characteristic for seaside part of Croatia in the winter period. The selection of weight matrix values for performance criteria is based on simulation for proposed values of Q and R matrixes. The analyses has been accomplished on previously described simulation models (with SIMULINK program package). The results justify the application of selected LQR regulator. The comparison with commercial systems in use (regulation with thermostat) shows the significant energy saving and almost entire absence of temperature variation in the closed spaces, if systems are controlled and monitored by modern methods of linear quadratic regulation. The analyses on simulation model for temperature regulation of complex building by LQ regulator extended with control of used return air has been also accomplished. The results show smaller temperature variations in building, if return air is used along with LQ regulator. Although temperature variations are present and with LQ regulation in case of bigger objects, they are negligible ($\pm 1^\circ$C), especially compared with classical methods (thermostat) of temperature regulation.

Literature

[1.] L. Fornera, A.S. Glass, P. Gruber, and J. Todtli, Qualitative Fault Detection Based on Logical Programming Applied to a VAV Air Handling Unit, Second IFAC Workshop on Computer Software Structures Integrating AI/KBS Systens in Process Control, Lund, Aug. 10-12, 1994.

[2.] S. Skogestad and I. Postlethwaite, Multivariable Feedback Control, J. Wiley and Sons Ltd., 1996.

[3.] V. Zanchi, Automatika I, FESB, Split, 1989.

[4.] R.C. Dorf, R.H. Bishop, Modern Control Systems, Addison Wesley Publishing Company, 1995.

[5.] D.M. Wiberg, Theory and problems of state space and linear systems, McGraw-Hill Book Company, 1971.

[6.] V. Zanchi, Optimizacija, FESB, Split, 1983.

[7.] A.S. Glass, P. Gruber, M. Roos, J. Todtli, Qualitative model-based fault detection in air-handling units, IEEE Contr. Syst., 1995.

[8.] A.E. Bryson, Y.C. Ho, Applied Optimal Control, J. Wiley and Sons, 1975.

[9.] O. Fabris, Osnove in'enjerske termodinamike, Svjetlost, Sarajevo, 1989.

[10.] V. Zanchi, Simulacija, FESB, Split, 1989.

[11.] B.D.O. Anderson, J.B. Moore, Optimal Control: Linear Quadratic Methods, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[12.] J.C. Doyle, B. Francis and A. Tannenbaum, Feedback Control Theory, Macmillan Publishing Company, 1992.

[13.] C.E. Rohrs, J.L. Melsa and D.G. Schultz, Linear Control Systems, McGraw-Hill International Editions, 1993.

Simulink Simulation of Hodgkin-Huxley Equation

Mojmil Cecić, Vladan Papić, Vlasta Zanchi

Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
R. Boskovica bb, 21000 Split, Croatia, mcecic@fesb.hr

Abstract

The simulation of dynamic process in biologic systems by analogue computer is well known and widely used problematic since 1960. The analogue simulation of Hodgkin-Huxley equation [1], describing the electric current through the surface of membrane, is given in [2]. Fast development of personal computers causes a large number of simulation languages such as, widely use, Vissim and Simulink. This paper deals with Simulink simulation of Hodgkin-Huxley equation. The Simulink program (Fig. 1) and the results of simulation will be included.

Hodgkin-Huxley equation

In the concluding paper of the series [1] Hodgkin and Huxley developed a mathematical model describing the electrical behaviour of the membrane. Based on experimental results the total current through the membrane is:

$$I = C_M \frac{dV}{dt} + \overline{g}_K n^4 (V - V_K) + \overline{g}_{Na} m^3 h (V - V_{Na}) + \overline{g}_L (V - V_L) \qquad (1)$$

where $C_M \dfrac{dV}{dt}$ - capacity current through the membrane

V - displacement of the membrane potential

V_{Na}, V_K, V_L - particular equilibrium potential

$\overline{g}_K, \overline{g}_{Na}, \overline{g}_L$ - ionic conductances

n, m, h - quantities which express the variation of conductance defined by the following equations:

$$\frac{dm}{dt} = \alpha_m (1 - m) - \beta_m m$$

$$\frac{dh}{dt} = \alpha_h (1 - h) - \beta_h h$$

$$\frac{dn}{dt} = \alpha_n (1 - n) - \beta_n n$$

α's and β's are experimental determines values of ionic transfer rate from outside to inside and reverse [1].

Simulation

Personal computer simulation, based on known analogue model described in [2], is done by use Simulink. Simulink model, created from equation (1), is given on Fig. 1. The only major reservation which must be made about equation (1) is that the membrane is in equilibrium state, so total current through the membrane is equal zero (I=0).
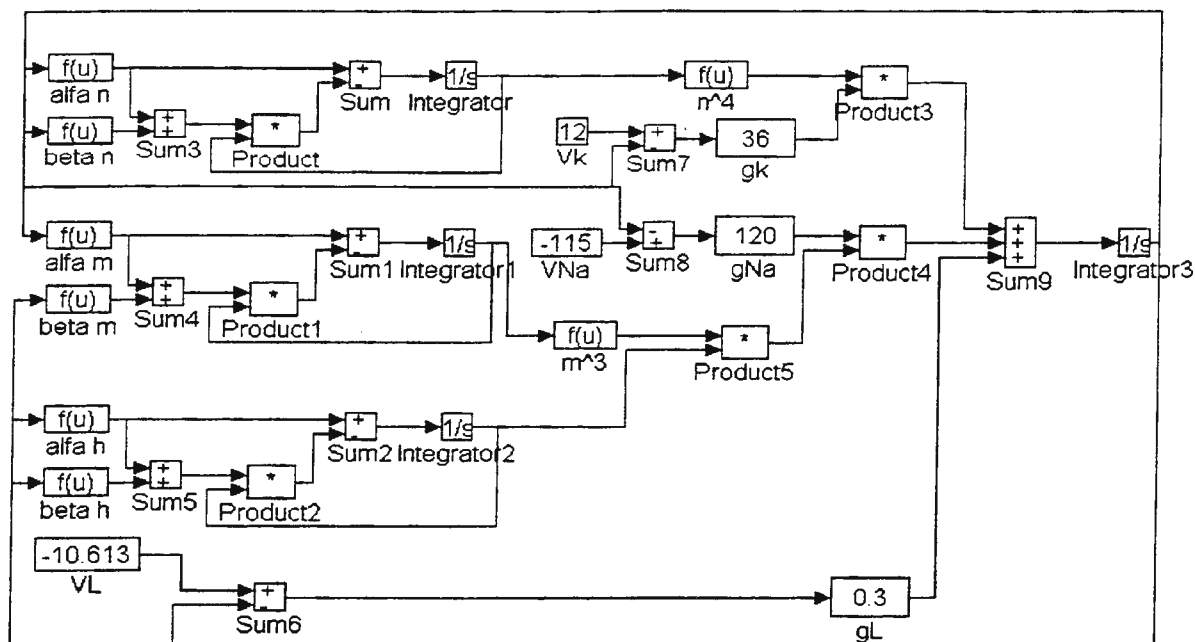
Fig. 1. Simulink model of Hodgkin-Huxley equation

The results of simulation for several different values of potential V is given on Fig. 2:
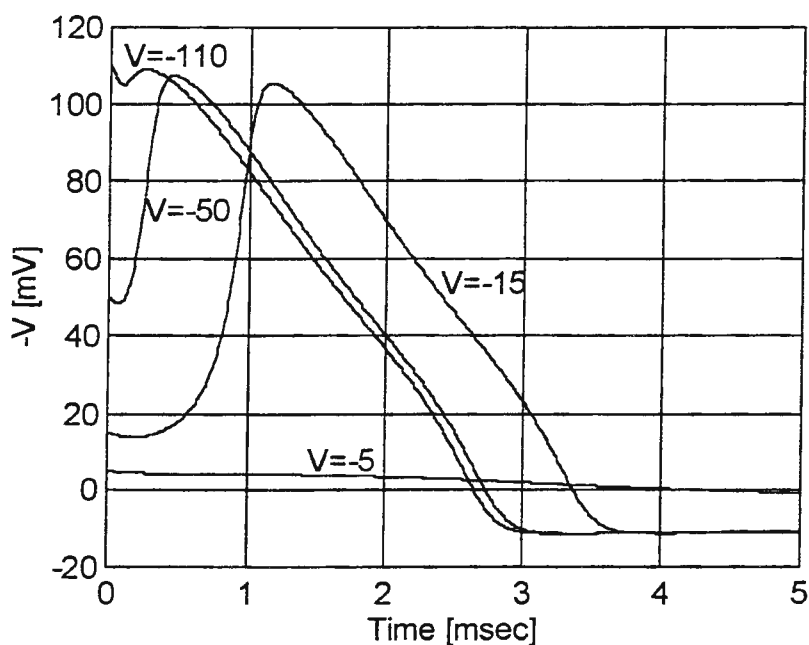

Fig. 2. The results of simulation

References

[1.] Hodgkin A. L., Huxley A. F., A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve, J. Physiol., Vol. 117, pp 500-544, 1952.

[2.] Jackson A. S., Analog Computation, McGraw-Hill Book Company, Inc., New York, 1960.

[3.] Cecic M., Zanchi V., Vlainic A., Vissim Simulation of Hodgkin-Huxley equation, Proceedings of 40th International Conference KoREMA, Zagreb, Croatia, pp 535-538, 1995.

MODEL AND THEORY

Martti Pekkanen
Helsinki University of Technology
Department of Chemical Engineering
02150 Espoo, Finland
martti.pekkanen@hut.fi

ABSTRACT

This presentation analyses the concepts of model and theory. The aim is to show that a model is a theory. If it is accepted that a model indeed is a theory, then all the considerations presented by the philosophy of science about theories can - and must - be applied to models. This has relevant implications on modeling and on model use - especially with respect to the questions of reality, knowledge and truth. This means questions about model verification, model falsification and model confirmation (or model validation).

THEORY

1. Law.

Within the context of philosophy of science a **lawlike generalization** (see e.g. Audi R., ed., *The Cambridge Dictionary of Philosophy*, Cambridge University Press, 1995) is a proposition that

- expresses a **physical necessity** and
- has **counterfactual force**.

The proposition 'Copper conducts electricity' is a lawlike generalization, because it expresses a physical necessity (i.e. 'Copper does not conduct electricity' is impossible) and it has counterfactual force, when it says that, **if** we would (which we do not) connect a potential difference over this (or any other) physical object of copper, **then** it would conduct electricity.

An **accidental generalization** is a generalization that is not lawlike. The proposition 'Electric wires are made of copper' is not an lawlike generalization but an accidental generalization, because it does not express a physical necessity or does not have counterfactual force, because it is possible that electric wires are made e.g. of aluminum.

A lawlike generalization is often called a **law**.

2. Theoretical and empirical.

Within philosophy of science 'a theoretical term' refers to an unobservable (not directly observable) entity, whereas 'an empirical term' refers to an observable (directly observable) entity. Examples of theoretical terms are e.g. electron, center of mass and social pressure, whereas table, mass and homicide are empirical terms. It is obvious that this theoretical/empirical -distinction is problematic in many ways, e.g. one may ask what 'observable' **means** and whether theoretical (unobservable) entities really **exist** or not? (Instumentalists say no and scientific realists say yes.)

Theoretical laws contain theoretical terms, whereas **empirical laws** do not.

3. Theory.

A **scientific theory** about a phenomenon is a systematic collection of empirical and theoretical laws about the phenomenon with existence assumptions for the - especially theoretical - entities.

A scientific theory both **explains** and **predicts** the phenomenon. A theory explains a phenomenon by giving the governing relations of the phenomenon. (The terms 'cause' and 'effect' are deliberately avoided here.) The explanation - the relations - can then be used for the prediction of the phenomenon in cases that exceed past experience.

MODEL

The word 'model' is used to refer to very many concepts and there is no **one** definition for the word 'model'. The most confusing - from the point of view of engineering - is the use of the word 'model' in logic (in predicate calculus), where it is possible to say that **the physical space is the model for a proposition**. For example, the physical space may be the model for the proposition '$\forall x [P_1(x) \rightarrow P_2(x)]$', where $P_1(x)$ = 'x is metal', $P_2(x)$ = 'x conducts electricity' and where $\rightarrow$ denotes implication, i.e. for the proposition 'All metals conduct electricity'.

This presentation is restricted to engineering, where the object to be modeled (the prototype) is (ultimately at least) a phenomenon (a fact, a state of affairs or an event) in the physical space. No definition for 'model' is attempted here. It is only understood that **a model is a representation** or a **description** of the physical phenomenon to be modeled.

A physical phenomenon can be represented or described (i.e. modeled) using a **physical model** or a **conceptual model**. For example, a ship can be modeled as a physical scale model used in laboratory experiments or as a conceptual model composed of e.g. mathematical equations. Given a wide interpretation for the word 'mathematical' all conceptual models (i.e. all non-physical models) can be said to be **mathematical models**, which is the common term used.

For simplicity, this presentation is restricted to **mathematical models of physical phenomena**.

1. Equation.

A mathematical **equation** used to represent or describe a physical phenomenon is a mathematically formulated proposition. The equation

$$F = \gamma \frac{m_1 m_2}{r^2} \tag{1}$$

is a proposition that expresses the **physical necessity** of how the magnitude of the force between two objects, F, depends on the masses of the two objects, m_1 and m_2, on the distance between the objects, r, and on a parameter, γ, that is assumed to be constant in space-time. Further, equation (1) has **counterfactual force**, when it says that, if we would (which we do not) stand on the planet Mars (or on any other planet), **then** the force on us would be of a magnitude predicted by equation (1). Accordingly, **an equation is a law.**

2. Theoretical and empirical.

The words 'theoretical' and 'empirical' as used with the word 'equation' seem to have different meanings than as used with the word 'law'.

It seems that in engineering 'theoretical equation' refers to a mechanistic equation, i.e. to an equation with the purpose (and with more or less success) of expressing the mechanism of a phenomenon. Thus 'theoretical' in this context does not refer to the concept of unobservable and it is perfectly possible to have a theoretical (mechanistic) equation for the description of empirical (observable) entities like a ship, speed, water and waves.

Consequently 'empirical equation' in engineering refers to an equation that does not try to express the mechanism of the phenomenon. An empirical equation is based on data obtained from observations (hence empirical) of the phenomenon. The relations between quantities found in the data are then compressed into what are usually called (empirical) correlations, that are statistical or probable in nature and not mechanistic.

Thus empirical equations can contain both empirical terms and theoretical terms in the sense of philosophy of science. (This is not only possible but also necessary, because it seems that all equations (and laws) are at the very end empirical and not theoretical (mechanistic), as it seems unlikely that the true structure of the world, i.e. the true mechanisms, will ever be known by man.)

3. Model.

A (mathematical) **model** as **a representation** or **a description** of a phenomenon (in the physical space) is a systematic collection of empirical and theoretical equations about the phenomenon with (usually implicit) existence assumptions for the entities.

A model (at least a good model) both **explains** and **predicts** the phenomenon. A model explains the phenomenon either mechanistically (theoretically) or statistically (empirically), which can then be used for the prediction of the phenomenon in cases that exceed past experience.

Because an equation is a law, both models and theories are systematic collections of laws. Further, both models and theories explain and predict phenomena. Thus, finally, it can be concluded that **a model is a theory**.

ONTOLOGY

According to Karl Popper (*Objective Knowledge*, Oxford University Press, 2nd ed., 1979, p. 106) the reality - i.e. that what exists - consists of three "worlds":
- world 1: physical objects
- world 2: (human) consciousness and mental states
- world 3: objective contents of thought: numbers, propositions, science, ...

If this is accepted, it seems obvious that both mathematical models and theories (after their publication) are entities of the world 3. Thus ontology is an additional link between models and theories.

To make (scientific) communication between human beings possible, both models and theories must be realized from world 3 to world 1 (via world 2). This can be done e.g. by spreading ink on paper or by producing pressure waves in air in ways that can be interpreted by human beings.

For an understanding of models (theories) it is thus crucial to conceptually separate the physical realization of a model (theory) from the thoughts about a model (theory) and the objective and public contents of a model (theory): A model is not the physical realization of the model, but rather what the physical realization says, i.e. the propositions the physical realization expresses.

Thus a model does not depend on the physical realization - whereas the physical realization depends on the model. Accordingly the following physical realizations express the same model (proposition): 'Two times two is four', 'Zwei mal zwei ist vier' and '2·2 =4'.

CONFIRMATION AND VALIDATION

"Our purpose in this chapter is primarily to emphasize the significance of the problem, and to propose that even the most straightforward considerations of model validation requires concepts in the philosophy of science that are not within the experience of most engineers." (M. Denn, *Process Modeling*, Longman, 1986, p. 271). Because the presentation of Denn, however, is conceptually unclear (at least), it cannot be recommended. For example, according to Denn: "... one never proves that a theory is correct (or validates a model!)"

'**Verification**' means epistemic justification, i.e. the establishment of truth. Verification is **absolute** in the sense that truth is made known with absolute certainty. Truth - in the sense of the correspondence theory of truth (there are other theories of truth) - is an **objective** concept, because the truth does not depend on whether the truth is known or not. Thus, finally, verification is both absolute and objective. Because it seems impossible to establish a physical phenomenon with absolute certainty, verification is impossible (in the sense of the correspondence theory of truth).

'**Confirmation**' means a weak form of verification. Thus, if a proposition is found only more or less probably true or more or less near the truth, this activity is called confirmation. This makes confirmation **relative** and not absolute. Further, because confirmation requires human judgment (as to what is "probable" or "near"), confirmation is **subjective** and not objective.

Further, confirmation is always **local**, i.e. confined to the areas of experience, and **fallible**, i.e. subject to the possibility of errors somewhere in the procedure.

NB! The everyday sense of confirmation (e.g. confirmation of flight tickets) comes close to verification. For example, The Random House Dictionary gives for 'confirm': **1.** to prove to be right or true.

In practice a proposition (law, hypothesis or theory) is confirmed through the comparison of the predictions of the proposition with **physical evidence** (observations or experiments in the physical space). The validation of a model means that the predictions of the model are compared with physical evidence. Thus what actually is **confirmation of models** is usually called **validation of models** (and sometimes verification, which is false, however). In summary:

Verification is impossible.　　　　　**Validation is confirmation.**

CONCLUSIONS

The analysis shows that a model is a theory. Both models and theories consist of laws (proposition that express a physical necessity and have counterfactual force), explain and predict phenomena and are entities of the objective and public world 3.

Because a model is a theory, all the considerations presented by the philosophy of science about theories can - and must - be applied to models. The most important example of such considerations is the relation of a model to the physical space, i.e. the question of truth of a model. Because it is impossible to establish the truth of a model with absolute certainty, verification is impossible. Thus only the relative, subjective, local and fallible confirmation or validation of a model is possible.

PROCESS MODELING, SIMULATION AND DESIGN

Martti Pekkanen
Helsinki University of Technology
Department of Chemical Engineering
02150 Espoo, Finland
martti.pekkanen@hut.fi

ABSTRACT

This presentation analyses the terms of 'model', 'modeling', 'simulation' and 'design' within the context of industrial processes. The main conclusions are: 1) Modeling is more than the act of producing the model, i.e. the solution of the model and the validation of the model are integral parts of modeling. 2) Simulation and design are two modes of solution of the model and thus simulation and design are modeling. 3) The crucial task - and the least studied task - within process modeling is model validation.

PROCESS AND MODEL

The concept of **process** is not defined here due to space limitations and thus an intuitive understanding for the concept has to be used. In the context of industrial processes the word 'process' may used to refer to a **whole of a process** (ammonia process, ...) or to a **process unit**. Process units are divided into **unit operations** (physical changes: distillation, mixing, ...) and **unit processes** (reactions: alkylation, polymerization, ...).

The concept of **model** is not defined here. It is only understood that **a model is a representation**, where 'representation' is again left undefined.

Within the context of industrial processes the word 'process' can be used to refer to 1) a physical process, 2) an thought of process and 3) a process model. Because subjective mental states are not considered within process engineering science, only physical processes and process models are considered here.

Analogously, 'process model' can be used to refer to 1) a physical process model, 2) an thought of process model and 3) a conceptual process model, which means a non-physical public process model. Again subjective mental models are not considered.

If it is understood that a process model represents the process and a process model is obtained from the process through modeling, the following relations are obtained:
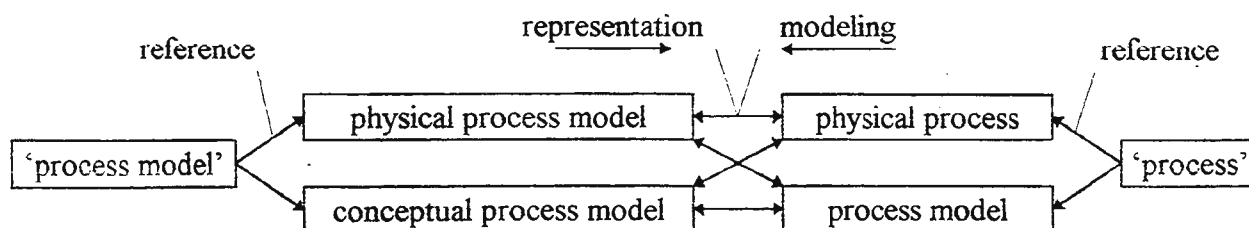

Figure 1: The 'process' and 'process model' in science

MATHEMATICAL PROCESS MODEL

The most common conceptual process model is a **mathematical process model** (with a suitably wide definition for 'mathematical' these are identical). A mathematical (process) model consists of
* **quantities** and
* **equations** connecting the quantities.
The kinds of equations in mathematical models are not considered here.

Quantities.

It is customary to classify the quantities in a mathematical model as 1) variables, 2) parameters and 3) constants. There is a problem, however, because within human knowledge there are **no true constants**. The speed of light, c, may be a true constant of nature, but within human knowledge c is no more constant, because the numerical value of c varies in time as the accuracy increases or errors are detected. Further, we may wish to use the model with a speed of light twice the "real" speed.

