FORECASTING BATCH COOKING RESULTS WITH INTELLIGENT DYNAMIC SIMULATION

Esko K. Juuso

Control Engineering Laboratory,
Department of Process and Environmental Engineering,
P.O.Box 4300, FI-90014 University of Oulu, Finland

esko.juuso@oulu.fi(Esko K. Juuso)

Abstract

Intelligent dynamic simulator has been developed with linguistic equations *(LE)* for a *SuperBatch* coking process. The LE models consist of two parts: interactions are handled with linear equations, and nonlinearities are taken into account by scaling with membership definitions. Data-driven modelling was done by *FuzzEqu* toolbox on the basis of the measurements of the cooking liquor analyser *CLA 2000*: the concentrations of alkali, total dissolved solids and lignin are measured on-line during individual cooks in a batch digester house. Effective delays depend on the working conditions. Universal approximators for fuzzy functions can be constructed as extension principle extensions of continuous real-valued functions which continuously map fuzzy numbers into fuzzy numbers. LE models can be extended to fuzzy inputs with this approach if the membership definitions are replaced by corresponding extension principle extensions of these functions. The argument of the model function is obtained by fuzzy arithmetics. Fuzzy extension of the classical interval analysis suits very well to these calculations. According to extensive on-line tests in an industrial pulp mill, dynamic *LE* models are well suited for forecasting the cooking result: residual alkali, lignin and dissolved solids. The models are adapted to the changing operating conditions with configurable parameters. Uncertainty of the results is clearly seen when the resulting residual alkali, final lignin and final solids are presented as fuzzy numbers. The width of the fuzzy numbers depends on the length of the calculation period. The fuzziness of the result is smaller for the simulations performed later during the cooking sequence as the accuracy of the prediction is improved when the cooking progresses.

Keywords: batch cooking, intelligent methods, dynamic models, forecasting.

Presenting Author's Biography

Esko K. Juuso has M.Sc. (Eng.) in Technical Physics from University of Oulu. He is currently a senior assistant in Control Engineering at University of Oulu, Oulu, Finland. He is active in Finnish Simulation Forum (FinSim), Scandinavian Simulation Society (SIMS) and EUROSIM, currently he is chairman of FinSim. His main research fields are intelligent systems and simulation in industrial applications, including software sensors, control and fault diagnosis. For these applications he has introduced the linguistic equation (LE) methodology.

ISBN 978-3-901608-32-2

1 Introduction

Cooking