Accordingly the concept of a constant is abandoned and quantities are classified into variables and parameters:
* **Variables** convey (causal) interactions.
  * **Input variables** convey (causal) interactions (i.e. causes) into the model.
  * **Output variables** convey (causal) interactions (i.e. effects) out of the model.
  * **Internal variables** convey (causal) interactions between the subphenomena within the model.
* **Parameters** modify the expressions and the model structure (i.e. define a specific case).
  * **Process parameters** that are **specific** to the process.
  * **Non-process parameters** that are **not specific** to the process.
For example:

Process	Process parameters: specific to the process	Non-process parameters: not specific to the process
distillation	number of plates, plate dimensions, packing height, packing size, ...	physical properties of the chemicals, e.g. vapor pressures, ...
reaction	size and shape of the reactor, catalyst particle size and shape, ...	physical properties, chemical properties, ...
heat exchange	heat exchange area, number of tubes, tube (plate) dimensions, ...	physical properties, ...
mixing	dimensions and size of the mixer, ...	physical properties, acceleration due to gravity, ...

It is seen that the non-process parameters come close to what customary is called a "constant".

PROCESS AND PROCESS MODEL

The relation of a (mathematical) process model and its constituents to the (physical) process and its constituents can be represented as follows:
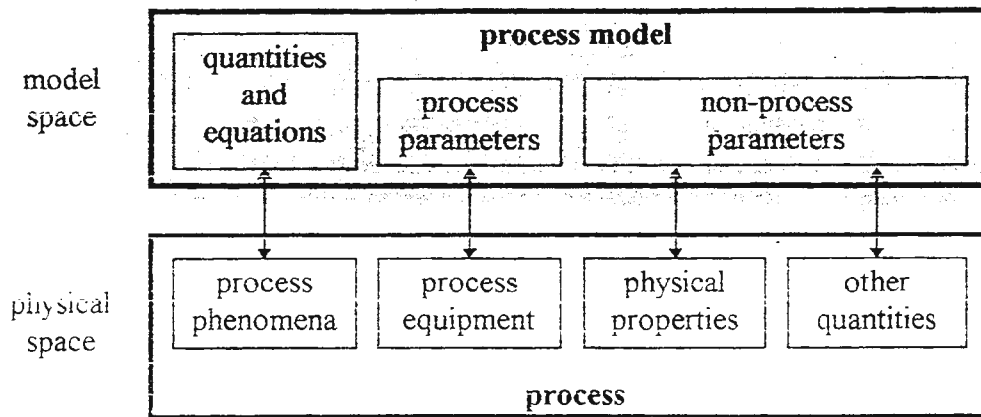

Figure 2: A process and a process model

The arrows between the spaces describe representation (from the model space to the physical space) and modeling (from the physical space to the model space) as in figure 1.

MODEL USES

A mathematical process model may be used in three basic ways:
- **Simulation (analysis):**   The calculation of the **output variables**.
- **Design:**   The calculation of the **process parameters**.
- **Parameter estimation:**   The calculation of the **non-process parameters**.

Parameter estimation is usually not done using a process model, but this use is mentioned here for completeness.

The use of a model usually means the (numerical) solution of the model. In any solution of a model the specifications (all other quantities not to be calculated) have to be fixed. Thus the model uses can be described as:

Solution mode	Calculated	Specifications
Simulation	output variables	input variables process parameters non-process parameters
Design	process parameters	input variables output variables non-process parameters
Parameter estimation	non-process parameters	input variables output variables process parameters

MODELING TASKS

A maxim of modeling says that no modeling activity is finished before the model is solved. Further, "model validation is an essential component of any study, since the model must be shown to be reliable .." (M. Denn, *Process Modeling*, Longman, 1986, p. 270). Thus it is seen that "modeling" is not equal to the production of "the model", but is wider and must include both the solution of the model and the validation of the model. Especially note that the production of the model expressions is but a part - and often a small part - of modeling.

Thus modeling can be divided into (at least) five separate tasks:

Modeling1 (Conceptual modeling)
• Evaluate and identify the relevant phenomena.
• Identify the relevant quantities and equations, i.e. model expressions.
Product: Model expressions.
Modeling2 (Applied modeling)
• Derive or acquire the algorithm for the solution of the model.
• Implement the model and the algorithm for solution.
Product: Solution algorithm.
Modeling3 (Experimentation)
• Evaluate the non-process parameters by experimentation or literature.
Product: The non-process parameters.
Modeling4 (Solution)
• Evaluate the specifications for solution.
⊕ Solve
Product: The solution.
Modeling5 (Validation)
• Validate the solution with respect to the phenomena in the process.
• If the solution is not valid, reconsider previous tasks.
Product: A valid model.

The tasks - with what may be called "the model", i.e. expressions, algorithm and parameters - can also be presented as:
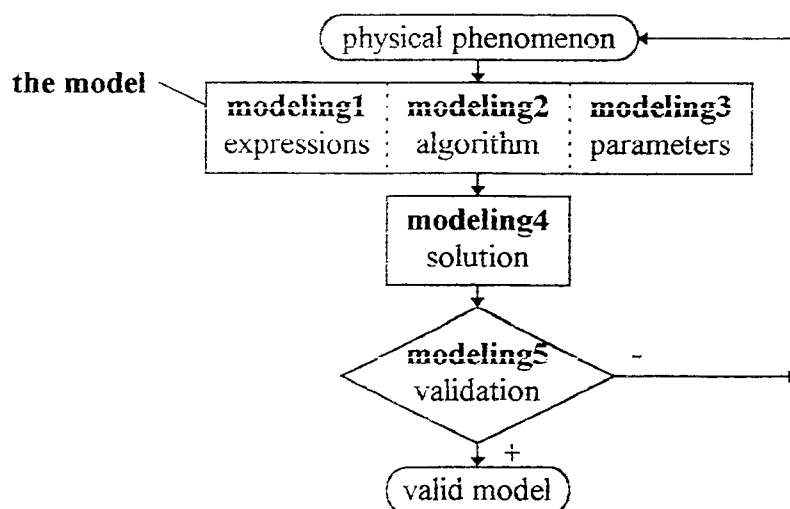

Figure 3: Modeling tasks

SIMULATION AND DESIGN

It seems that there are (at least) two possible modes of modeling and model use (simulation and design):

- **The conventional mode**, where it is understood that the product of modeling is a valid model. This valid model is then used for simulation and design. This implicitly entails the notion of absolute and global validation. This mode also excludes simulation and design (as model use) from modeling.

- **The critical mode**, where it is understood - and emphasized - that validation is always **subjective** (depends on human judgment as to what is "valid"), **relative** (a model can be more or less valid), **local** (depends on the local conditions where the model is found valid) and **fallible** (validation may contain errors). Thus this mode incorporates simulation and design in modeling as two modes of solution before model validation.

The two modes of modeling and model use can be presented as follows:
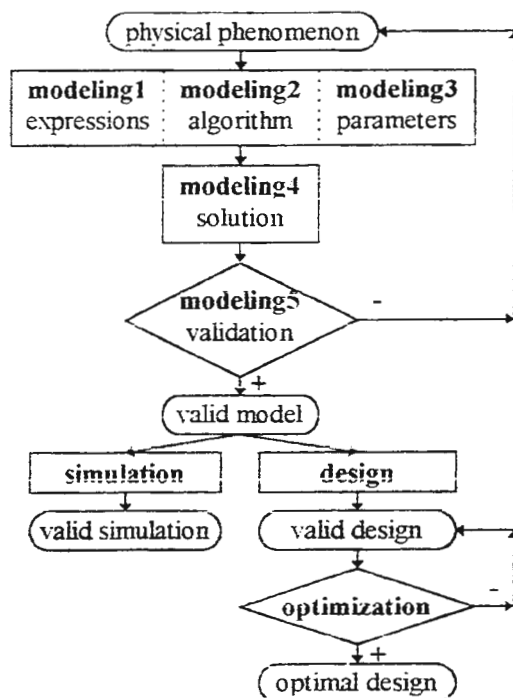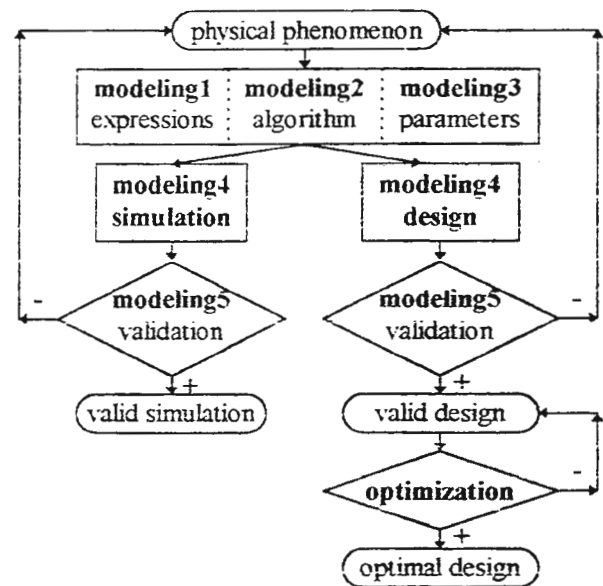

Figure 4a: The conventional mode   Figure 4b: The critical mode

CONCLUSIONS

An analysis shows that within industrial processes the word 'process' can be used to refer to a physical process and a process model and 'process model' can be used to refer to a physical process model and a mathematical process model.

Mathematical process models are shown to contain input, output and internal variables and process parameters and non-process parameters. This analysis enables the definition of simulation and design as modes of model solution.

It is shown that modeling is not equal to the production of "the model", but is wider and must include both the solution of the model and the validation of the model. This leads to the division of modeling into five separate modeling tasks.

The analysis emphasizes validation. Validation not only precedes any model use (simulation or design) as a modeling task but must also be included in any modeling use as the validation of the solution obtained. This critical (and holistic) view thus incorporates both simulation and design in modeling as two different modes of model solution.

Bassist — a tool for MCMC simulation of statistical models

Hannu Toivonen, Heikki Mannila, Marko Salmenkivi, and Karri-Pekka Laakso

University of Helsinki, Department of Computer Science, PO Box 26, FIN-00014 University of Helsinki, Finland
Hannu.Toivonen, Heikki.Mannila, Marko.Salmenkivi, Karri-Pekka.Laakso@cs.helsinki.fi

Introduction

Bassist is a simulation tool for the analysis of complex statistical models. Given a high-level specification of a full probability model, Bassist generates a simulator for analysing the model with respect to given data. Bassist follows the Bayesian modeling approach. Model parameters and other unknown quantities are assigned prior distributions by the modeler; the likelihood of observed data follows from the specification of the statistical model. The matter of interest is the posterior probability of the model, i.e., the joint probability distribution of all unknown quantities conditioned on the observed data. Bayesian statistics provide a well-founded means for assigning probabilities for models. For most realistic models, however, analytical methods for the computation of the posterior distribution are not feasible. To solve the problem, Bassist applies Markov chain Monte Carlo (MCMC) simulation techniques to approximate a sample from the joint posterior distribution.

MCMC simulation techniques are gaining popularity in the statistical modeling community, due to the growth of computational power to the level needed for serious simulations, and the success of Bayesian modeling. General purpose MCMC simulation software, however, has been almost non-existent, except for [9] and the Bugs system [12]. Simulator programs have been written specifically for each model, either from scratch of by reusing previous code. In either case, the task is tedious and error-prone. Bassist aims at helping modeler at implementing and analysing their models. In this paper we give a short overview of MCMC simulation and the Bassist system, and describe some of the applications in which Bassist has been used.

Full probability models and MCMC simulation

Bassist is a simulation tool for the analysis of hierarchical Bayesian models, also called full probability models; see, e.g., [6] for an excellent overview. Such a model describes the phenomenon of interest by a set of quantities and their statistical relationships. In a typical setting, some of the quantities are parameters of the model, and the model specifies the probability distribution of the rest of the quantities, or data, given values of the parameters. In a full probability model, the model parameters have prior probability distributions and the result is a distribution on the parameters, not just a point estimate (as in, e.g., maximum likelihood approaches).

In Bassist terminology, all quantities considered in a model are called variables, whether actually observed facts or model or hyper parameters. The full probability model is expressed as the product of densities of all variables in the model. The densities may be conditional on other variables, as long as the chain rule applies and the product of the specified densities equals (or is proportional to) the joint probability of all variables. In a typical simple setting, the density of data depends on model parameters and distributions of model parameters are expressed as unconditional prior probabilities, but there can be an arbitrary number of layers of conditional distributions.

More formally, a hierarchical Bayesian model is defined by giving a directed acyclic dependency graph $(V, \leq)$ where V is the set of model variables. Given a variable $v \in V$ and the set $pred(v) \subset V$ of its immediate predecessors in a dependency graph, the variable v is independent of other variables. The model specification is completed by defining the conditional probability $\Pr(v \mid pred(v))$ for all $v \in V$. The joint probability distribution of all variables is then $\prod_{v \in V} \Pr(v \mid pred(v))$.

In Bayesian analysis some variables are given as data, and the task is to compute the (posterior) distribution of the remaining variables. Let $D \subset V$ denote the set of variables given as data, and let $\theta = V \setminus D$ be the set of variables to be simulated; the task is to find the distribution $\Pr(\theta \mid D)$. Bayesian inference is a central tool used in this task. Denote by $succ(\theta) = \{w \in V \mid \exists v \in \theta \cap pred(w)\}$ the set of immediate successors of θ in the dependency graph. By the application of Bayes' Theorem we have that the posterior density of θ is proportional to the product of the prior density and likelihood of θ:

$$\Pr(\theta \mid D) = \frac{\Pr(\theta \wedge D)}{\Pr(D)} \propto \Pr(\theta \wedge D) \propto \prod_{v \in \theta} \Pr(v \mid pred(v)) \prod_{w \in succ(\theta)} \Pr(w \mid pred(w)). \tag{1}$$

MCMC and the Metropolis–Hastings method

The idea in MCMC techniques is to define a Markov chain with the domain of θ as state space, and the distribution $\Pr(\theta \mid D)$ as stationary distribution. A sample from the posterior distribution $\Pr(\theta \mid D)$ is obtained by simulating the Markov chain and outputting states of the chain as they are visited.

The Metropolis–Hastings method [8] used in Bassist constructs such a Markov chain by repeating, in each state θ_{i-1}, the following two steps. First, a possible next state θ_i is drawn from some distribution $q(\theta_{i-1}, \theta_i)$ (the proposal distribution). Then the new state θ_i is accepted with probability

$$\min\left(1, \frac{\Pr(\theta_i \mid D) q(\theta_i, \theta_{i-1})}{\Pr(\theta_{i-1} \mid D) q(\theta_{i-1}, \theta_i)}\right). \tag{2}$$

Unlike the more widely used Gibbs sampling [11], the Metropolis-Hastings algorithm is not limited to conjugate distributions but can handle (in principle) arbitrary distributions. Equation (1) shows how the posterior density can be computed from the dependency graph; the only computational requirement is that the user-specified conditional densities can be computed. By the application of the Metropolis–Hastings method (formula (2)), one can simulate samples from extremely complex posterior distributions. Metropolis-Hastings-algorithm can even be applied so that the dimension of the model is also a random variable and thus the amount of parameters can change during simulation [7]. This property makes it superior to Gibbs sampling in many modeling tasks by allowing non-parametric models with an arbitrary amount of parameters.

For computational convenience, usually only one parameter is updated at a time, i.e., vector θ_i differs from θ_{i-1} in one position only. In Bassist, the default for the proposal q is for each variable its own prior distribution unless the distribution is a user-defined function. In such cases the user must specify a suitable q, such as a random walk method. Bassist also supports updates of multiple variables at a time.

Design of Bassist

Bassist is a domain independent, general purpose tool for Bayesian analysis. Bassist compiles a model specification to a simulation program that performs Metropolis-Hastings sampling to obtain an approximation of the posterior distribution of the model. In addition to models that can be expressed as simple and finite graphs, models can also contain non-parametric components, including piece-wise constant functions with a varying number of pieces. Another special feature supported by Bassist is the processing of event sequences and intensity models [1,3]. This makes the tool quite useful for modeling event data ubiquitous in epidemiology, fault management, and quality control.

For Bassist, the major differences when compared to Bugs [12] are the capability of simulating models with complex non-parametric components and with event sequences. In principle, models of arbitrary probability distributions can be simulated as long as the user is able to specify the corresponding density functions.

Bassist allows the user to specify a full probability model using a high-level model description language. The model specification language of Bassist is declarative, and it has been designed to have a clear separation between the model specification, the simulation details, and the particulars of the data. The language is based on a data modeling view of hierarchical Bayesian models; this makes the language easy to use and suitable for data mining applications.

Bassist is implemented as a simulator generator. The model specifications are first transformed into a C++ program which is then compiled and linked with the Bassist library. The produced executable reads the data and performs the actual simulation. Produced programs can simulate any combinations of variables of the model: the input data determines which parts are given as data. This design not only allows easier experimentation but also makes the model specification easier.

The system uses the class mechanisms of C++ to obtain a clean and fast implementation of parameter updates in the Metropolis–Hastings algorithm. Constant parts of simulators are available as components in a library. These include, for instance, classes for variables, standard distributions, and sampling. Depending on the model specification, code is generated to cope with the dependency structures, proposal methods, and user-defined density functions. The density functions are provided by the user as C++ code that is linked to the system, so there are practically no limits for them. While Bassist helps in the implementation of models, the user naturally remains responsible for the correctness and convergence of the model. – Bassist is currently under development. A publicly available version will appear in 1998.

Example applications

Palaeoecological reconstruction

In the palaeoecological reconstruction task of [10] the goal is to estimate past temperatures based on fossil assemblages of chironomid taxa. The idea is to specify an ecologically plausible model for the response of the organisms to lake temperature, fit this model using modern data where both temperatures and taxa abundances are known, and finally apply this model in the reconstruction task where only organism abundances are known.

The central part of our model for this task is the specification of how the abundances of taxa respond to the lake temperature. Following the traditional response-curve model in palaeoecology [2], we model the response function of a taxon k to temperature z as a bell-shaped curve, defined by three parameters $\alpha_k, \beta_k, \gamma_k$:

$$r(\alpha_k, \beta_k, \gamma_k, z) = \alpha_k \exp\left(\frac{(z - \beta_k)^2}{\gamma_k}\right).$$

Here β_k is the optimum temperature for the taxon and the location of the maximum of the response curve, α_k is the height of the curve at the maximum, and γ_k determines how steep the curve is. Given a temperature z, we assume the abundances of taxon k are Poisson distributed with parameter $r(\alpha_k, \beta_k, \gamma_k, z)$. For simplicity, we assume here the taxa are independent. To complete the model, we specify prior distributions for the parameters $\alpha_k, \beta_k, \gamma_k, z$ such that the probability of a value combination reflects the probability of such response function occurring in the nature. Note that throughout the process of designing the model the role of domain experts is essential. In the provisional Bassist language the model is expressed in the following way.

```
record taxon {
      var alpha  ~ Unif(0.1, 100),
      var beta  ~ Normal(12.1, 4),
      var gamma  ~ Gamma(50, 5)
};
record lake {
      var z  ~ Normal(12.1, 0.4)
};
record measurement {
      ref taxon,
      ref lake,
      var amount  ~ Poisson(r(taxon.alpha, taxon.beta, taxon.gamma, lake.z))
};
```

The function r is defined elsewhere in C++. The user may additionally specify proposal methods for parameters and in this way affect the convergence. The actual data, and the number of iterations and burn-in iterations are specified at run time.

Now with the modern calibration set we know the actual responses of each taxon in a number of lakes, and the parameters $\alpha_k, \beta_k, \gamma_k$ are simulated for each taxon k. In the actual reconstruction, temperatures of unknown lakes are simulated based on the current values of response parameters. Within Bassist, every unknown variable is simulated with the same principles and mechanisms. There is no need at all to discriminate, e.g., between model parameters, data, and missing data, and actually the whole simulation process can be performed at once: all parameters $\alpha_k, \beta_k, \gamma_k$ are unknown, whereas some of the temperatures z are known and some unknown. A more detailed description of the model can be found in [13].

Figure 1 below shows the trace of a simulated temperature during the first 1000 iterations. In the setting of the figure, much longer runs are necessary to ensure convergence. Figure 2 below displays a summary of the result for the period of last 10000 years.
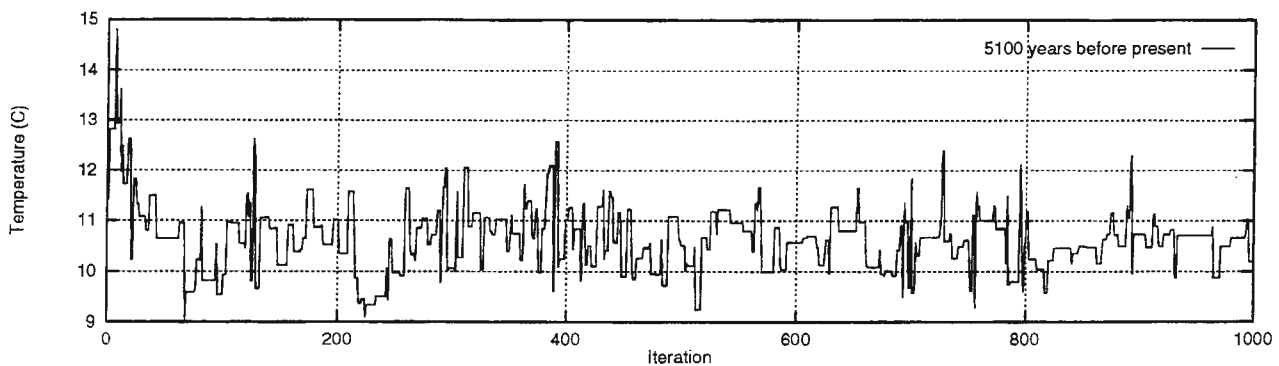

Figure 1: The trace of the first 1000 iterations of a simulated temperature of approximately 5100 years ago.
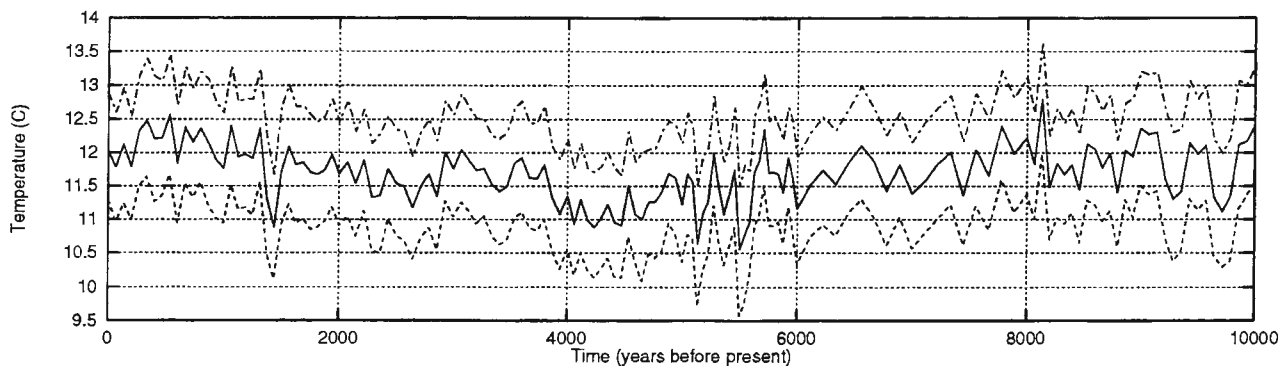

Figure 2: A result of a simulation for the whole reconstruction problem: solid line is posterior mean, dotted lines are 90 % credibility intervals.

Ear infections

As an example of modeling event sequences we discuss analysing occurrences of middle ear infections in small children. In addition to the occurrences of ear infection episodes, our data consists of changes in several risk factors. For illustration purposes we will only consider here the effects of two risk factors: weaning and day care attendance. A more detailed description of the problem is found in [5].

Intensity functions are a common methodological tool in applications of this kind. They are used to describe the instantaneous risk of observing the event of interest over time [1,3]. We modeled the age-dependent intensity by a piece-wise constant function. The levels of the function, the jump times and also the number of

pieces were random variables. We added observed history in terms of time-dependent indicator functions of weaning and day care multiplicatively into the model. To separate the effects for the first and subsequent infections, we used the following parameterisation [4] to model the infection intensity $\lambda_i(t)$ (for individual i):

$$\lambda_i(t) = f(t) \cdot Y_i(t) \cdot \beta_i(t) \cdot \gamma_i(t) \cdot \alpha_i(t) \cdot \beta_i^*(t) \cdot \gamma_i^*(t)$$

Here $f(t)$ is the piece-wise constant baseline function and $Y_i(t)$ is an indicator function of being at risk at time t. The time-dependent functions of weaning ($\beta_i(t)$) and day care ($\gamma_i(t)$) indicate the relative risk of these factors compared to baseline hazard $f(t)$. The value of β_i is defined to be 1 after weaning, otherwise it is a parameter (β) to be simulated. Similarly, the value of γ_i is 1 before day care attendance and a random variable (γ) after that time. The values of β or $\gamma < 1$ correspond to protective effect and the values > 1 to increased risk. The indicator function $\alpha_i(t)$ is used to distinguish between those who have had at least one infection already. The functions $\beta_i^*(t)$ and $\gamma_i^*(t)$ correspond to the additional effect of weaning and day care, respectively, on recurrent infections compared to the effect on the first infection. The total relative risk for recurrences is therefore $\beta \cdot \beta^*$ (and $\gamma \cdot \gamma^*$ respectively).
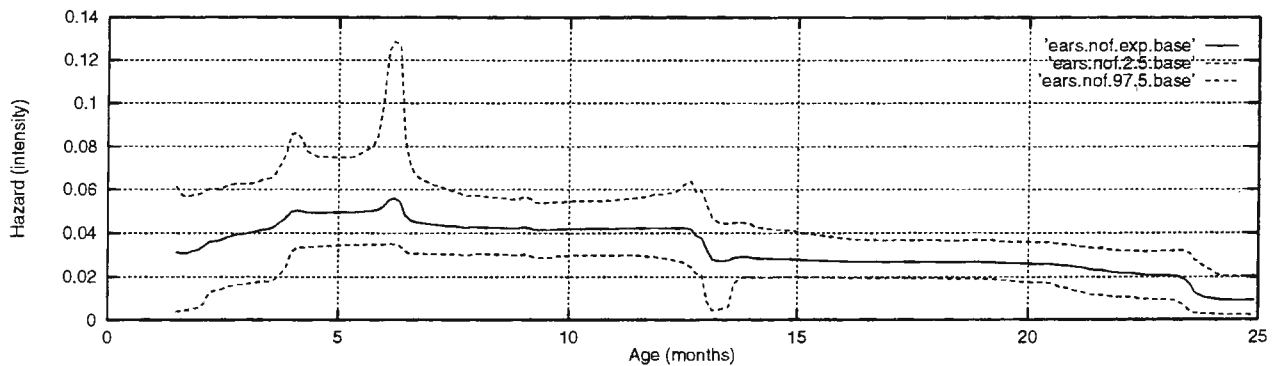

Figure 3. Posterior mean and 95 %-probability intervals of the piece-wise constant baseline function f(t).
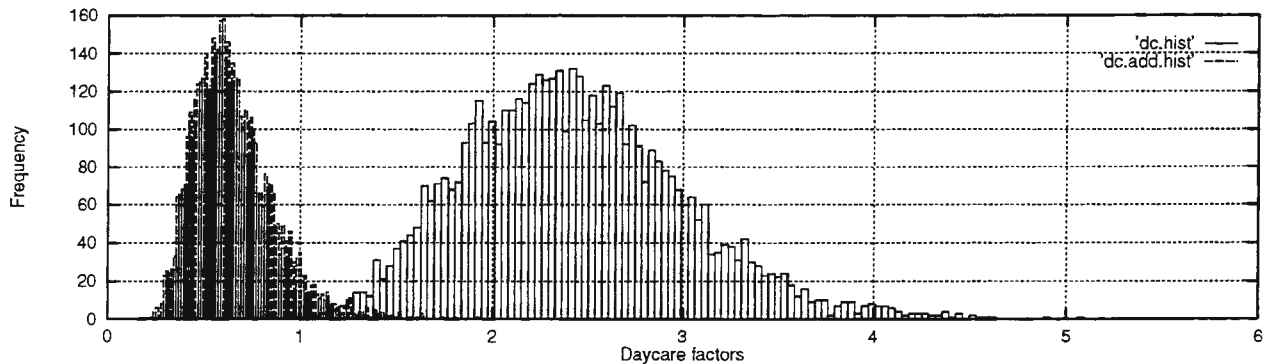

Figure 4. Posterior distributions of the parameters γ and γ^* that were related to the day care attendance.

The point-wise averages and the 95 %-probability intervals of the piece-wise constant posterior baseline function $f(t)$ are shown in Figure 3. The strong risk effect of attending day care for the first infection indicated by high values of parameter γ loses significance for recurrent events, as the interaction parameter γ^* is clearly below 1. The posterior distributions of these factors are presented in Figure 4. The simulation was run for 50000 updates for each parameter and the parameter values were picked up in every tenth round.

Conclusions

We have described Bassist, a tool for the simulation and analysis of statistical models. Bassist is based on the Bayesian full probability paradigm, and it uses MCMC methods to approximate complex posterior

distributions. Bassist is targeted mainly at the statistical modeling community, where Bayesian modeling is gaining popularity but the implementation of simulators is a burden.

We described two applications of Bassist. Bayesian modeling provides sound conceptual and mathematical tools for reasoning with complex probability models, and for incorporating available prior knowledge to the reasoning. In the reconstruction task this means that the models can and should be ecologically realistic and plausible, irrespective of their possible statistical complexity. Reasoning on full distributions is another advantage. Distributions give more information than simple point estimates available, e.g., from maximum a posterior analysis. Distributions are especially useful in the estimation of the uncertainty of the inference.

Acknowledgements

References

[1]   Arjas, E.: Survival models and martingale dynamics. *Scandinavian Journal of Statistics*, 16, p. 177–225, 1989.

[2]   Birks, H. J. B.: Quantitative palaeoenvironmental reconstructions. *Statistical modelling of Quaternary science data, Technical Guide 5,* edited by Maddy, D. and Brew, J.S. , p. 161–254, Quaternary Research Association, Cambridge, 1995.

[3]   Cox, D. R.; Oakes, D.: Analysis of survival data. *Monographs on Statistics and Applied Probability 21.* Chapman & Hall, 1984.

[4]   Eerola, M.: Repeatable events in event-history analysis; the effect of childhood separation on future mental health. *Bull. Int. Statist.Inst. vol LIII(1)*, p. 213-225, 1989.

[5]   Eerola, M., Mannila, H., Salmenkivi, M.: Frailty factors and time-dependent hazards in modelling ear infections in children using Bassist. Preliminary version. To appear in *Proceedings of 13th Symposium in Computational Statistics*, Bristol, 1998.

[6]   Gelman, A., Carlin, J., Stein, H., Rubin, D.: Bayesian Data Analysis. Texts in Statistical Science. Chapmann & Hall, 1995.

[7]   Green P.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination.. *Biometrica*, 82, 4 , p. 711-732, 1995.

[8]   Hastings, W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrica*, 57, p. 97–109, 1970.

[9]   Neal, R. M.: Bayesian learning for Neural Networks. Ph.D. thesis, Dept. of Computer Science, University of Toronto, 1994.

[10]   Olander, H.; Korhola, A.; Birks, H. J. B.; Blom, T.: Surface sediment Chironomidae (Insecta: Diptera) distributions along an ecotonal transect in subarctic Fennoscandia: developing a tool for palaeotemperature reconstructions. *Journal of Paleolimnology,* 18, p. 45–59, 1997.

[11]   Smith, A. F. M.; Roberts, G. O.: Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society*, Series B, 55, No. 1, p. 3–23.

[12]   Spiegelhalter, D. J.; Thomas, A.; Best, N. G.; Gilks, W. R.: BUGS Version 0.5, Bayesian inference using Gibbs sampling, MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK, 1995.

[13]   Toivonen, H; Mannila, H; Korhola, A.: Paleoecological reconstruction using Bayesian methods and MCMC simulation. Manuscript, 1998.

Simulation-based Training Using SITA

G.Jonins[1], Dz.Tomsons[2]
[1] Institute of Mathematics and Computer Science, University of Latvia, Rainis boulv.29, LV-1459 Riga, Latvia, jonin@cclu.lv
[2] Liepaja Pedagogical Academy, Liela Street 14, LV-3400 Liepaja, Latvia, dzt@cs.lpu.lv

Abstract

The current paper presents the possibilities to apply simulation in training process, incorporating software tools designed for several classes of tasks. Implementation of integrated simulation-based training courses is highlighted. The described approach is illustrated by the use of discrete event simulation system SITA.

Training and Simulation

Training is a multi-sided process influenced by numerous factors. Knowledge (theoretical domain) and skills (practical domain) are more essential sides of that. Simulation is the applied branch of the science. Thus, it is used mainly to develop skills and get experience of practical work. That is comparatively easy to manage simulation experiments. However, user must carefully to consider and clearly understand investigated processes. Otherwise simulation model can turn into environment for generation of tries and errors.

Usually, several courses on specific, quiet closed knowledge domains are provided, for example, (1) on the structure of the given system and its operating principles, (2) on the system design and specification techniques, (3) on the planning and analysis of the system operation, (4) on mathematical statistics, its applications to the system analysis, (5) on the design and implementation of experiments, (6) on optimization and forecasting, etc. In every such course, student gets more or less closed knowledge on some part of the problem-solving process. The ability to tie these separated parts of knowledge are profited by personal experience in practice. Modern simulation software tools provide possibilities to combine the knowledge domain of these separate courses. Such approach has an amount of advantages:

- Depending on specific features of the model domain, it can be used for training of specialists in different branches.
- Student develops skills initialized in several courses.
- The same simulation model can be used for different purposes, for example, generation of research data, demonstration of processes in the real system, experimentation, etc.
- The use of simulators as data generator is essential feature. That allows to develop skills for analysis with different values of the system parameters.
- Student gets deeper understanding on essential factors in the system.
- Using simulation, student develops creative approach to his/her future work.

Simulation Tools

In initial and narrow sense, simulation can be expected as a technique for computation and data processing. At present, simulation software tools are supplemented by means of different task classes, for example, system design and specification, data visualization, animation, data analysis, forecasting, etc. The reason of such development is following: system management involves all of the mentioned activities.

There are different ways, how to combine simulation with tools of other kind of tasks -- (1) simulation software is supplemented with specific functions, (2) simulation features are added to specific software systems, (3) interface is built between simulation environment and the one of specific tasks.

Building some additional functions into some software system, user must pay less effort to learn how to manage with the tools, because everything needed is in the same package. However, built-in tools are restricted by features of base environment. More general way is the last approach. User can apply more efficiently facilities of both simulation and specific tools. Simulation system SITA provides facilities for the approaches (1) and (3). The system has amount of built-in tools for system description, processing and

analysis of simulation results, visualization and animation, interactive simulation, etc. Besides, there are implemented several data interfaces between SITA and animation, statistics, database and other software systems.

Case study
Description of the systems model

Let us look at the operations of a harbor with V piers (V = 1, 2, 3, ...). The harbor is separated from sea by narrow channel providing movement of only one ship through that simultaneously. Its scene is shown in the Fig.4. Let us investigate how efficiency of the harbor operation is influenced by widening the channel and creating additional piers? In the current case, coefficient of the efficiency is inverse proportional to the average time spent by the ships in the harbor.

In order to solve the problem, it is necessary (1) to describe the given system, (2) to choose more essential characteristics of the harbor, (3) to estimate their interdependencies, (4) to design and debug simulation model, (5) to design and run simulation experiments, and (6) to proceed and analyze the results of experiments. The given system can be described as queuing system. Its action scheme, design by SITA, is show in Fig.1. The incoming ships constitute Poisson's flow with parameter L (L > 0). The time, necessary for service of the ships, has exponential distribution with parameter M (M > 0). The arrived ship takes a free pier. If it cannot find free pier, that stays in sea near the entry into harbor.
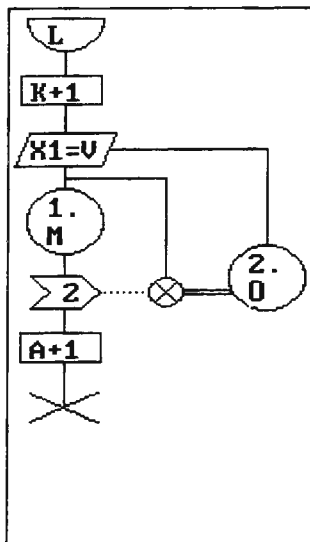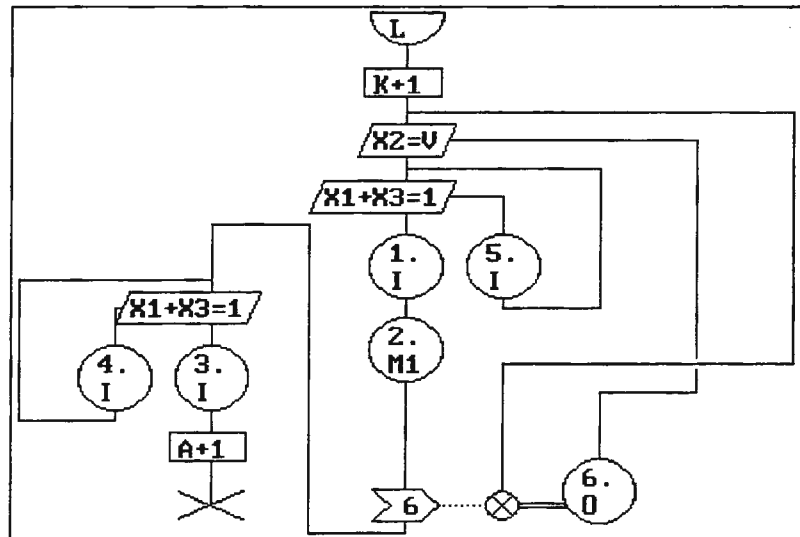

Fig.1. Simulation model the harbor

Fig.2. The harbor simulation model for animation.

Calculations and Statistical Analysis

The simulation system SITA allows to run in iterative way the simulation experiments and store the gotten results. Its in-built tools provide possibility to calculate values of chosen characteristics, their average values, confidence intervals, and to visualization of the simulation results. In Fig.3, correspondence between incoming load L and average time spent in queue is depicted. The representation is gotten by SITA Graphic system. In order to use more complicated data processing and analysis techniques, like calculation of correlation and regression, hypothesis testing, analysis of time-series, etc., it is necessary to use specialized software, e.g., MINITAB, SPSS, SAS, Statistica, etc. The simulation system SITA provides interface to different software systems of data processing and analysis.

In the given case, the following situation is investigated more detailed. The load L/V of incoming ships per a pier is as twice as service load M. Using SITA and MINITAB tools, the following inferences can be gotten. (1) Widening of a short channel there is not sensible improvement in the harbor's operation. (2) Shortening service time affect the system behaviors more strongly than in case of increasing the number of the piers.

The animation of simulation models is convenient way to show the operation of simulated system and to be convinced in its correctness. The main advantages of simulation animation find expression in the debugging of simulation programs, in validation and verification of simulation models, in study of system dynamic behaviors, and in training. Fig.4 represents screen-copy of animation scene, designed by ProofAnimation.

However, the design of animation requires additional intellectual efforts and sometimes even re-design of the simulation model. In the given example, to run the animation it is necessary to improve the simulation model, emphasizing the behavior of the ships in the channel. Fig.2 represents the improved action scheme of ships, when only one ship can be in the channel at time. In the improved model, service time has exponential distribution with parameter M_1, where $\dfrac{1}{M_1} = \dfrac{1}{M} - \dfrac{2}{1}$, I is the average time, necessary to go through the channel equal to I (I>0), M is the same as in Fig.1.
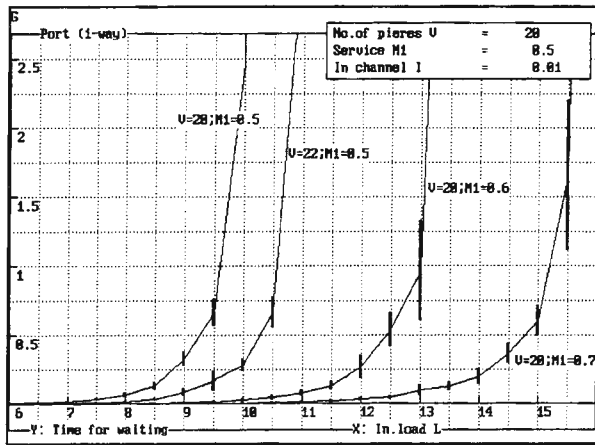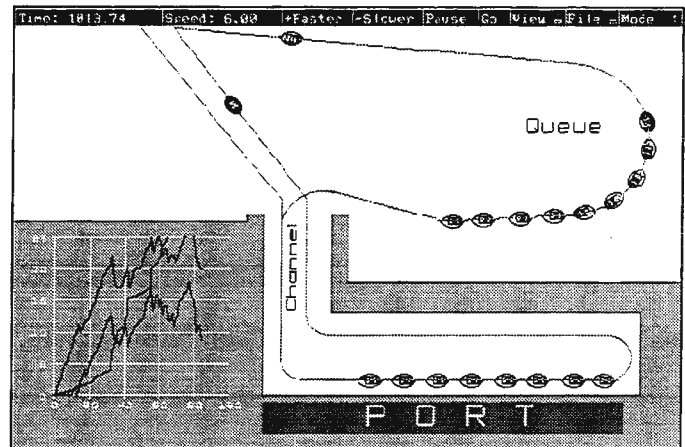

Fig.3. Simulation results

Fig.4. Animation scene of the system by ProofAnimation

Summary

Computer simulation is just one of Computer Based Training (CBT) facilities. When simulation features are combined with tools of different specific tasks, application of computer in training is extended. That allows to integrate different study subjects and to get experience of practical work in various situations.

The use of simulation advances the knowledge in theory and provides possibilities to master their applications to problem-solving. The processing and analysis of data, given in study materials, are not the only choice for a student. Using simulation models, he/she can design and run experiments, and investigate how the changes in system parameters and structure can influence its behavior and results of data processing. Thus, investigating processes in real system, student can profit data processing and analysis techniques.

References

1. Blumel, E. et.al. 1997. Managing and Controlling Growing Harbor Terminals. SCS Europe BVBA, Ghent.
2. Sierenberg, R.W. and de Gans, O.B. 1992. Personal Prosim: A Fully Integrated Simulation Environment. In the *Proceedings of the European Simulation Symposium*, Dresden, pp.167-173.
3. Turin, Y.N. and Makaroff, A.A. 1995. Data Analysis on Computers. Finansy i Statistika, Moscow.
4. Law,A. and Kelton, D. 1991. Simulation Modeling and Analysis. McGraw-Hill, Inc.

ONE MODEL FOR SIMULATION-BASED APPROACH TO COMPUTER SYSTEM PERFORMANCE EVALUATION

S.L. Frenkel

The Institute of Informatics problems (IPIRAN), Russian Academy of Sciences, 30/6 Vavilova, 117900, Moscow, Russia. Fax: (095) 310-70-50 , E-mail: slf@dep34.ipian.msk.ru

1.Introduction

The simulation is the main tool for computer system (CS) performance evaluation because of analytical methods computational complexity. The very important example of the simulation-for-performance evaluation is so-called "workload characterization", when some programs are run on a platform (or its prototype) to simulate its behavior for some specific program class [Raiston83, Saavedra 89]. Workload model is based on the system profiles/traces recorders. In fact, the characterization is a way to represent a computer system behavior both in terms of frequencies of some events concerned with programs execution and in terms of their duration. These events, as a rule, are connected either with the processor's instructions or with some high-level operations [Saavedra89, Gong95]. The frequencies and the times are combined in the product form, and then they are used to range the system prototypes variants in accordance with their performance expected to find the optimal implementation.

This approach is based on the intuitive reasons the frequenter T the frequenter event the stronger its impact on the performance. However, there are, at least, two questions concerning the way of performance evaluation:
(i) the mentioned above measurements are performed for the specific suite of programs (benchmarks [Dong87]; whether it is possible to justify the results for any programs from some programs class specified; (ii) how accuracy of frequencies/times estimation do we need to provide some reliable machine variants separating for the optimization with a performance criterion?
As for (i) it might seem that this is just common philosophical question which, obviously, takes place for any area of simulation ("whether we reflect all problem domain, simulating several patterns only?"), because, in fact, it refers to the common induction justification problem!). However, in reality, this problem is rather practical. The point is that dissemination (in the rigorous manner) of the simulation pattern-based results for the domain modeled is possible in two cases:

- the simulation results use to verify some analytical model ;
- the simulation covers all objects of the area modeled (the fault simulation of the digital circuits [Ben84], as an example).

However, in the traditional approaches to the characterization is even absent the uniquely definition of "performance" itself. In fact, performance is a vector of temporal parameters, which are set of measurements giving the utilization either of the major system resources (and the amount overlap in CPU and I/O utilization) or other times for the separate events in the system (e.g., duration of primitive operations of some high-level language [Saavedra89]). So, on the one hand, the performance is intuitively understood as a time to execute some programs at the given hardware platform, and, on the other hand, there is no any mathematical model to connect the partial parameters with any integral time characteristics. The problem (ii), as can be easy seen, is determined by the same reasons. The possible increasing of the CSs design cost can be pointed out as an immediate effect of these circumstances. The point is that optimization results obtained with benchmarks measuring may be inadequate (from the point of view of time execution) for one or another program from application domain (AD) considered. As a first step to overcome this problem, based on profiling/tracing performance estimation rigorous model should be developed.

This paper represents one from possible approaches to such modeling, showing both the way of justified using of mentioned above characterization data and outlining some experimental results. The approach is mostly intended for CSs, which are designed for specific applications and are defined both by hardware and application-oriented software. First of all, this is various embedded systems (e.g., digital signal processors (DSP)), as well as universal computing system which include emulators to run applications on platforms for which they were not initially

targeted (so-called "software migration problem". [Halfhill94] .

2.Performance Evaluation Model

As mentioned above, the characterization is performed in terms of the sum of the products of the frequencies and the times. Therefore, in order to model the performance evaluation we should understand if these sums reflect some performance measure as a time characteristic for a class of programs with regard to the some application domain (AD).

Let P= $\{(P_1,ID_1), (P_2,ID_2), ,(P_s\ ID_s)\}$ be a sequence of programs P_i i=1,...s with input data suites $\{ID_i\}$ from some AD, $O_1,.. O_m$ are the set of "basic operations" in terms of which the program are represented (that is, some software specification language (e.g,. VHDL, C, or some subroutines mapping the vertices/edges of corresponding program data/flow graph [Gong95])), $\tau_1,.. \tau_m$ are the times of each O_i execution, which are independent random values relatively to P (but they are constants during each program execution). The randomness of-the times (over $\{ P_i \}$ set) is due to cache miss, context switching etc.

Let's consider the sum $\sum f_i *\tau_i$, where M is the number of the basic operations in the sequence P (that is, length of the sequence), i=1,m.

Theorem. Let T be expected time of execution of sequence P.

Then, under above conditions we have for T:

$$T/M \to^P \sum_{i=1}^{M} f_i *\tau_i \qquad (1)$$

where $\to^P$ is the convergence in probability, *Proof.* Obviously, the execution time T_P of P is:

$$T_P = \sum_i k_i * \tau_i = \sum_i t'_i$$

where k_i, i=1,...m- is the number of using of i-th operation during of P execution, $t'_i = k_i * \tau_i$.

Taking into account our above assumption on the τ_i independence (as well as natural assumption on the uniform-limited of τ_i dispersions), we can write in accordance with the law of large numbers [Feller57] :

$$(1/M) \sum_i t_i' \to^P (1/M) \sum_i k_i *\tau_i = \sum_i f_i *\tau_i$$

and, correspondingly, taking into account that $(1/M) \sum t_i' = T_P/M$, and transiting to the T, which is the average value of T_P, we have proved equivalence (1).

Obviously, we can eventually write:

$$T/M \approx \sum f_i *\tau_i \qquad (2)$$

(Because if the law of large numbers is truth, that $\forall \varepsilon >0 \lim_{M \to \infty} Prob(| T_P-T| /M>\varepsilon) \to 0$

((Note, that $\{ f_i *\tau_i \} \to$ T monotonous dependency property is unaffected by assumptions "τ_i are constant during of the P execution").

Let's discuss the theorem result. Let us suppose the programs sequence P is an united program for the input data set $\{ID_i\}$. Then, the sum of the weighted (by the frequencies) basic operation execution times (2) is proportionally with the average time of execution of the some large program from specific AD. Obviously, if this program had some "characteristic" properties relatively programs from AD, T could be considered as a natural performance measure. First of all, to be a characteristic, P must include all semantic elements of AD (e.g., all image processing operation, if AD corresponds to the class of image processing operation).

Furthermore, let's consider AD as a general population, and P as a corresponding statistical sample [Pollard77]. Then, to pretend to be a characteristic program, the size of P should provide the suitable (from the statistical point of view) confidence intervals for all values f_i , τ_i, $f_i *$ τ_i. In this context, T reflects the expected time of program execution over AD, and, therefore, this is relevant characteristic of the performance. By this means, there is the explicit monotone dependence between T and both f_i and τ_i . Such dependence determines the possibility to use the percentage terms $f_i *\tau_i/ \sum f_i *\tau_i$ as a guide to locate performance bottlenecks, knowing the parts of CS hardware/ software system which are responsible for the corresponding operation groups execution. The techniques to use of like data is well-known [Das94, Bashr97]. However, to use well of above percentages to variants ordering, we have to provide $\{ f_i \}$ estimation with a suitable accuracy (in a statistical sense), otherwise it would be wrong results of comparisons of design versions (e.g. if f_i confidence intervals are too broad).

So, mentioned above model (with its interpretation) demonstrates that traditional characterization techniques based both on the frequencies and the times of basic operations can provide rational performance estimations for CS, if we take into account the requirements to programs sample, which, in fact, are like the requirements for testing of system specification [Howden86].

3. Example of the model using

Our experience of using of mentioned above simulation-based performance model regards to RISC system that must execute the x86 Windows NT applications. The using of (f_i, τ_i) data for the performance evaluation to improve the applications execution time at this RISC platform has been examined. This goal had to be basically achieved by the improving of the x86 instructions emulating ways. One of the specific aspects of this investigation was the studying of the possibility to provide above requirements to software measurement, i.e. to provide suitable confidence intervals for the frequencies measured in the so complicate environment as Windows NT.

The conceptual model of this environment is the "abstract machine under given workload" [Raiston83].

We had to solve some methodological problems to separate of system's and applications calls to provide AD statistics collection, as well as the problems of various conditional probabilities definition, etc. These problems have been solved by using both of organization of statistics collection program and of some mathematical decisions. The frequencies of all possible x86 instructions types have been measured by this program (in total for eleven types have been extracted, e.g. vector of addressing forms, vector of prefixes size etc. All these type correspond to mention above AD semantic classes (section 2)). The measurements have been grouped in the vectors, which represent the characteristic of applications migrated in accordance with these types. Each vector includes the number of bits, corresponding to the number of values of the characteristic. For example, the vector *Register*:

{al,cl,dl,bl,ah,ch,dh,bh,ax,cx,dx,bx,sp,bp,si,di,ea,ec ,eb,ed,es,cs,ss,ds,fs,gs}, contains the frequencies of using of corresponding registers under workload considered.

Let's consider briefly the ways of measurement of these frequencies for AD on the background of system's programs. There are two types of system's calls impact on the frequencies estimation. First, this is various dialog boxes, which should not migrate to target CS. The instructions of this component may be excluded from the statistics collected by the stopping of corresponding collection ("Suspend" regime). Secondly, this is influence of various "invisible" functions like Winglon. As showed the analysis, the best way to smooth the such "noise" influence is to combine of statistics, gathered for each of (P_i, ID_i) (Section 2) to calculate above frequencies from

the unified sample. In fact, this is the way to achieve result closely approximating the Bayesian frequencies estimations procedure.

It is easy to see that a natural model for the above frequencies measurement is the polynomial trials [Pollard77]. Correspondingly, the confidence ellipsoid for parameters of the polynomial model can be used as an accuracy characteristic:

$$(m_1 - M^*p_1)^2 / m_1 + (m_2 - M^*p_2)^2 / m_2 + ... + (m_1 - M^*p_N)^2 / m_N \leq \chi^2(N-1,t)$$
$$p_1 + p_2 + ... p_N = 1$$

where N is the number of mentioned above characteristic vector bits, all p_i are the frequencies of these bits in P, $\chi^2(N-1,t)$ is chi-squared criterion with N-1 degrees of freedom, and t is a given significance level (e.g., 5%) [Pollard77], m_i is the measured number each of bit in P (under given $\{ID_i\}$

The results of our investigations show the possibility to estimate above frequencies with a good accuracy (a huge statistical materials there is in [Fr96]). As mentioned above, one from significant requirements to the data obtained in the framework of the model is to be separable to make a decision during variants comparison, that reduced to the requirement to provide suitable confidence intervals for the $f_i \cdot \tau_i / \sum f_i \cdot \tau_i$ (but not for the frequencies and times only). As can be seen from figure 1, the distances between values are very significant for the various x86 Windows NT applications (e.g., between MS-Office and Mathcad ones), that points out the good possibility for the variants separation (see next page).

Mentioned above program has been implemented using of the single-step instruction execution mode (i.e. using the single-step trap flag). Current mnemonics-or-opcodes diagrams are displayed on the screen during the statistics collection and collected values of opcode/mnemonics can be indicated by the mouse click in the corresponding diagram's point. Besides the frequencies, this programs calculates the various histograms of used memory size as well as basic blocks length, branches displacement size, and - statistics of using

PERCENTAGE OF INSTRUCTION EXECUTION MEAN TIME
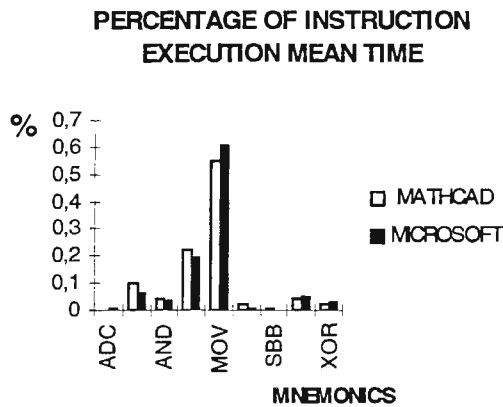
Fig. 1 Diagram of $f_i *\tau_i/ \sum f_i *\tau_i$ data for five upper (in terms of this percentage)   x86 instructions emulated on the RISC platform.

of various resources (memory area, registers etc.).

4. Some results and conclusions

The mathematical model of simulation-based performance evaluation has been suggested and investigated in this work The simulation activity corresponds to design prototyping stage, when the workload of system designed are measured intensively. The analysis of the model demonstrates that traditional characterization techniques of workload measurement based both on the frequencies and the times of basic operations measurements can represent the real performance of CS, if we take into account the requirements to programs sample, which, in fact, are like to requirements for test patterns for testing of system specification [Howden86]. As can be shown, this model is a generic for many of recently suggested software characterization models [Gong95].

Especially, the conclusions are useful for the cases dealing with migration software (including many cases of so-called hardware/software co-design [Gong95]).

To investigate experimentally the possible ways to the software migrated characterization the x86 Windows NT environment have been selected as a prototype of instrumental machine. We relied on it because   on the one hand, this is the multithreaded software (this is very important for the today's high-reliable and high-performance systems), and on the other hand, because the possibility to use well-known tools to support the data collection.

References

[Feller57] W.Feller, An Introduction to Probability Theory and its Applications, John Wiley&Sons, 1957.

[Pollard77] J.H. Pollard, A Handbook of Numerical and Statistical Techniques, Cambridge University Press, 1977.

[Raiston83] Encyclopedia of Computer Sciences and Engineering ,(ed. by A. Raiston), John Wiley&Sons, 1983

[Ben84] R.G. Bennets, Design of Testable Logic Circuits,Addison-Wesley Pub. Company, 1984.

[Howden86] W.E.Howden, Functional Program Testing and Analysis, McGraw-Hill, 1986.

[Dong87] J.J.Dongarra et al, Computer benchmarking: Paths and pitfalls, IEEE Comput. Mag., vol.24, no. 7,pp.38-43, July, 1987.

[Saavedra89] R. H.Saavedra-Barrera et al. Machine characterization Based on an Abstract High-Level Language Machine, in IEEE Trans. On Comp. Vol 38, No 12, December 989, pp. 1659-1679.

[Halfhill94] [Tom R. Halfhill, "Emulation: RISC's Secret Weapon", BYTE, April 1994.

[Das94] D.Das et al, Reducing setup cost by automatic generation of redesign suggestions, In Proc. Of ASME Conference " Computers in Engineering", 1994.

[Bash97] D.Bash and M. Zagar, ATLAS, IEEE Design & Test, July-September, 1997, pp. 104-112

[Gong95] J. Gong et al, Software Estimation Using of Generic-Processor Model, Proc. of EDAC'95, 1995. [Fr96] S.L. Frenkel, D.L.Petrov, x86 Applications Characterization, and x86 Interpreter Optimization, IPIRAN project report, Contract 795-34-è, Moscow, 1996.

On the efficiency of the splitting and roulette approach for finite Markov chains

Viatcheslav B. Melas

St. Petersburg State University
Bibliotechnaya sq. 2, St.Petersbyrg 198904, Russia
viatcheslav.melas@pobox.spbu.ru

Abstract

The paper devotes to the development of the mathematical theory of the splitting and roulette approach suggested in [2]. The problem of estimation of a Markov chain stationary distribution is considered. As the efficiency criterion it is used the ratio of the magnitude of the covariance matrix determinant under optimal branching design to the same quantity for the immediate simulation. It is proved, in particularly, that for the set of transition matrices with a fixed stationary distribution the supremum of the efficiency is achieved in the limit when the matrix tends to an extreme point of the set.

Introduction

The splitting and Russian roulette approach, suggested by von Neumann (see [5]) can be considered as one of the most universal variance reduction techniques. The idea of the approach consists of a parallel simulation of several independent path of the random process corresponding to the behavior of a complex system under investigation. To each path procedures of splitting and roulette are applied. The main problem of the approach is the optimal choice of the number of paths for splitting and probability of cancellation for the roulette procedure. A number of interesting papers (see [4]) devoted to this problem was published. However, a systematic and rather universal mathematical theory of the approach was seemingly first suggested in the book [2]. This theory is based on the introduction of a probability measure which governs the splitting and roulette procedures.

The present paper devotes to the review and the development of this theory for the case of a finite Markov chain simulation.

Branching technique

Let us consider a finite Markov chain with m states $\{0,1,2,...,m-1\}$, a transition matrix $P = \left(p_{ij} \right)_{i,j=0}^{m-1}$ and a stationary distribution $\pi=(\pi_0,...,\pi_{m-1})^T: \pi^T P=\pi$. Assume that for an integer number N all elements of the matrix P^N are positive. The problem is to estimate the vector π by simulation.

As the immediate simulation method we will consider the well known regenerative method. It is known [1] that the quality of the method does not depend on the choice of the regenerative state. Let us take $\{0\}$ as a such state. The method consists of the simulation of several (say L) paths of the chain beginning in $\{0\}$ up to returning to this state.

As the estimation of π_i $(i=0,...m-1)$ the number of hits to the state $\{i\}$ by all paths devided by the number of all steps by all paths is used.

To describe the branching technique which is a formalization of the splitting and roulette approach let us introduce the notion of experimental design.

Let $\beta_0,\beta_1,...,\beta_{m-1}$ are arbitrary fixed positive numbers. When a path transits from a state i to a state j $(i=0,1,...,m-1, \ j=1,2,...,m-1)$ when $\beta_j \geq \beta_i$ we will simulate additionally η paths beginning in the state j where

$$\eta = \begin{cases} \lfloor \beta_j / \beta_i \rfloor & \text{with a probability} \quad 1-\alpha, \\ \lfloor \beta_j / \beta_i \rfloor + 1 & \text{with a probability} \quad \alpha, \end{cases}$$

$\alpha = \beta_j/\beta_i - \lfloor \beta_j/\beta_i \rfloor$ and $\lfloor a \rfloor$ designates the integer path of a. When $\beta_j < \beta_i$ we will cancel the current path with probability β_j/β_i.

As the estimation of π_k $(k=0,1,\dots,m\text{-}1)$ we will take the number of hits into the state k by all paths multiplied by β_k and divided by the sum of all such numbers. We will call experimental design the discrete probability measure $\tau = \{\tau_0,\dots,\tau_{m\text{-}1}\}$, $\tau_i > 0$ $(i=0,\dots,m\text{-}1)$, $\sum \tau_i = 1$, where $\tau_i = \beta_i \pi_i \, / \sum\limits_{k=0}^{m-1} \beta_k \pi_k$.

In [2] it was proved that the estimators are asymptotically $(L \to \infty)$ unbiased and the covariance matrix of the estimators of $(\pi_1,\dots,\pi_{m\text{-}1})$ multiplied by the expected number of all steps by all paths is approximately equal

$$constB(\tau), \qquad B(\tau) = \sum_{k=0}^{m-1} \frac{\pi_k^2}{\tau_k} B_k$$

where

$$B_k = \left(p_{ki}\delta_{ij} - p_{ki}p_{kj} \right)_{i,j=1}^{m-1},$$

$$\delta_{ij} = 1 \ \ for \ \ i = j \qquad and \qquad \delta_{ij} = 0 \qquad for \ i \neq j.$$

The immediate simulation method is the particular case when $\beta_0 = \beta_1 = \dots = \beta_{m\text{-}1}$ that is $\tau = \pi$.

The efficiency criterion

In the optimal regression design theory (see, f. e. [3,6]) as an optimality criterion it is usually used the magnitude of the determinant of the covariance matrix (the square root of this magnitude is proportional to the volume of the tolerance ellipsoid of least squares estimators).

Therefore as an optimality criterion of a design τ we can take the magnitude $det\ B\ (\tau)$. We will call a design τ^* D-optimal if $det\ B\ (\tau^*) = inf\ det\ B\ (\tau)$, where the infimum is taken over all possible experimental designs. In [2] it was shown that for the described above finite Markov chains a unique D-optimal design exists. And an iterative numerical technique for the calculation of D-optimal designs was introduced.

One of possible measures of the branching technique efficiency in compare with the immediate simulation is the magnitude

$$I = \frac{det\ B(\pi)}{det\ B(\tau)}.$$

Let us denote $I = I_P$, $B = B_P$ to show the dependence on the matrix P. Consider the set $\wp = \wp_{(\pi)}$ of all transition matrices P with a fixed stationary distribution π.

Let us consider the quantity

$$I^* = \sup_{P \in \wp} I_P \, .$$

Basic results on the efficiency of the branching technique

Consider the case $m = 2$. Then any matrix $P \in \wp$ can be written in the form

$$P = P_x = \begin{pmatrix} 1 - x & x \\ \dfrac{\pi_0}{\pi_1} & 1 - \dfrac{\pi_0}{\pi_1} x \end{pmatrix}, \quad x \in [0,1].$$

In [2, p. 84] it was shown that in this case

$$I^* = \max\left\{ \lim \frac{\det B_{P_x}(\pi)}{\det B_{P_x}(\tau^*)}, \quad \frac{\det B_{P_1}(\pi)}{\det B_{P_1}(\tau^*)} \right\} =$$

$$= \max\left\{ \frac{2}{\left(\sqrt{\pi_0} + \sqrt{\pi_1}\right)^2}, \quad \min\left\{ \frac{1}{\pi_0}, \frac{1}{\pi_1} \right\} \right\} \leq 2.$$

Thus the upper bound I^* is achieved in the limit under P tending to one of two extreme points of $\wp$.

For $m > 2$ the following result takes place.

Theorem 1. For any $m \geq 2$ the upper bound $\{I_P\}$ is achieved in the limit under P tending to one of extreme points of $\wp$.

The proof is given in the Appendix.

But for $m > 2$ the set of extreme points of $\wp$ does not allow a simple description. Then we will restrict our attention to one class of extreme points of $\wp$. Namely, let us assume without loss of generality that $\pi_0 \leq \pi_1 \leq \dots \leq \pi_{m-1}$ and consider matrices of the form

$$\begin{pmatrix} 1 - x_0 & x_0 & 0 & \dots & 0 & 0 \\ 0 & 1 - x_1 & x_1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{m-1} & 0 & 0 & \dots & 0 & 1 - x_{m-1} \end{pmatrix} \tag{1}$$

where $x_1 = x_0 \dfrac{\pi_0}{\pi_1}, \dots, x_{m-1} = x_0 \dfrac{\pi_0}{\pi_{m-1}}$.

Theorem 2. For the matrices of the type (1) D-optimal design with $x_0 \to 0$ tends to $\left\{ 0\frac{1}{m-1} \frac{1}{m-1} \dots \frac{1}{m-1} \right\}$ and $I_p \to [(m-1)^{m-1} \pi_1 \dots \pi_{m-1}]^{-1}$.

The assertion of the theorem can be verified by a direct calculation. Note, as a corollary, that the efficiency of the branching technique can be arbitrary large. To check this it is enough to consider the case $\pi_1 \to 0$.

Examples

To illustrate the efficiency of the branching technique let us consider a few examples of Markov chains with 3 states. Each of the following transition matrices has the stationary distribution $\pi = (0.0005.\ 0.9985, 0.001)^T$:

$$P_1 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \\ 0.999 & 0.001 & 0 \end{pmatrix}, \qquad P_2 = \begin{pmatrix} 0.3 & 0.4 & 0.3 \\ 0 & 0.9995 & 0.0005 \\ 0.5 & 0.5 & 0 \end{pmatrix}$$

$$P_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.9995 & 0.0005 \\ 0.5 & 0.5 & 0 \end{pmatrix}, \qquad P_4 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0.9995 & 0.0005 \\ 0.5 & 0.5 & 0 \end{pmatrix}.$$

For these matrices $I_{P1} = 164.7$, $I_{P2} = 193.3$, $I_{P3} = 199.2$, $I_{P4} = 248.3$. Note that the efficiency is rather close to the quantity $I^* = (4\ \pi_1\ \pi_2)^{-1} \approx 250$ given by Theorem 2 in spite of that only the matrix P_4 has the type considered in the theorem.

Acknowledgments

References

[1]. Crane, A.M., and D.L. Iglehart. 1974. Simulating stable stochastic systems. 2: Markov chains.. *Journal of Association of Computing Machinery*. 21 (1): 114-123.

[2] Ermakov, S.M., and V.B. Melas. 1995. *Design and analysis of simulation experiments*. Dordrecht, London: Kluwer Academic Publishers.

[3] Fedorov, V.V. 1972. *Optimal experimental design*,Academic press, New York.

[4] Glasserman, P., P. Heidelberger, P. Shahabuddin, and T. Zajic. 1996. Splitting for rare simulation analysis of simple cases. In *Proceedings of the 1996 Winter Simulation Conference*, IEEE: Computer Society Press, 302-308.

[5] Kahn, H. 1956. Use of different Monte Carlo sampling techniques. *Symposium on Monte Carlo methods*, ed. H. A. Meyer. New York: John Wiley and Sons, 146-190.

[6] Kiefer, J. 1974. General equivalence theory for optimum designs. *Annals of Mathemathical Statistics*. 30: 271-294.

Appendix

The proof of Theorem 1.

Let us suppose that the assertion of the theorem does not take place. Then there exists a matrix P^* which is not an extreme points of $\wp$ and such that

$$\inf \det\ B*\left(\tau\right)/\det B*\left(\pi\right) = \inf_P \inf_P\ (\det B_P(\tau)/\det B_P(\pi)),$$

where $B^*(\tau) = B_{P^*}(\tau)$. Consider matrices of the form $\tilde{P} = \tilde{P}(x,\alpha) = (1-\alpha)P^* + \alpha P(x)$, where

$$P(x) = \left(P(x)_{kl} \right)_{k,l=0}^{m-1}, \qquad P(x)_{ji} = 1 - x, \qquad P(x)_{ii} = x,$$

$$P(x)_{ij} = \frac{\pi_i}{\pi_j} x, \qquad P(x)_{jj} = 1 - \frac{\pi_i}{\pi_j} x,$$

$$P(x)_{il} = 0, \qquad l \neq i, j, \qquad P(x)_{kj} = 0, \qquad k \neq i, j,$$

$$P(x)_{kl} = P_{kl}^*, \qquad k, l \neq i, j,$$

for some fixed $i \neq j$, $\quad i, j \in \{0, 1, \ldots, m-1\}$

By the definition of $P*$ the following inequality should take place

$$\lim_{x \to 0} \frac{\partial}{\partial \alpha} \left[\det B_P(\tau^*) \,/\, \det B_P(\pi) \right] \Big|_{\alpha = +0} \geq 0,$$

where τ^* is the D-optimal design for matrix $\; \tilde{P} = \tilde{P}(\alpha, x)$. Calculating the left side we obtain

$$trB^{*-1}(\tau^*)\left(-B_i^* \frac{\pi_i^2}{\tau_i} - B_j^* \frac{\pi_j^2}{\tau_j} \right) \geq trB^{*-1}(\pi)\left(-B_i^* \pi_i - B_j^* \pi_j \right),$$

where B_i^*, B_j^* are elements of the representation of $B^*(\tau)$ in the form $B*(\tau) = \sum \dfrac{\pi_i^2}{\tau_i} B_i^*$. $\quad \tau *$

— D-optimal design for $P*$.

By theorem 3.7.1 from [2]

$$m\tau_k^* = \pi_k^2 trB^{*-1}(\tau^*)B_k^*, \qquad k = 0, 1, \ldots, m-1.$$

Therefore we have

$$m(\tau_i^* + \tau_j^*) \leq trB^{*-1}(\pi)\left(B_i^* \pi_i + B_j^* \pi_j \right)$$

for any $i \neq j$, $i, j = 0, 1, \ldots, m-1$.

Let us sum both sides of this inequality by $i, j = 0, 1, \ldots, m-1$, $i \neq j$. The sums of both sides are equal to $2m(m-1)$. Therefore for any $i = 0, 1, \ldots, m-1$ we have

$$m\, \tau_i^* = tr\, B^{*-1}(\pi)\, B_i^* \, \pi_i.$$

Dividing both sides by τ_i^*/π_i and summing by $i = 0, 1, \ldots, m-1$ we obtain

$$m = tr\, B^{*-1}(\pi)\, B^*(\tau^*).$$

But according to the theorem cited above we have

$$tr\, B^{*-1}(\tau)\, B^*(\tau^*) < m$$

for any $\tau \neq \tau^*$. With $\tau = \pi$ the last inequality contradicts to (2). This contradiction proves the theorem.

METHODS, ALGORITHMS AND SOFTWARE TOOLS FOR CAE OF STOCHASTIC CONTROL SYSTEMS

Sinitsyn I.N., Korepanov Ed.R., Shin V.I.

Institute for Informatics Problems, Russian Academy of Sciences,
Russia, Moscow, GSP-1, str. Vavilova 30/6
Tel: (095) 332-4933/ Fax: (095) 310-7050. E-mail: sinitsin@dol.ru

Abstract

The paper is devoted to analytical, statistical and combined analytical and statistical simulation methods, algorithms and software tools for stochastic control systems (SCS) CAE technologies (analysis and modelling, filtering and optimization, identification, calibration, experiments planning and testing). The SCS model is taken in the form of non-linear Ito stochastic differential equations. Software tools "StS-Analysis", "StS-Filter", "StS- Control" and their applications are considered.

1. Introduction

Stochastic Control Systems (SCS) serve as a suitable mathematical model in modern statistical control engineering. SCS simulation is one of the most efficient tools of information technologies for real control system functioning under random disturbances during all stages of design, testing, manufacturing and development. Following [1], we state that there several reasons why SCS computing simulation has become now more important: (1) modern information technologies of SCS problems solving and corresponding hardware and software tools are so flexible, intellectually powerful and cheap that SCS simulation can be more widely used; (2) for problems of detailed statistical analysis and synthesis the need of simulation especially for multidimensional, large-scale and man-machine SCS is increasing; (3) in many cases it is economically not feasible or not possible to experiment on system itself once or even many times to receive representative statistical data base.

The simulation, analysis, optimization (filtering and control) and corresponding software tools are the basic components of modern CAE. The structure of system, optimal values of its parameters satisfying the design requirements or minimizing the given qualitative characteristics are chosen while using these tools. The using of up-to-date simulation, analysis, filtering and control methods makes possible research processes in control systems, take into account restrictions on parameters and combine of control system design and simulation processes with testing of control algorithms in on-line operating conditions.

2. SCS Analysis. Analytical Simulation Methods

Consider SCS determined by Ito stochastic non-linear differential equation [2]:

$$dX = a(X,U,t)dt + b(X,U,t)dW, \quad X_0 = X(t_0), \quad t \geq t_0, \tag{1}$$

where $X \in R^n$ being a state vector; $U \in D$ is a control vector, $D \in R^d$ is some compact set., .. _ R^r is a Wiener vector process, $E(dW)=0$, $E(dWdW^T)=vdt$, v being the intensity of W; E is mathematical expectation; $a=a(x,u,t)$ and $b=b(x,u,t)$ are known functions, $a: R^{n+d+1} \to R^n$, $b:$

$R^{n+d+1} \to R^{nr}$. Initial state X_0 represents a random vector with a priori known distribution $p(x_0)$.

The analysis problem of SCS (1) consists in determining the probability density $p(x,t)$ of state vector or its moments

$$\tag{2}$$

The survey of up-to-date methods of non-linear SCS analysis is given in [2]. Let us consider two approximate analytical methods: a normal approximation method (NAM) and quasi-moments method (QMM) as basic analytical simulation methods (ASM).

NAM is based on the approximation of an unknown density $p(x,t)$ by normal one:

$$p(x,t) \approx N(m,K) = [(2\pi)n|K|]-1/2 \exp[-(x-m)TK-1(x-m)/2], \tag{3}$$

where $m=EX(t)$ being the expectation, $K=cov\{X(t),X(t)\}$ being the covariance matrix, $|K|=\det(K)$. The parameters $m=m(t)$ and $K=K(t)$ are determined by the following ordinary differential equations:

$$\dot{m} = E[a], \quad \dot{K} = E[(X-m)a^T + a(X-m)^T + \sigma], \quad t \ge t_0,$$
$$m_0 = m(t_0), \quad K_0 = K(t_0), \tag{4}$$

$$a = a(X,U,t), \quad b = b(X,U,t), \quad \sigma = b v b^T.$$

All expectations in the right-hand sides of Eqs (4) are calculated for a normal density (3),

$$E[g] = \int g(x,u,t)N(m,K)dx, \quad g = a \quad or \quad g = (X-m)a^T + a(X-m)^T + \sigma \tag{5}$$

The total number of Eqs (4) for $m=[m_1...m_n]^T$ and $K=K^T=[K_{ij}]$ ($m_i=Ex_i$, $K_{ij}= cov\{X_i X_j\}$) is equal to $Q_{NAM}=n(n+3)/2$. In particular, for $n=10$ and 100 we have $Q_{NAM}=65$ and 5150, respectively.

In particular case when the function b in Eq (1) is independent of the state vector X, $b=b(U,t)$, NAM gives the known equations for m and K based on statistical linearization of non-linear function $a(X,U,t)$ in Eq (1) [2,3].

QMM is based on more accurate than NAM approximation of unknown density by Hermite polynomial expansion [2]:

$$p(x,t) \approx p^*(x,t) = N(m,K)[1 + \sum_{k=3}^{L} \sum_{|i=k|} c_i(t)h_i(x-m)], \tag{6}$$

where $i=[i_1...i_n]$ is a multiindex, c_i is a quasi-moment; $h_i(x) = H_i(x)/(i_1!...i_n!)$, $H_i(x)$ is Hermite polynomial. Quasi-moments c_i ($i=1,...,L$) are determined by corresponding set of ordinary differential equations similar to Eqs (4),(5) [2]. The total number of Eqs for quasi-moments c_i ($i=[i_1...i_n]$, $i_1,...,i_n =1,...,N$) is equal to $Q_{QMM}= (n+N)!/(n!N!)-1$.

In particular, for $n=10$ and 100, $N=6$ we have $Q_{QMM}=8007$ and 1.7×10^9, respectively. So for different SCS classes approximate reduction methods for ASM were developed (ellipsoidal approximation modified moment-and-semiinvariant methods etc [2]) At practice the corresponding ASM are designed for concrete systems.

3. SCS Analysis. Statistical Simulation Methods

Statistical simulation methods (SSM) are based on many times simulation of SCS while using numerical solution algorithms of Eq (1). Let us give some basic formulae for approximate calculating the moments α_i (2) of the state vector $X=X(t)$,

$$\alpha_i(t_k) \approx (1/M)\sum_{j=1}^{M}[X_1^{(j)}(t_k)^{i_1}]K[X_n^{(j)}(t_k)^{i_n}], \tag{7}$$

where t_k $(k=1,2,...)$ is a discrete time moment; $X_1^{(j)}$ $(t_k),...,X_n^{(j)}$ (t_k) are values of the components X_1 $(t_k),...,X_n$ (t_k), corresponding to jth realization; j being a number of realization; M is a total number of the realizations. The values $X^{(j)}$ $(t_k)=[$ $X_1^{(j)}$ $(t_k)...X_n^{(j)}$ $(t_k)]^T$ for each (jth) realization are given by numerically solution of Eq (1).

The known modified Euler method, Milshtien method, Rozenbrock method and Runge-Kutta-Felberg methods for numerical integration of Ito stochastic differential Eq (1) are the basic SSM for different applications [6-11].

We also develop combined analytical and statistical simulation methods (CASM) for simulation of special classes of SCS with complex and quaternion right-hand sides of Eq (1) presented in analytical and table forms [8].

4. Conditionally Optimal Filtering and Conrol

Let us consider control problem determined by the following Eqs:

$$dX = a(X,U,t)dt + b(X,U,t)dW, \quad Y = a_1(X,t) + b_1(t)V, \quad t \ge t_0, \tag{8}$$

where $X \in R^n$ is a state vector; $Y \in R^m$ is an observation vector, $U \in R^d$ is a control vector; $W \in R^r$ is a Wiener vector process; $V \in R^m$ is a Gaussian white noise.

There are many papers devoted to the approximate solution of the control problem (8) (see, [12-14]). We dwell on methods of finding approximate controls which realize in real-time incoming current observations . These methods are based on conditionally optimal estimation and control theory. According to this theory the control U is taken from some class of admissible controls which is determined by the differential equation of a given structure [15,16]. In [15] admissible control depends on observations $U_t=U(Y_t,t)$ and is taken in the following form:

$$U_t = \delta(t)\zeta(Y_t,t) + \rho(t). \tag{9}$$

In [16] admissible control $U_t = U(\hat{X}_t,t)$ depends on the estimate $\hat{X} = \hat{X}_t$ defined by Eqs of the given structure:

$$U = \delta(t)\zeta(\hat{X}_t,t) + \rho(t), \tag{10}$$

$$d\hat{X}_t/dt = \alpha(t)\xi(\hat{X}_t,t) + \beta(t)\eta(\hat{X}_t,t)Y + \gamma(t), \tag{11}$$

In Eqs (9)-(11) $\xi(\hat{X}_t,t)$, $\eta(\hat{X}_t,t)$ and $\zeta(\hat{X}_t,t)$ are given non-linear functions determining structures of estimates and controls $\hat{X}$, respectively. Unknown structure filter gains and $\alpha=\alpha(t)$, $\beta=\beta(t)$, $\gamma=\gamma(t)$ control structure gains $\delta=\delta(t)$, $\rho=\rho(t)$ are determined from the following statistical local criteria

$$I_t = E[(\hat{X}_t - X_t)^T (\hat{X}_t - X_t)] \to \min \quad and \quad J_t = E[F(U_t, X_t, \hat{X}_t)] \to \min,$$

respectively. As it is known from [2,15,16] the structure gains α, β, γ, δ and ρ depend on expectations $E[\Phi(X_t, \hat{X}_t,t)]$ ($\Phi(X_t, \hat{X}_t,t)$ being known function of composite random vector $[X_t^T, \hat{X}_t^T]^T$). Therefore ASM, SSM, CASM are used for finding the expectations $E[\Phi(X_t, \hat{X}_t,t)]$ and the structure gains. The conditionally optimal control (Eqs (9),(10)) and

filter (Eq (11)) are on-line control and on-line filter, respectively, as the gains α, β, γ, δ and ρ depend only on time t and therefore they can be computed before control and filter design.

5. Algorithms and Software Tools

The PC-based software tools "StS-Analysis" [2,4] for polynomial non-linear functions a and *b* in Eq (1) uses ASM described by Eqs (3), (4). The "StS-Analysis" sets up and numerically solves multidimensional non-linear ordinary differential equations (4) by means of Runge – Kutta methods of *s*th order (*s=1,2,3,4*).

The PC-based software tools "StS-Filter" and "StS-Control" were developed for on-line filter (11) and on-line control (10) design [5,17,18].

The software tools "StS-Filter" and "StS-Control" were assigned for solving simulation, analysis, filtering and control problems. These tools are oriented to the analysis, signal processing and control problems of multidimensional SCS. Using these tools end user receive full solution of control problem during some hours. The results of solution have a good documentation's: listings, texts, and graphics. Dialogue interface of these tools allows the designer to solve the problems for himself without help of the programmers. We use these tools as functional module of large program complexes while designing concrete control systems. They permit to increase essentially the efficiency of control systems analysis and design.

Now we are preparing efficient ASM, SSM, CASM, algorithms and software tools for CAE of high dimensional non-linear SCS using Digital workstations with Alpha processors.

6. Conclusion

Modern applied SCS theory gives the real possibility to develop very efficient sequential and parallel methods for simulation, on-line filtering and control problems.

The corresponding software tools for PC and workstations are preferable for different stages of CAE activities.

The developed software tools were implemented for guidance and navigation transport systems, vibromechanical systems control and in financial informatics.

The work was supported by Russian Fond of Fundamental Research (Project 95-01-00426)

REFFERENCES

1. L.Dekker. Preface. Simulation Practice and Theory. 1993, No. 1, 1-3.

2. Pugachev V.S., Sinitsyn I.N. Stochastic Differential Systems. Analysis and Filtering. 1st ed., Moscow, Nauka, 1985, 2nd ed., 1990; John Wiley&Sons. N.Y. 1987.

3. Sinitsyn I.N. Methods of statistical linearization (Survey). Automation and Remote Control, 1974, v. 35, No. 5, 765-776.

4. Pugachev,V.S., I.N.Sinitsyn, A.A.Cherednichenko, V.I.Shin and V.I.Sinitsyn. Mathematical software for analysis of multidimensional non-linear stochastic systems. Automation and Remote Control, 1991, v.52, No. 1, 73-80.

5. Sinitsyn I.N. Lectures on PC-based Nonlinear Stochastic Mechanical Systems Research, Praha, 1992, Ustav Thermomechaniky CAV.

6. I.N.Sinitsyn, A.P.Karpenko. Combined parallel statistical and analytical modelling methods, algorithms and software for dynamical stochastic systems research/ L.Dekker W.Smit, J.C.Zuidervvaart (Eds). Proc. EUROSIM 1996 Intern. Conf. "HPCN

Challenges in Telecompand Telecom: Parallel Simulation of Complex Systems and Large Scale Applications", 10-12 June 1996, Delft, The Netherlands. ELSEVIER. Publishers B.V. 1996. Amsterdam-Lausanne-New York, 187-194.

7. Sinitsyn I.N. Parallel simulation technologies for stochastic systems research. Proceedings of 4th International Conference "Parallel Computing Technologies (PACT97), Jaraslavl, 8-12 September, Springer Verlag LNCS, 383-388.

8.Sinitsyn I.N. Parallel methods in non-linear statistical dynamics. Proceedings of "EUROMECH": 2nd European Non-linear Oscillations Conference, 1996 Publishing House of Czech Technical University. V.1, 407-410.

9. Kloden P.E. and E. Platen, The Numerical Solution of Stochastic Differential Equations, Berlin-Heidelberg-New-York, Springer,1992.

10. Artemiev S.S., Yakunin M.A., Mikhailichenko I.G. and I.O. Shkurko. Dynamic and Control (in Russian). Novosibirsk. 1995.

11. Artemiev S.S., Mikhailichenko I.G. and I.N. Sinitsyn. Statistical Simulation of Risk and Forward Financial Transactions (in Russian). Publishing House of Novosibirsk Computer Center Syberian Division of Russian Academy of Sciences. Novosibirsk. 1997.

12. Bryson A.E., Yu-Chi Ho. Applied Optimal Control. Optimization, Estimation and Control. Massachusetts. Toronto. London, 1969.

13. Sage A.P., White C.C. Optimum Systems Control. 2nd ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

14. Naumov B.N. (Ed.) Nonlinear Control Systems: Philosophy of Nonlinear Control Systems. CRC Press, Inc., 2000 Corporate Blvd. N.W., Boca Raton, Florida, 1990.

15. Pugachev V.C. Condilionally optimal control (in Russian). Doklady Akademii Nauk, 1993, v.330, No.6, 710-712.

16. Kazakov I.E. Analytical design of conditionally optimal control in nonlinear stochastic systems by complex local criterion. Automation and Remote Control, 1995, v.56, No.3.

17. Korepanov Ed.R., V.S.Pugachev, I.N.Sinitsyn, A.P. Khatunsev, V.I.Shin and V.I.Sinitsyn. Mathematical software for desiging conditionally optimal filter and analysis processes in discrete stochastic systems. Automation and Remote Control, 1992, v.53, No.6, 851-857.

18. Sinitsyn I.N. Ill-posed problems of on-line conditionally optimal filtering. Proceedings of the Intern. Conf. Ill-posed Problems in Natural Sciences, Moscow, August 19-25. 1991/Eds. A.N.Tikhonov et al.- Utrecht. VSP/Moscow: TVP Sci Publ, 174-176.

An Hybrid approach to Performance Prediction through Integration of Static Analysis and Simulation

R. Aversa[1], B. Di Martino[1], A. Mazzeo[1], N. Mazzocca[2], U. Villano[3]

[1]Dipartimento di Ingegneria dell' Informazione
Second University of Naples - Italy

[2]Dipartimento di Informatica e Sistemistica
University "Federico II" of Naples - Italy

[3] IRSIP - CNR - Naples - Italy

{aversa,dimartin,mazzeo,mazzocca,villano}@grid.unina.it

Abstract

In this paper we present an approach for performance prediction of irregular and data-dependent message passing programs. The approach is based on integration of static analysis and simulation techniques. In particular a solution is proposed for the problem of statically determining all possible interaction patterns among the processes of message passing programs.

This static analysis technique has been integrated with MPSS [2], a tool for simulation of PVM message passing programs, in Heterogeneous and Network Computing environments, to provide for a complete program characterization in terms of idle-, cpu-, communication and synchronization time.

1   Introduction

With increasing the complexity of parallel and distributed architectures, the demand is growing for more sophisticated and cost effective performance analysis techniques and environments [9]. Performance analysis and parallel software characterization and validation is particularly demanding when targetted towards heterogeneuos and loosely coupled distributed architectures, which are nevertheless gaining more and more importance due to the explosion of the network computing area.

Increasing interest is gaining the approach based on *Performance Prediction* [6], due to the advantages it gains over dynamic evaluation techniques for not needing program execution on the target machine. Beyond the straightforward benefits of allowing performance evaluation when the target architecture is not available, those techniques can enable the evaluation of partially implemented program design (prototyping analysis), and scalability analysis.

Performance prediction is usually based on static analysis or simulation techniques [6, 3].

The main disadvantage with static analysis techniques is the need for worst-case assumptions, arising from the unability of such techniques to take into account irregular and data-dependent applications. Even numerical and regular algorithms, such as matrix-matrix product or fast Fourier Transform, may result into parallel programs with irregular and data-dependent behavior, if typical issues arising with heterogenous and loosely coupled architectures are to be taken into account, like workload balancing or message vectorization [8]. whose introduction produces anomalies and deadlocks detection which, although could occur as matter of principle, cannot be encountered in practice ("unreachable paths").

Simulation techniques, which are able to overcome the limitations of the strictly static approach, present on the other hand the disadvantage of a complexity for the produced representations which exponentially grows with the data and code size of the program to be analyzed.

We have designed an approach, and developed a prototypic implementation, based on integration of static analysis and simulation techniques, which provides a complete program characterization in terms of idle-, cpu-, communication and synchronization time for each composing process. The target

language paradigm is message passing, namely C or Fortran with insertion of PVM [7] message passing primitives. We can currently handle SPMD programs, but without any restriction on their execution model.

The approach we are depicting integrates in a coherent way static analysis and simulation techniques in one framework, which is able to determine all possible interaction patterns among processes, remaining cost effective at the same time.

Our approach can statically evaluate parallel code which is data-dependent with respect to number of operations performed. It relies on a probabilistic technique which obtains an estimate on the number of single-node operations to be performed, when the non-determinism is introduced by conditional structures or loop structures whose exit-conditions cannot be evaluated at compile time. A new static analysis technique is introduced for the determination of the possible *interaction patterns* among the processes: it is based on linear techniques for the solution of diophantine sets of unequalities. It provides for a relevant cost effectiveness, but it introduces the restriction that the communication end-points (i.e. the identity of sender and receiver) must be determined at compile-time.

Such a static analysis is able to yield to a representation which is fed to a simulation tool (MPSS), in order to obtain sufficiently accurate performance estimates of its execution on heterogeneous architectures. MPSS can produce performance predictions of the program execution on different target system configurations that can vary in terms of the processing speed of each computing node and of the task-processor mapping.

2 Overview of the hybrid approach

The static analysis phase consists of three steps:

1. generation of a set of partial *sequential event sub-graphs* for each of the processes composing the program under analysis;

2. execution time prediction for the sequential portions of code belonging to each process;

3. generation of an *interaction graph*, which represent the interaction schemes among the composing processes.

The above described phase produces an intermediate representation which represents the parallel program behaviour, in terms of computational and communication events, in a way which is independent from the parallel target system. This representation is then evaluated by the Message Passing System Simulator (MPSS) [2], which takes into account the characteristics of the particular architecture chosen, models the static mapping processes/processors, the heterogeneity of the nodes, the interconnection topology and network load. In addiction MPSS can be used like an analysis tool since it allows the collection of aggregate and analytical indexes related to the heterogeneous system performance (e.g., efficiency, throughput, response time, individual processor utilization), or traces which can be processed off-line by a variety of tools for performance visualization and analysis (e.g., ParaGraph)

The coupling of the technique we are going to describe for nondeterminism resolution with the MPSS system allow us to obtain a full and effective simulation of irregular and data dependent applications, without the need of pre-execution of the program in object.

3 The Static Analysis Phase

The first step generates a set of *process event subgraphs*, one for each of the composing processes. Each process subgraph is a reduced control flow graph for the process code, composed of three different kind of nodes, each of them representing a program *event*:

- nodes representing blocks of sequential statements, not including communication primitives. The execution of these sequential blocks represent a program's *sequential event*. The main information associated with these events is the estimated execution time.

- Nodes representing communication statements. They represent the *communication events* of the program; main information associated with them are the communication partner, the message size and the variables whose values are exchanged.

- Nodes representing control flow branches and loops, whose contexts contain communication primitives. The explicit representation of the execution of these statements in terms of events allow us to keep track of all the communication patterns the parallel program execution can present at run-time.

Nodes representing sequential events are then annotated with an estimate of their execution time. We depict in the following the approach taken.

Given a sequential code excerpt, its execution time is estimated as the sum of the execution times for each of its statements. The single statement's execution time is expressed in terms of parameters which model the characteristics of the target system compiler-architecture. Such parameters represent the execution times needed to perform atomic operations such as arithmetic operations, load and store, control statements' overhead, pattern accesses for arrays etc. Cache effects are also taken into account.

Non-deterministic behavior for the sequential code, arising in presence of data-dependent conditional constructs and iterative constructs, is handled with a statistic technique. In particular: execution time for conditional constructs is estimated as a weighted mean of execution times of their branches (which are assumed by default as equiprobable); number of iterations for iterative constructs is assumed (by default) as half the maximum iteration number (if this latter can be determined statically). It is a priori defined otherwise. All default values can be easily modified by the user.

The annotated process subgraphs are then connected, in order to generate the interaction graph. The connection is realized by representing, with *communication edges*, the "rendez-vous" between communication primitives. Communication primitives at the two endpoint processes (which are restricted to be statically determined) are connected ("match") if they belong to equivalent control flow contexts, i.e. they falls into equivalent control flow paths within the event subgraphs of the two endpoint processes.

In the following the static analysis technique we designed for the determination of the possible *interaction patterns* among the processes is described. It is based on linear techniques for the solution of diophantine sets of unequalities. The technique is at the moment limited to SPMD programs, but without any restriction on their execution model.

For any S and R communication statements in a $d-$nested loop of an SPMD message passing program, s.t.:

- S: send statement (e.g. `send(PROCS(`$f(\vec{I},p)$`),...)`) (PROCS: array of processor id's (1D for sake of easiness))

- R: receive statement (e.g. `receive(PROCS(`$g(\vec{I},p)$`),...)`)

- $\vec{I} = (I_1, \cdots, I_d)$, and f, g linear expressions of $I_1, \cdots, I_d$ and p, with p representing any variable having the semantic of the task number of the process (e.g. `my_tid`)

the condition for having a communication between the process p (sender) and p' (receiver) at the iterations $\vec{I}$ (for the sender) and $\vec{I'}$ (for the receiver) is:

$$\begin{cases} L_i \leq I_i \leq U_i & \forall i \in (1, \cdots, d) \\ L_i \leq I_i' \leq U_i & \forall i \in (1, \cdots, d) \\ L_p \leq p \leq U_p \\ L_p \leq p' \leq U_p \\ p \neq p' \\ f(I_1, \cdots, I_d, p) = p' \quad \&\& \quad context_f(I_1, \cdots, I_d, p) = true \\ g(I_1', \cdots, I_d', p') = p \quad \&\& \quad context_g(I_1', \cdots, I_d', p') = true \end{cases} \qquad (1)$$

$context_f, context_g$ are the control contexts of the statements to which f, g belong. Please note that the loop bounds L_i, U_i can be linear functions of the indices of the more external loops.

Thus, finding the communications which occur in that loop nest means to find the set of values

$$(I_1, \cdots, I_d, I_1', \cdots, I_d', p, p')$$

which satisfy the system of diofantine equations 1.

5 This step determines one particular communication graph for the execution.

4 The Simulation phase

On the basis of the program interaction graph obtained we can feed MPSS for the simulation, through a complete trace file or an equivalent higher-level representation of the application virtual execution.

Some further details on the architecture and use of MPSS as a performance prediction and performance analysis tool are now in order. This is a tool for the performance prediction of heterogeneous computing environments (most typically, networks of workstations) programmed in any language based on a message-passing paradigm. The current implementation of MPSS is modeled on PVM and relies upon the existence of a daemon process but this limitation can be easily removed. We are working in order to have MPSS support some more message- passing programming languages, such as MPI and Express. Although MPSS could also be suitable for the performance analysis of Massively Parallel Processor systems (MPPs), its structure is optimized for relatively loosely-coupled computing environments. When the task granularity is fine and communication networks are characterized by very high data transfer rates, such as in MPPs, MPSS may be relatively inaccurate.

Eventually, MPSS is an integrated software package containing everything one might need to set up, run and visualize a simulation session. Its kernel is a trace-driven simulator, which inputs an application trace file and outputs the trace file that the same application would produce if it were run on a different hardware configuration. The trace can be produced by executing the parallel application on any computing environment, not necessarily on the actual target [2]. It is thus possible to exploit systems less powerful than the target one, or even a single- processor machine, e.g., the workstation used for program development. After that, the use of simulation techniques makes it possible to derive information on the performance for several different (real or hypothetical) hardware configurations from the trace collected in the test run.

All task interactions through the run-time support of the programming environment are simulated in great detail using a simulation kernel consisting of three distinct levels: User level, Message- passing System level, Network level. The objects implementing the three levels are sufficiently general to be adapted (through a suitable setup phase) to the simulation of most available hardware and of the majority of message-passing programming environments. An exception is the network level, which is highly dependent on the communication medium and on the access procedures to communication services. Currently, the only supported networks are Ethernet and Fast Ethernet.

Interfaces between different levels are defined in terms of the events they exchange. Network contention is detected, also taking into account the effect of external network load, which is characterized by the injection of a suitable statistical load.

Code written in any message-passing programming language may be analyzed on top of MPSS. The behavior of non-deterministic applications cannot however be simulated, because the simulation will

consider a single path through the program, the one followed in the traced run.

The procedure described above for sequential execution time prediction, interaction patterns detection and non-determinism resolution has thus been designed as a preprocessing phase for MPSS, in order to eliminate the need for a pre-execution of the code to be analized, and to take into account data-dependent codes.

It's worth while emphasizing that this preprocessing phase and in particular the interactions patterns detection can be a valuable tool to reduce the computation complexity of the simulation, that remains one of the major drawbacks of the simulative approach to the performance evaluation. In fact, the static analyzer determining the interaction patterns of a distributed application can be able to recognize those patterns inside an iteration loop whose communication end-points are independent from the iteration count number. The simulated execution of such patterns, under certain conditions, can be performed just once by MPSS without a substantial loss of the simulation accuracy.

5   Example

In this section we present some preliminary results of the application of the described technique to a quite complex numerical algorithm. The algorithm we consider is the *Successive Over Relaxation* (SOR) method for iterative solution of systems of linear equations.

The parallel implementation we are considering (described in [5]) preserves the update ordering of the elements through a particular work distribution strategy. After updating its own elements, each processor computes partial sums for updating elements assigned to subsequent and preceding processors disposed in a pipeline fashion, using elements assigned to it only. As soon as these partial sums are computed, they are sent to the processors that will need them, so that these processors can start a new update step. This step now requires, for each element to be updated, only the computation of the partial sum of the elements assigned to that processor, and a final sum of the partial results received from the other stages of the pipeline.
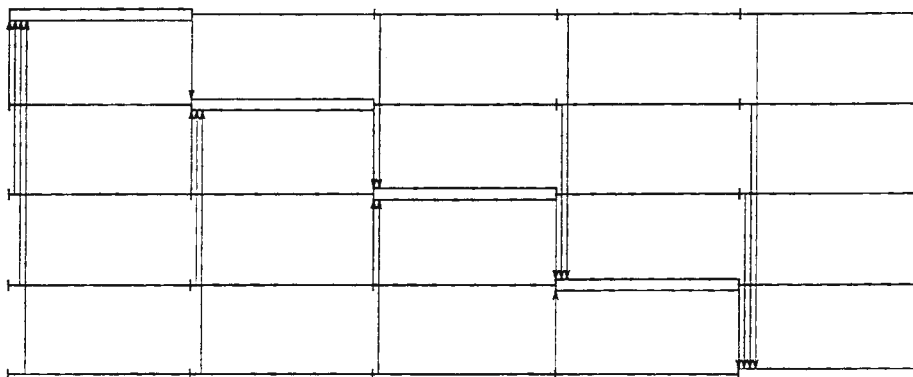

Figure 1: Gantt diagrams of the execution scheme in the ideal hypotesis of zero latency and communication overhead.

Figure 1 shows the Gantt diagrams of the execution strategy just described. The white rectangles represent the update of the elements assigned to each processor, whereas the thin horizontal lines represent the computation of partial sums to be sent to other processors. Arrows represent communications of partial sums. To avoid too many symbols for each iteration, we have only reported one arrow to represent all communications between each pair of processors. The arrow is then placed at the beginning of the segment representing the computation of a group of partial sums to mean that such sums are communicated individually as soon as their evaluation has been completed (i.e. no message vectorization is performed). Finally, communications are assumed to be ideal, i.e. without overhead (the length of segments representing computations is not modified by the presence of communications)

and with zero latency (arrows are drawn vertically). In the more realistic assumptions that communications have an associated overhead and a non-zero latency, communications of partial results should be aggregated (*message vectorization*) to reduce communication overhead. This introduces an idle time for processors. This realistic behaviour is shown in figure 2, which shows the Gantt diagram of algorithm execution on five processors and assuming full vectorization of messages. As requested by the algorithm, the i-th iteration starts as soon as partial sums sent by all other processors have been received. The behavior shown in figure 2 is quite general and independent of the number of processors used.
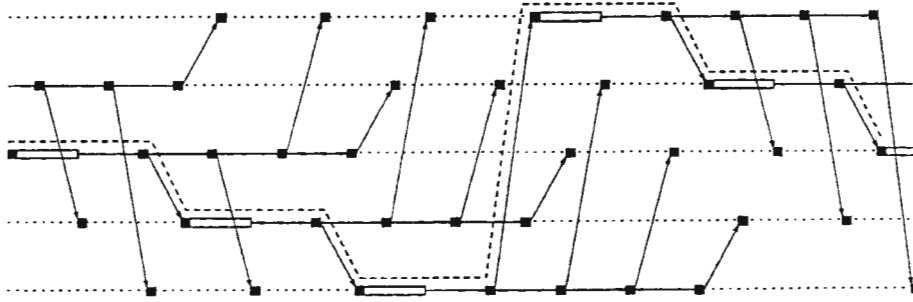
Figure 2: Gantt diagram of the algorithm in the case of non-zero latency and communication overhead.

The two figures show the complex communication pattern which characterizes the parallel algorithm, composed by a gather communication and a corresponding series of scatter communications for each process, in each iteration. This pattern is exactly detected by our approach, after static analysis and simbolic evaluation of the implementing message passing program, developed in Fortran under PVM by following the SPMD (Single Program Multiple Data [4]) paradigm.

A pseudo code follows:

```
call pvmfmytid(mytid)
call pvmfparent(parenttid)
if (parenttid.eq.PvmNoParent) then
  tids(1) = mytid
  m = 1
  call pvmfspawn("sor",PVMDEFAULT," ",nprocs,
    tids(2:nprocs),rcode)
  do i = 2, nprocs
    call pvmfinitsend(PvmDefault,bufid)
    call pvmfpack(integer,i,1,...)
    call pvmfpack(integer,tids,nprocs,...)
    call pvmfsend(tids(i),any,info)
  enddo else
  call pvmfrecv(parenttid,any)
  call pvmfunpack(integer,m,1,...)
  call pvmfunpack(integer,tids,nprocs,...)
endif

do k = 1,niter
  do l = 1, nprocs
    if (l.ne.m) then
      work for the others
      call pvmfinitsend(PvmDefault,bufid)
      call pvmfpack(real4,contribute,...)
      call pvmfsend(tids(l),any,info)
    endif
    if (l.eq.m) then
```
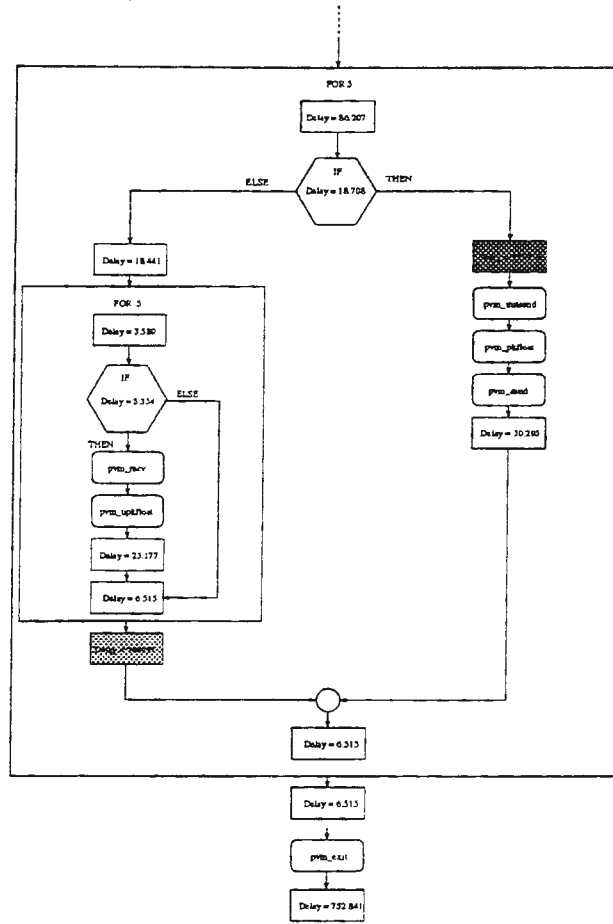
Figure 3: Partial event subgraph for each (SPMD) process, obtained from the first step of the procedure, whose sequential event nodes are labelled with the predicted execution times obtained from the second step.

```
do n = 1, nprocs
  if (n.ne.m) then
    call pvmfrecv(tids(n),any)
    call pvmfunpack(real4,contribute,...)
    store contribute (n)
  endif
enddo
work for me
endif
enddo
enddo
```

In the first step, the partial event subgraphs are created for each process, as shown in figure 3, which illustrates one event subgraph which is representative for all the processes (the code follows the SPMD paradigm), restricted to the portion of code related to one generic iteration of the algorithm.

Blocks of sequential statements are represented as sequential event nodes, and communication statements are isolated and represented as nodes within their control flow context. In the second step, the execution times for the sequential event are predicted, and the corresponding nodes are labelled with their values, as shown in figure 3. The execution times of the algorithm are predicted, with

good accuracy: the sequential block marked with light background in figure 3 represents the updates of elements assigned to each process (corresponding to the white rectangles in the Gantt diagram of fig. 2; the block marked with darker background corresponds instead to the computations of partial sums to be sent to the other processes (the "altruistic" work) which corresponds to the thin horizontal segments in the Gantt diagram.

The third step determines the interaction pattern. The system of equations for the example is:

$$\begin{cases} 1 \leq k, k' \leq niter \\ 1 \leq l, l' \leq nprocs \\ 1 \leq n' \leq nprocs \\ 1 \leq p, p' \leq nprocs \\ p \neq p' \\ l = p' \quad \&\& \quad l \neq p \\ n' = p \quad \&\& \quad (n' \neq p' \quad \&\& \quad l' = p') \end{cases} \qquad (2)$$

The system 2, as represented by the omega library, is:

```
S := {[p,p',k,k',l,l',n'] :
    1<=k,k'<=1, 1<=l,l'<=4 && 1<=n'<=4 &&
    1<=p,p'<=4 && l=p' && l != p &&
    n' = p && n' != p' && l' = p'
    };
```

The resulting set of $(p, p', k, k', l, l', n')$ values, as computed by the omega library, is:

```
{[p,p',1,1,p',p',p]: 1 <= p' < p <= 4} union
{[p,p',1,1,p',p',p]: 1 <= p < p' <= 4}
```

If we want a "section" of the communication space, which represents the communications between p and p', at varying the communication steps $l = l'$, we can take, with the use of the omega library, a corresponding section of that space, and the result is the tridimensional matrix of values (p, p', l):

```
{[p,p',p']: 1 <= p' < p <= 4} union
{[p,p',p']: 1 <= p < p' <= 4}
```

which can be represented graphically (fig. 4) and which is equivalent to the Gantt diagram represented in fig. 1.

The overall event graph for the example under consideration is thus built and represented in fig. 5.

In this case, the communication pattern is deterministic, and the application of the fourth step (non-deterministic resolution) is not needed. The resulting overall event graph exactly predict the communication pattern of the algorithm (this can be noticed very easily by comparing figure 5 with the Gantt diagram of figure 2).

The final result of the above described phase of statical analysis is a labeled event graph of the SPMD code. Such a graph is exploited by an off-line tool to yield, in a straigtforward manner, to a MPSS input format trace file. A similar off-line processor is already available with the simulator for a PVM program real execution trace collected by PGPVM. The MPSS input trace file together with the parameters characterizing the target execution system (i.e. the computing speed of the processing nodes, the type of interconnection network and the percentage of the external network load) makes it possible to start the simulation of the traced program. The event and the timestamps representing the (simulated) execution of the program on the target environment are written into an output trace file. Such a file (in a PICL format) will finally be examined by means of Paragraph graphical tools, to obtain the desired performance figures and diagrams.
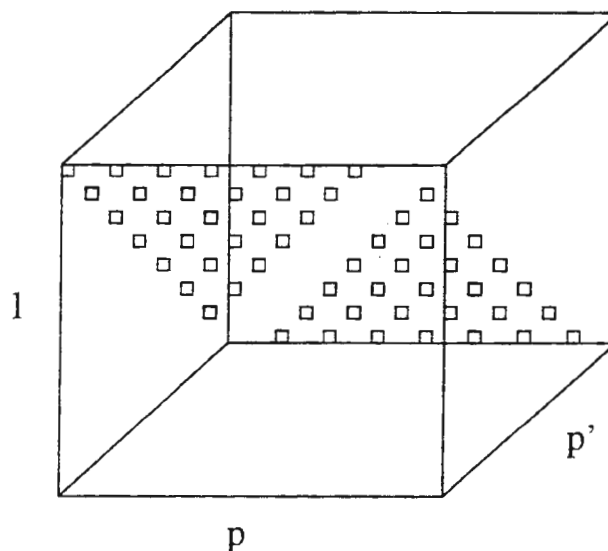
Figure 4: Communications between p and p', at varying the communication steps $l = l'$.

6 Conclusions and Future Directions

The native MPSS input preparation phase requires the availability of the PVM code or an equivalent executable prototype and its execution on almost every possible hardware configuration, possibly less powerful than the target one, or even on a single processor machine. The new proposed approach, on the contrary, allows to obtain an event trace whose label (produced by the static analyzer) are independent from the architecture of the test hardware configuration. In addition, the location of interaction patterns whose communication end-points don't depend from the iteration count number of outer loop, can drastically cut down the execution times of the simulation. This case can be intuitively illustrated by means of the above presented example, where the interaction pattern among the processes (the gather-scatter communication) is repeated *niter* times.

Currently, we are working to this aspect of our approach, both in enhancing the features of the static analysis tool to recognize such patterns, and in fixing the conditions under which it can be assumed with a good approximation that the performance figures of those communication patterns scale up linearly (or according to a whatever foreseeable law) with the number of iterations.

References

[1] R. Aversa, A. Mazzeo, N. Mazzocca, U. Villano. The Use of Simulation for Software Development in Heterogeneous Computing Environments. In Proc. Int. Conf. on Par. and Distr. Processing Techniques and Applications, pages 581-590, Nov. 1995.

[2] R. Aversa, N. Mazzocca, U. Villano. Design of a Simulator of Heterogeneous Computing Environments. *Simulation Practice and Theory*, Vol. 4, May 1996, pp. 97-117.

[3] E. A. Brewer, W. E. Weihl. Developing Parallel Applications Using High-Performance Simulation. In Proc. 1993 ONR/ACM Workshop on Parallel and Distr. Debugging, 1993.

[4] F. Darema et al., "A single-program-multiple-data computational model for EPEX/FORTRAN", *Parallel Computing*, 7, pp. 11–24, 1988.

[5] B. Di Martino and G. Iannello, "Parallelization of Nonsimultaneous Iterative Methods for Systems of Linear Equations", *Lecture Notes in Computer Science* n. 854, Sett. 1994, Springer-Verlag eds.
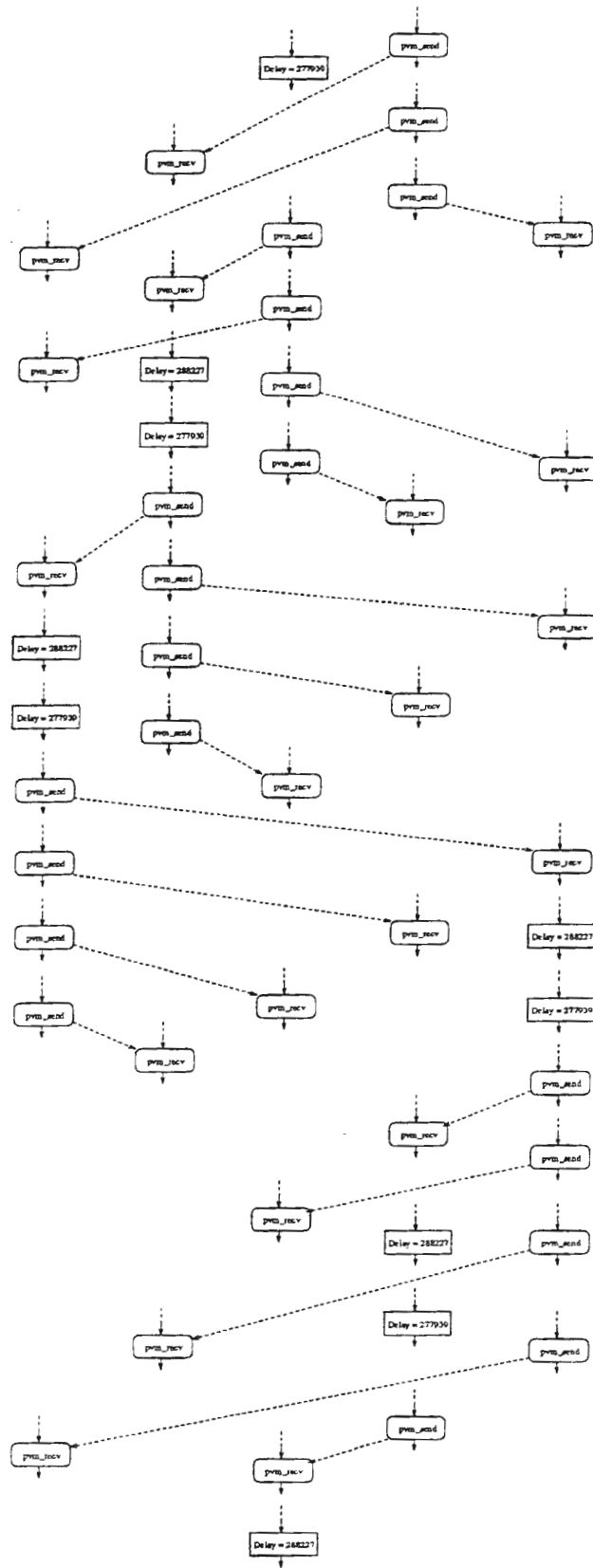
Figure 5: The overall event graph, obtained from the third step. We represent here only communication nodes and the main computation phases. The nodes skipped from previous figure are here replaced by dashed lines. Dashed arrows represent communication edges.

[6] T. Fahringer. Automatic Performance Prediction of Parallel Program s. Kluwer Academic, 1996.

[7] A. Geist et al.. PVM: Parallel Virtual Machine - A Users Guide and Tutorial for Networked Parallel Computing. MIT Press, Cambridge, 1994.

[8] A. Khokhar, V. K. Prasanna, M. E. Shaaban, C. Wang. Heterogeneous Computing: Challenges and Opportunities. IEEE Computer, 26(6):18-27, June 1993.

[9] C. M. Pancake, M. L. Simmons, J. C. Yan. Performance Evaluation Tools for Parallel and Distributed Systems. IEEE Computer, 28(11):16-19, Nov. 1995.

OPTIMAL DESIGN ALGORITHM FOR TECHNOLOGICAL LINES

V. V. Sysoev, I. G. Amrahov, A. O.Movshin

Department of Mathematical Modeling of Technological Systems
Voronezh State Technological Academy,
19, pr-kt Revolutsii, 394017 Voronezh, Russia
Phone/fax: (+7-0732) 552-550,
e-mail: sysoev@vtinauca.voronezh.su, astra@ns1.cbg.ru

ABSTRACT

Technological process is a material basis of any manufacturing. Its features define choice of model-building technique and algorithms of CAD tasks of technological lines. One of such features (for instance. products of electronic manufacturing) is its variance, which consists in capability to get the same results by the various processing manners [1-3]:

- by the applying of various technological processing manners on the same equipment;
- by the applying of various equipment at the same processing techniques;
- by the applying of various technological processing manners on the various types of equipment.

A lot of technological transactions and their variance lead in aggregate to the consideration of a large number possible design decisions. In this paper the vector optimization problem which often arises at the estimation of technological lines running effectiveness, of their designing, scheduling and control is considered. To the studies of such class of tasks are devoted to works [4-7].

1. INTRODUCTION

Let technological line (or the set of equipment) is characterized by some aggregate of parameters. From this aggregate we shall mark a portion which have direct influence on the line running quality. We shall call these parameters by quality indexes and denote q_j ($j = \overline{1,s}$). All the rest of the parameters we shall call varied and denote $\overline{x_i} = (x_{i1}, x_{i2}, \ldots, x_{il})$. $i = \overline{1,m}$, where m - the number of technological transactions of designed line. Each vector $\overline{x_i}$ has its domain of definition D_i. Define as $D = D_1 \times D_2 \times \ldots \times D_m$ the domain of admissible design decisions defined at designing. Coordinates of vectors $\overline{x_i}$ may be vary and it brings to changing of interior structure of technological line. Hence, quality indexes $\overline{q}$ ($\overline{x_1}$, $\overline{x_2}$, ..., $\overline{x_m}$) are changing. Choice of concrete values $\overline{x_i} \in D_i$, $\forall i = \overline{1,m}$ defines admissible design variant $X = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_m}) \in D$.

The problem is formulated in section 2. Then, the global solution approach is presented in section 3. Finally. some numerical examples are given in section 4.

2. PROBLEM FORMULATION

Studied task may be is formulated as following [3]:

$$\overline{q}(X) = (q_1(X), q_2(X), \ldots, q_s(X)) \xrightarrow[X \in D]{} \text{opt} ; \qquad D = D_1 \times D_2 \times \ldots \times D_m ; \tag{1.a}$$

$$x_{ij} = x_{ij0}, \quad j = 1, 2, \ldots, r; \tag{1.b}$$

$$x_{it\,min} \le x_{it} \le x_{it\,max}, \; t = 1, 2, \ldots, k \; (i \ne j \; , \; r+k = l) ; \tag{1.c}$$

$$f_p(X) \le 0, \quad p = 1, 2, 3, \ldots \tag{1.d}$$

where *opt* - operator realizing some optimization principle (for definiteness - minimising). The domain of admissible design decisions D is defined by technological and manufacturing requirements and are formed by the aggregate of the constraints such as equalities (1.b), inequalities (1.c), functional and balance links (1.d).

Generally, for vector criterion we shall introduce rule allowing to estimate the design variants of lines as to Pareto. We shall speak that design variant X_2 is uncoditionally better than design variant X_1 ($X_2 \nmid X_1$) in the sense of vector criterion $\bar{q}$, if $q_i(X_2) \leq q_i(X_1)$ for every i and at least one inequality is strong. If all $q_i(X_2) = q_i(X_1)$, than design variant X_2 is equivalence to X_1 ($X_2 \propto X_1$).

Such rule allows to mark a portion from the whole design decisions set D a subset of the best conflict decisions which are defining Pareto set. Among of that needs to looking for optimal realization variants of designed line.

Optimal designing algorithms are an iterative procedures, per step which are actualized of two stages - the numerical search of Pareto optimal design variants as corresponding to approximations and the sour tan liquor of this set (neglect as to certain rules the parts of these decisions). We shall consider in more detail such iterative procedures of search [8].

3. THE DESIGN ALGORITHM

3.1. Converge theorem

Denote as M_i - the set of conflict decisions X obtained on i-th search iteration; $opt(X)$ - operator carrying out on i-th search iteration procedures of Pareto optimization; $Par(X)$ - operator carrying out along certain rule on i-th iteration the selection of R conflict decisions; $\Gamma(M_i)$ - operator generating on i-th search iteration new admissible decision by given subset M_i. As an operator Par we can use, for instance, the random selection principles along given distributions, selection along aggregate complementary criteria, along utility function, selection by means of DMP (decision-making person). In this case iterative procedure of numerical techniques of optimal designing may be write down in the form:

$$M_i = Par(opt(\Gamma(M_i))) , \quad i = 0, 1, 2, \ldots, \tag{2}$$

where M_0 - some initial approximation. The shutdown condition is $M_k \propto M_{k-1}$, $\propto$ is a relation of equivalence. We shall consider general case. Let selection per the each search iteration is carried out as to vector of complementary criteria $\bar{C}$, operator Par acts on complementary criteria set $\{C_1(\bar{q}), C_1(\bar{q}), \ldots, C_v(\bar{q})\}$.

Theorem 1. If operator Par acts on set $\{C_1(\bar{q}), C_1(\bar{q}), \ldots, C_v(\bar{q})\}$, function C_i and q_i belong to the class of functions $C^{1,1}(E_n)$ (the gradients of these functions satisfy Lipschitz property) than algorithm (2) is converged and $\lim_{i \to \infty} M_i \subset Par(opt(D))$.

Proof. We shall consider the task of search $X^* \in (Par (opt (D))$. It equivalents to vector optimization task for vector-function $\bar{C}(\bar{q}(X))$. We introduce $\bar{\varphi}(X) = \bar{C}(\bar{q}(X))$. If now to show that $\bar{\varphi}(X) \in C^{1,1}$. According to Theorem 3.7 [9], it will be mean that Theorem 1 is proved. We shall show it.

Under hypothesis of theorem functions C_i and q_i belong to the class of functions $C^{1,1}(E_n)$, id est for them are fulfiled Lipschitz property as:

$$\left\| \bar{C}'(\bar{q}) - \bar{C}'(\bar{q}_0) \right\| \leq L_1 \left\| \bar{q} - \bar{q}_0 \right\|,$$

$$\left\| \bar{q}'(X) - \bar{q}'(X_0) \right\| \leq L_1 \left\| X - X_0 \right\|,$$

L_1, L_2 – are some constants.

According to derivation rule for complex function and in consideration of the properties of norm we have:

$$\left\| \bar{\varphi}'(X) - \bar{\varphi}'(X_0) \right\| = \left\| \bar{C}'_q(\bar{q}(X)) \cdot \bar{q}'(X) - \bar{C}'_q(\bar{q}(X_0)) \cdot \bar{q}'(X_0) \right\| \leq$$

$$\leq \left\| \overline{C}'_q (\overline{q}(X)) \cdot \overline{q}'(X) - \overline{C}'_q (\overline{q}(X_0)) \cdot \overline{q}'(X_0) + \overline{C}'_q (\overline{q}(X_0)) \cdot \overline{q}(X_0) - \overline{C}'_q (\overline{q}(X_0)) \cdot \overline{q}'(X_0) \right\| \leq$$

$$\leq \left\| (\overline{C}'_q (\overline{q}(X)) - \overline{C}'_q (\overline{q}(X_0))) \cdot \overline{q}'(X_0) \right\| + \left\| \overline{C}'_q (\overline{q}(X_0)) \cdot (\overline{q}'(X) - \overline{q}'(X_0)) \right\| \leq$$

$$\leq Q \cdot \left\| \overline{C}'_q (\overline{q}(X)) - \overline{C}'_q (\overline{q}(X_0)) \right\| + W \cdot \left\| \overline{q}'(X) - \overline{q}'(X_0) \right\| \leq Q \cdot L_1 \cdot \left\| \overline{q}(X) - \overline{q}(X_0) \right\| + W \cdot L_2 \| X - X_0 \|.$$

Via Q and W were estimated from above values of norms of vector-functions $\overline{q}'(X)$ and $\overline{C}'(X)$, respectively, on the whole their domain of definition. Such estimation lawful, because, if derivative satisfies Lipschitz property than as derivative as function are bounded functions in the domain of their definition. For the norm estimation $\left\| \overline{q}(X) - \overline{q}(X_0) \right\|$ take advantage of well-known mean-value theorem, when $q_j(X) - q_j(X_0)$

$= <\mathrm{grad}(\overline{q}'_j (\xi_j) , X\text{-}X_0>$. Then $\left\| \overline{q}(X) - \overline{q}(X_0) \right\| \leq Q \cdot \| X - X_0 \|$. Having introduced constant $L_3 = Q^2 \cdot L_1 + W \cdot L_2$, finally obtain

$$\left\| \overline{\varphi}'(X) - \overline{\varphi}'(X_0) \right\| \leq L_3 \cdot \| X - X_0 \|$$

The latest means that vector-function belongs to class $C^{l,l}$. Hence, algorithm (2) is converged and $\lim_{i \to \infty} M_i \subset Par(opt(D))$.

3.2. Iterative principle of decisions selection

Let vector $\overline{r} = (r_1, r_2, \dots, r_m)$, where $r_j = \overline{1, S_j}$, $(j = \overline{1, m})$ - the number of realization variant of j-th technological transaction. The cost-performance ratio of designed line, describing by vector X, are uniquely defined by assignment of vector $\overline{r}$.

We shall believe that choice of the concrete variant of technological process on line is evaluated by vector of main effectiveness criteria $\overline{q} = \overline{q}(X)$, which must be minimized, and by vector of complementary criteria $\overline{C}(\overline{q})$, less significant than any q_i. In such case the choice of technological process variant from some domain of admissible decisions $D = X(R)$, where $R = \{ \overline{r} : \overline{r}(r_1, r_2, \dots, r_m) , r_j \in [1, S_j], r_j$ – integer, $j = \overline{1, m} \}$, may be split into two consecutive stages. On the first stage the set of Pareto optimal decisions $opt (D)$ relatively vector of main criteria $\overline{q}(X)$ is built. On the second stage the search of decisions on $opt (D)$ realizing one of the possible schemes of compromise relatively complementary criteria $\overline{C}(\overline{q})$ is carried out. By other words, task

$$\overline{C}(\overline{q}) \xrightarrow[q \in opt(D)]{} opT.$$

where opt - operator realizing optimization on Pareto set which was generated by vector $\overline{q}$; opT - operator realizing one of the possible schemes of compromise relatively $\overline{C} = \overline{C}(\overline{q})$ is solved.

At such formulation we consider one of possible realizations of iterative procedure (2), which scheme is assigned by following easily algorithmic correlations:

$$Par(opt(\Gamma(M_i))) = R_1^* - R_2^* + R_3^* = M_{i+1}, \quad i \geq 0, \tag{3}$$

where $R_1^* = M_i \cap opt(\Gamma(M_i))$ – decisions from M_i, for which in $\Gamma(M_i)$ there are no the best decisions relatively q_j; $R_2^* = \bigcup_{X \in B} \{X^*\}$, $B = opt (M_i)/ opt(\Gamma(M_i))$ – subset of decisions from M_i, for which in $opt(\Gamma(M_i))$ there are the best decisions relatively $\overline{q}$; X^* - decision, mimimizing $C_{n+1}(\overline{q}(X))$ on the set $M(X) = \{X^l : X^l \in opt(\Gamma(M_i)); X^l \} X \}$ - of all decisions from $opt(\Gamma(M_i))$, which are the best relatively $\overline{q}$;

$$
C_{n-1}(\overline{q}(X)) = \begin{cases} \max\limits_{1 \le i \le v} \gamma_i \omega_i c_i(\overline{q}) \\ \quad\text{or} \\ \sum\limits_{i=1}^{v} \gamma_i \omega_i c_i(\overline{q}), \end{cases} \tag{4}
$$

where $\gamma_i > 0$; $\sum\limits_{i=1}^{v} \gamma_i = 1$; ω_i – monotone transformation of C_i criteria, reducing its to the same dimension and to minimization problem; R^*_i – set of decision, consists of less than $R = \min \{card(R^*_1 \div R^*_2),\ card(R^*_4)\}$ of first elements of set $R^*_4 = opt(\Gamma(M_i))/(R^*_1 \div R^*_2)$, which is ordered in ascending order.

Criterion $C_{n+1}(\overline{q}(X))$ assigned by correlation (4), defines the possible representation of the compromise scheme relatively vector of complementary criteria $\overline{C}$ at the task solution.

Considered algorithm allows in various degree to involve DMP to task solution:

a) DMP before began making-decisions chooses type of function C_{n+1} and assigns weight γ_i;
b) DMP on the search iterations (in particular, not on all), depending on previously iteration decision, corrects type of function C_{n+1} and assigns weight γ_i;
c) DMP completely fulfils the role of criterion C_{n+1}, carries out per each iteration regulation and the choice of receivable decisions.

Each of variants a) - c) defines one of possible principles of the decisions selection *Par* on the iterations of algorithm which are put into base at development of algorithmical and program parts of automated subsystem as to the choice of optimal variants of the equipment and technologies composition for the technological lines of the products of electronic manufacturing.

4. OPTIMAL CHOICE OF DESIGNED TECHNOLOGICAL LINE EQUIPMENT

The optimal equipment choice problem was solved with the objective to study the interaction between plates technological processing workshop and photolitography workshop.

The manufacturing of the electronic product is a linear chain from 80 transactions. Taking into account equipment capability to be used for different transactions with various numbers, technological line is reformed into type consist of ten unified transactions. Variance on the transactions is defined by possibility of application of the various equipment types. The quantity of possible transactions execution variants is given by following correlations: $S_1 = S_9 = 3$; $S_2 = S_4 = S_5 = S_6 = S_7 = S_8 = 2$; $S_3 = 4$; $S_{10} = 5$. The gross amount of the possible technological process variants is defined as like $N = \prod\limits_i S_i = 11520$.

Optimization was conducted along 8 criteria. Four of them were chosen as main: the cost of unit released product ($q_1 \to$ min); specific mentioned costs ($q_2 \to$ min); product takeoff with units area ($q_3 \to$ max); automation and mechanization level of production process ($q_4 \to$ max). The rest of the criteria - area occupied by system; the cost of the equipment set; economical efficiency index; aesthetic index - were relegated to complementary for the choice of decisions per the search iteration. The plan of the product output from line was varied in interval [115000 item/year; 200000 item/year] with step 85000 item/year. Such variation, in its turn, increases the number of possible line realization variants up to $N=115200$.

On each optimizing iteration procedure 2-level task is solved. On the upper level is solved

$$
\overline{q} = (q_1(X), q_2(X), q_3(X), q_4(X)) \xrightarrow[X \in M_D]{} opt, \tag{5}
$$

where – M_D the finite set of decisions $X \in D$, from which the search is started. As a result of task (5) per each iteration given number of variants of designed line incomparable on Pareto is formed. That is set M_i. On downlevel is solved

$$0.3\ q_5(X) + 0.3\ q_6(X) + 0.25\ q_7(X) + 0.15\ q_8(X) \xrightarrow[X \in M_D]{} par\ , \tag{6}$$

where par - operator realizing rankig and the selection of given number $X \in M$ in the sense of the minimum of complementary criteria convolution; 0.3; 0.3; 0.25; 0.15 – are weights corresponding to complementary criteria (are assigned by MDP). As a result of task (6) solution the ordered set of the given number of design variants is formed. It is a Rang $\subset M_i$.

MDP in optimization procedure was used for the generating of initial variants, with which descent was started.

As to results of two local descents and final «global» sifting as to Pareto of received obtained decisions the four optimal variants was found: $X_1 = (\bar{x}_{12}, \bar{x}_{21}, \bar{x}_{34}, \bar{x}_{41}, \bar{x}_{52}, \bar{x}_{62}, \bar{x}_{71}, \bar{x}_{82}, \bar{x}_{93}, \bar{x}_{105})$ for output plan $Z = 115000$ item/year; $X_2 = (\bar{x}_{12}, \bar{x}_{21}, \bar{x}_{34}, \bar{x}_{41}, \bar{x}_{52}, \bar{x}_{62}, \bar{x}_{71}, \bar{x}_{82}, \bar{x}_{91}, \bar{x}_{105})$ for $Z = 115000$ item/year; $X_3 = (\bar{x}_{13}, \bar{x}_{21}, \bar{x}_{31}, \bar{x}_{41}, \bar{x}_{51}, \bar{x}_{62}, \bar{x}_{71}, \bar{x}_{81}, \bar{x}_{91}, \bar{x}_{105})$ for $Z = 115000$ item/year; $X_4 = (\bar{x}_{12}, \bar{x}_{21}, \bar{x}_{34}, \bar{x}_{41}, \bar{x}_{52}, \bar{x}_{62}, \bar{x}_{72}, \bar{x}_{82}, \bar{x}_{93}, \bar{x}_{105})$ for $Z = 115000$ item/year.

5. CONCLUSION

Considered scheme may be used not only in design tasks but for solving problems of optimization with vector criteria allowing 2-level search iteration procedure. We can refer to such procedures the generalization on vector case such well-known techniques as coordinatewise optimization, random search with local optimization, dynamic programming, etc.

6. REFERENCES

1. R.V. Kornilov, V.P. Sanderov, "Calculation of Equipment Complexes in Microelectronics". Moscow, 1979. (in Russian)
2. V.A. Egorov, "Computer-Aided Design of Enterprises". Leningrad, 1983. (in Russian)
3. V.V. Sysoev, "Computer-Aided Design of Lines and Set of Equipment of Semiconductor and Microelectronics Manufacturing", Moscow: Radio i sviaz, 1982, 120 p. (in Russian)
4. V.V. Podinovsky, V.D. Nogin, "Pareto Optimal Decisions in Multicriteria Problems". Moscow, 1982. (in Russian)
5. O.I. Larichev, "Science and Art of Making Decisions". Moscow, 1979. (in Russian)
6. Fishbern
7. B. Rua, "Problems and Methods of Making Decision in Tasks with a Lot of Target Function", in Problems of Analysis and Making Decision, Moscow, 1976, pp.20-58. (in Russian)
8. V.V. Sysoev, "Mathematical Modelling in Frame of Computer-Aided Design of Lines and Set of Equipment Semiconductor Manufacturing". Proc. of "Mathematical and machine modelling" Conference, Part. 2. Voronezh. 1988. pp. 3-7. (in Russian).
9. V.V. Sysoev, "Structural and Algorithmic Models of Computer-Aided Design of Electronic Production", Voronezh: Voronezh Technol. Inst., 1993, 207 p. (in Russian)

Perceptual Models of Artificial Intelligence in Agriculture

Gennady A. Moskvin

Latvia University of Agriculture
Liela iela 2, Jelgava LV-3001,LATVIA

Abstract: The scientific work contains the researches of automatic measuring and control systems on the basis analysis and synthesis of the perceptual, mathematical and fractal models ,of elaboration of physical " metric images " by means of creation of intellectualized bootstrap -methods of measuring and identification ,for example, of critical control points in the *HACCP* -systems in different fields of agricultural production by using operating microcomputers. The prior tasks of the research are the fundamental *AI* theoretical problems and perceptual - metrical ensuring of measuring channel "Primary Sensing Elements -microcomputer", substantiation and elaboration of structurally-functional ties of *AI* automatic control and expert systems in agriculture.

Keywords: Modelling,*AI* application, Automatic Control and Expert System.

Introduction

The research objects are perceptual and fractal models of *AI* in agriculture, agricultural technological processes and installations, intellectualized measuring systems, primary sensing elements *(PSE)*, patented *"AI-language"* and *"AI- nose"* testers and other measuring microcomputer devices *"Logicor"*, raw materials and food products, quantity and quality evaluation technology of agricultural products (control and classification of physical properties and characteristic , etc.), programs and algorithms of "intelligent" recognition, learning, teaching, self-diagnostics, perceptual, fractal and metrical ensuring of measuring channel *"PSE* -microcomputer" evaluation, control and regulation.

Methods

The peculiarity of method of the research has "intelligent" automatic and expert perceptual modelling and fractal computer vision control of quantitative and qualitative parameters including the ones in the conditions of possible informational and technically-technological insufficiency, of not fully right order and production process technology organize, when information and knowledge material resources are limited, at the presence moral and physical wear of measuring system elements and possibilities of flexible operating when there is not formalized state of not optimal structures and lack of sufficient volume of systematized data, entering information stochastic character in agriculture. With that complex problems optimal solution is found when reaching limited price and mass-volume size indices.

Results

On the basis measuring process perceptual, mathematical, fractal modelling and experimental research results, are to find:

1. Basic Principle of Perceptual Modelling of AI Systems

Fundamental research in the field of Artificial Intelligence *(AI)* allows to create abstract universal models which "*improve*" real processes and, in the result, the investigated situation becomes "*transparent*", accessible to theoretical analysis, generalisations and for the defining of new laws and the acquired new knowledge.

The elaborated perceptual model of *AI* systems is an abstract, universal object. Regarding real processes it is a peculiar theorised model, a universal prototype for all definite *AI* control systems carrying out the role of "*secondary*" models. In a certain meaning, the universal perceptually-abstract *AI* object is richer in its content than its "*secondary*" models because there, in a narrower form, are included many yet non-realised solutions.

AI perceptual modelling principle can also be used in mathematical biology and bionics which now are rapidly developing. Relationship among such different biological prototype "features" of *AI* "reason" as "sensing", "perception", understanding", "notion", "conception", "judgement", "deduction" (Fig.1), etc. characterises the whole of *AI's* "reason and organism" as a systemic structure. This relationship remains the same, t.i., invariant for all *AI* structures, "*organisms*" regardless the differences they may have in the technical-technological structure of their elements and component-parts.

Conformity among "*biological*" features of higher and lower *AI* systems of a various complexity can be expressed by means of the mathematical concept "*epimorphism*", t.i., by unequivocal and ambiguous compliance of two group elements when one group reflects the other group. Then the structures of different *AI* systems "*organisms*" epimorphically reflect each other. Moreover, the basic relationship characterising *AI* "cyber-organized organism" as a whole system remains.

Let's introduce the concept "*primary*" or "*initial organism*" as a perceptual model's auxiliary construction of *AI* "*organism*". Such generalised organism is a system of bio-invariants, t.i., it includes the most significant "*biological*" features and relationship among them.

Subsequently, the real "*organisms*" of *AI* systems can be expressed by means of the accepted conditioned standard of the perceptual model (experts knowledge). Besides, the most significant *AI* "*biological*" features remain.

It can be said also in other way: real *AI* organisms are initial *AI* "the projections of the organism" designed by experts-theoreticians, models on in reality existing organical reason forms.

It is easy noticeable that the "*initial AI organism*" is a peculiar sensorically-metrically realised world outlook spatially-temporal image of perceptual consciousness. Generally, it can be interpreted as a universal abstract object created by modern bionics through normalisation and mathematization of its basic notions (1).

For example, in accordance with calculations carried out on the basis of the biological epimorphism principle, the number of species which might originate on the Earth is 10^8. But the number of species living on our planet is 10^6, that is, one hundred times less. Subsequently, it can be concluded that there are possible different biological forms from which only one is realised in the course of evolution of the organic world.

It is also significant that the elaborated universal *AI* abstract perceptual model in its contents one hundred times surpasses the real species group of live matter by their diversity.

Noteworthy is the condition that just biological cybernetics, bionics already now can implement the idea of designing *AI* universal abstract object. It allows a simpler way to elaborate *AI* bionic control systems making use of not simply a flow of information data group from sensors and primary measuring transducers but knowledge ready for use and "*perceptual processing*" or esoteric knowledge (2).

The offered information processing and application principle in *AI* bionic expert systems will allow the operator (a person or apparatus substituting man's control functions) to receive more complete, operative, "*momentarily*" knowledge for the monitoring, control and regulation of bionic processes and objects, and not a "bare", non-processed number information group which rather misinforms, disorientates than informs or takes away uncertainty about features, peculiarities and control parameters of the technological process under exploration, for example, in the food product quality control *HACCP* expert system - *Hazard Analyses in Critical Control Points*.

Not in vain *U. R. Eshby*, one of the founders of cybernetics, assumed that the object of cybernetics is the branch of all the possible machines, and only secondly it is interested in the fact that some of these machines are not yet created neither by a man nor by nature (3).

AI perceptual model gives a general carcass on which automatic control principles of all individual machines can be located, correlated and understood. By "*machine*" there is meant a device (its features can be most various) accomplishing a definite kind of behaviour by means of automatic control.

The universal, generalised abstract *AI* object comprises a too great diversity as compared to the aggregate of all the definite devices. As to it, the real "*machines*" are "*models*" or "*projections*".

At the moment in biology a huge file of the information has collected. By generalizing and by ordering it, there is the opportunity to create generalized functional models of separate parts of the person, each of which would be considered from the various points of view.

For example, within the framework of the project BioNOME (1997,UCSD -San Diego Supercomputer Center in California, USA) creation as of mechanical model of heart is possible, and its electrical analogue. The problem consists in organic connection these mutually of adding models in uniform organism . But for achievement of this purpose understanding of interrelations between these diverse systems is necessary. The project BioNOME is in an initial stage and the central place in it is occupied by researches,connected

with study of ways both mechanisms of carry and exchange of the information by chemical substances in biological cells and between them, also study of functionally integrated models of work of heart.

Elaboration and further perfection basic principles of bionic AI and expert systems are revealed in the light of systems theories realising general control functions.

2. AI Application in Agriculture

In the conditions of modern open society special significance is paid to the interests of consumers and a free exchange of information on the quality of products and changes in food properties during production, marketing, storing and consumption. Unfortunately, present achievements do not offer wide possibilities for the use of comparatively simple, safe and inexpensive intellectualised sensors to express-control different food products and raw materials in all the stages of the obtaining, production and realisation of agricultural produce. The problem still remains after the realisation of the production when its quality and nutritive property changes are not subjected for further control. Moreover, consumers and wholesalers do not have neither measuring devices, nor relevant knowledge. It creates the situation when consumers health and wholesalers business activities are in a certain risk situation.

The results of previous experimental research prove that the quality of food products and raw materials in the *HACCP* system can be operatively determined by patented electronic devises and methods *"Logicor"* making use of correction methods and intellectual sensors with *Artificial Intelligence* elements.

Such electronical devices in the form of expert systems can also be practised in private and dairy farms for the determination of milk freshness (sourness), naturality, state of health and physiological condition of cows, for the detection of oestrus and mastitis, etc. These devices are already used in dairies and clinical laboratories. In general, it considerably allows, on the one hand, to improve the informative express-service basis of the complex program of the Latvian Republic "Feed" and "Selection", and on the other hand, to implement practically the Law of the Latvian Republic on the protection of consumers interests and rights, as well as *HACCP* National program.

3. Automatic Identification of the Critical Control Points

CCP - critical control points should be determined in all the stages of food production starting with the obtaining of raw materials or the production of component parts up to their marketing. Therefore, firstly, during the technological process a precise, safe, operative and objective information flow has to be established throughout all the production stages.

The solution of this problem is hindered by the lack of such measuring devices and suppliers which during testing, regulation and control of technological process parameters systematically, energetically, constructively, informatively, exploitatively and, what is most important, metrologically could be joined not only to the control systems but also to technical and measuring devices in real conditions. Just in such way can be explained the world tendency towards the *"intellectualisation"* of measuring devices and suppliers (3).

In industrial production there are no analogous for such food product properties as nonuniform flow of materials and informative resources, significant changes of their properties and quality in time, presence of inertia in the communications with a biotechnical system.

The above said does not allow to apply the traditional methods and means in the control of technological processes and in the food production processes. The situation is worsened by the low technical level of the existing suppliers and devices and, in the first place, their preciseness and credibility. Theoretic investigations prove that suppliers lag behind the development of food and other technologies therefore all over the world intensive financing is observed just in the field of technical progress.

The elaboration of intellectual suppliers becomes the main direction of automatization. Improvements can be achieved by providing the measuring transducers with microprocessors not only for the supplier but also for the whole device.

Intellectual suppliers can essentially perfect the whole control system due to the increase of preciseness and a rational processing of signals received from the sensory element. Such existence problem of a measuring device and a supplier is the problem of a precise control of production processes, the problem of consumers provision with qualitative food products, efficiency problem of any production.

Therefore it is topical to design new generation *CCP* measuring devices with the application of computing microdevices which can effectively work under changing operation regime of equipments, as

well as adopt themselves to definite food technology processes with unsystemized parameters and unformalized requirements in real exploitation conditions.

The leading elaborators abroad are of the same opinion that it is practically impossible for a primary measuring transducer (supplier) to realize precision requirements in the designing, production and exploitation stages. Therefore recently throughout the world new conceptions are searched for in the elaboration of appliances and automatization devices taking into consideration the growing demands of consumers. One of the development directions can be the elaboration of suppliers with "Artificial Intelligence" (AI) elements which continue the development of microprocessors technique technology. Within the framework of the National *HACCP* program(two grants of Latvian Scientific Council are accepted - LZP 96.0072 and TPP- 96-48) in laboratory of Artificial Intelligence of Latvian Agricultural University intensive work on creation of microdevices " *of artificial taste* " (artificial language) and " *of an artificial smell* " (artificial nose) are conducted.

Such intellectual suppliers are primary measurement transducers with electrical or other exit initial signal and the synthesis of computing microdevices with a control keyboard and microprocessors.

Food production peculiarities, conditions and specifics require to elaborate simple, safe, inexpensive and precise suppliers. Elaboration of such suppliers is the decisive factor in the quality of food and other products control system operation.

4. The Future of AI in Agriculture : Myth or Reality?

The scope of measuring tasks is growing and measuring amount is increasing. The portion of measuring devices (in a monetary expression) reaches even 10...18 % from the value of production enterprise equipment. Robotized enterprises need up to 100 000 different suppliers and measuring devices. In special nano-technologies there are applied up to 1 000 000 different suppliers.

Technological and informative revolution in all the production spheres, especially in computing- and research-comprising technology branches determined the application of local (divided intellect) systems in the existing (functioning) structures and a further development of local microcomputers.

Subsequently, in the automatization of technological processes the leading part belongs to electronization and intellectualization of suppliers and measuring devices. Just electronics showed the impact on the development of measuring devices at the same extent as achievements of physics on the development of metrology. Intellectualized measuring devices are at the basis of the elaboration of all automated production processes.

Many measuring devices usually consist of two functional knots: a primary measuring transducer and a registering device. Suppliers usually have the electric exit signal and a further processing measuring information is completed by using different electrical schemes, mainly, of an analogous type.

As to functional opportunities, preciseness and signal stability, the processing of digital data has significant preferences. The presence of a microcomputer in the measuring channel allows (at a definite mathematical programming) to carry out also "identification" experiment of the computing (modelling) measurable medium by making use of many physical effects (1).

Possibilities of the computing technique allow to process operatively a great amount of information by a definite algorithm and promote not only an optimal collection of measuring data from suppliers but also a correction of their mistakes and interpretation of the measuring result in accordance with the requirements of information users and production consumers.

Biocybernetic, bionic approach in the modelling of biotechnical measuring systems allow to examine two types of "*intellectual*" control models.

For the elaboration of the first type models it is sufficient to study in isolation only "inner" parameters and processes of the object under exploration without taking into consideration the impact of outer medium factors and, in relation with it, "behaviour" changes of the structure intended for synthesis. The modelling of this type can be useful for a preliminary "*metric image*" identification of the object under exploration.

The further use of the model depends only on the success of the acquired model's theoretic and technical continuation.

"Behaviour" factor analysis of the object of interest has to be considered at the basis of the second type of biocybernetic models. Further on "reference" functions of the influence factors are determined and a feedback algorithm is synthesized in the form of mathematic programs through "*self- learning*". The basic

contours of models technical realization are formed in a complete agreement of the existing notions, data, levels of knowledge about the investigation process or object with the exploration task and aim.

"*Artificial Intelligence*" refers to any object in such a way as it is required by its nature. Universal independence of thoughts, "intellect" can find original, effective and even unexpected answers, for example, "*watch - fractal images*" of *CCP* of fruits and vegetables (Fig. 2).

Such methodological approach in the elaboration of intellectualized *CCP*-suppliers very often in the measuring process does not provide a sufficiently complete idea about all the "inner" processes going on in the object under exploration because the functional structure of the synthesized device is aimfully oriented only towards the realization of in advance programmed functions. Successively, it can be assumed that registering co-operation results of the investigation object with different measuring devices, the object's properties, which have to be fixed, can be so significantly different that there can be no "*essence*" at all in the indications of the measuring device because the exploration object is always connected with a definite "*space-time*" co-ordinate system (*principle of relativism*).

As the result, three main reasons are pointed out, that define the errors in modern developed *AI* bionic and intelligent measuring systems:

• errors, due to relativity theory theses incomplete, unfinished, unperfectness - fundamental problems of world outlook, space and time;

• fundamental problem of error excerpt removal;

• errors of perceptual, mathematical, fractal, physical, statistical, normative, metrical and associative imagination models inaccuracy.

Presence of specific agricultural technological and bionic -industries processes caused the creation of such expert system model for intelligent measuring processing in precision agriculture which at a higher rate would correspond to perspective tasks and aims both of *AI* systems and accurate physical magnitude measurements with the account of above mentioned error relativity and world outlook spatial - temporal fundamental problems of *AI*.

From the point of view of *CCP*-parametrical identification bionic systems intellectual compatibility of measuring processes and functions of the "*compensating stage*" can be taken over by the cognition subject with its intellectual apparatus which adds to the possibilities of applied investigation methods. In the elaboration of bionic intellectual measuring systems it has to be understood that such a system has to be open to man's (expert's, specialist's) intellect, knowledge, practical experience (also nonformalized and unsystemized) and even to intuition. Then the intellectualized measuring system can be continuously improved through training simultaneously and in accordance with the development of standards and criteria.

In the result of a dialogue "through self- learning" is formed for further development and perfection the open "intellectual" *CCP*-indication, identification, measuring and correction functionally pragmatical metric image identification system of the physical medium which is arranged (organised) *not towards the measuring process but towards the measuring result.*

AI and qualimetry laboratory specialists of the Latvia University of Agriculture have elaborated a complex of electronical *CCP* equipment easing the quality control problems of everyday products quality in production, processing, realization and commercial payment stages.

At the present moment it is planned to commence the production of these electronic devices, *CCP* testers-analysers, "*AI*-nose" and "*AI* -language" sensors, intelligent equipments of *CCP*-indicators , of automatic dosing and registering devices (4,5) for various raw materials and liquid products (6,7). Experimental research and practical tests proved the undeniable effectivity of these intellectualized electronic *CCP*-devices.

It will be possible to order intellectual *CCP*-testers of food product quality control (milk and milk products, strong and alcohol-free drinks, beer, sunflower oil, meat products, confectionery and bread products, flour, grain, fruit and vegetables, etc.), oil products quality identification testers, dash and products naturality testers in different versions: with a measuring scale, with light-sound alarm signals, or with a measuring scale, light and sound (3).

HACCP future system includes a systematic *CCP* evaluation for all food production stages starting with the obtaining of raw materials or production of component parts up to the moment when finally the product comes to the consumer, defining the stages which are critical to healthy food products. Intellectual informative technology and intellectual *CCP*-identification devices is a safe and universal work instrument for the bringing nearer and changing of *HACCP* implementation future for the benefit of food and other product consumers.
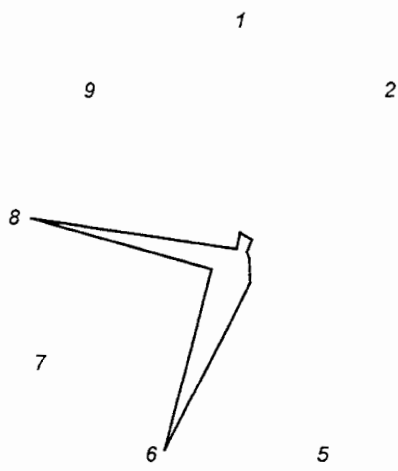
References:

1. Москвин Г.(1997).Пространственно- временная идентификация и коррекция биотехнических систем искусственного интеллекта. Материалы международной конференции "Автоматизация с.х. производства ". - Москва, Россия, РАСХН- ГКНТ, 14 - 16.05. 1997, №15, 6 с.

2. Кибалион, (1992), Учение геометрической философии Древнего Египта и Греции. Пер.с англ.- Москва.,1992,108 с.

3. Moskvin G.A. (1996) Intellectualized systems of automatic measuring. A review of sc. works cont.LLU,Jelgava,p.67-70.
4. Moskvin G., (1991) USA Patent N 5.012. 762.
5. Moskvin G., (1991) USA Patent N 5.016.569.
6. Moskvin G., (1992) USA Patent N 5.161.483.
7. Moskvin G., (1991) USA Patent N 4.989.445.

"He who does not know where he goes, has no difference which way to take" (C. Lewis, "Alice in Wonderland")

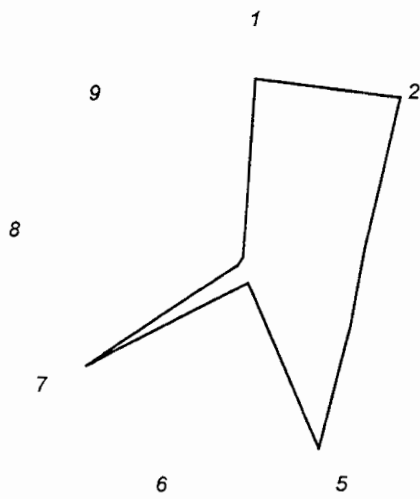Fig.1 Generalized model of knowledge tree for the AI and expert systems in Agriculture

Fractal Image of Banana-4

Double-Fractal Metric Image of "Banana"
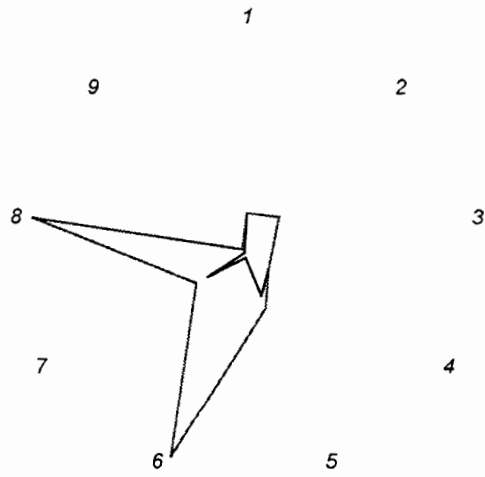
Watch-Fractal Image of Banana-3

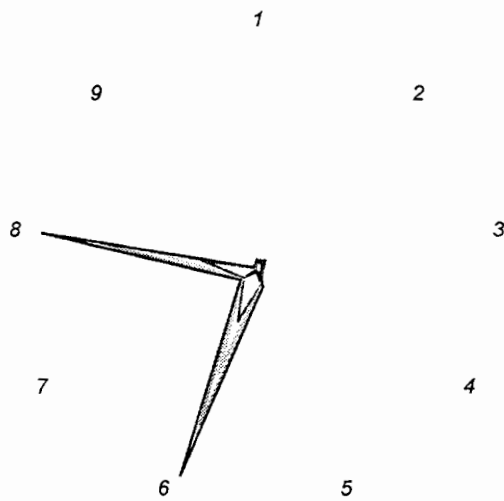Generalized Watch-Fractal Image of "Banana"

Fig.2 "Watch-fractal" metrical images of the fruits and vegetables ("Banana-3" have damages)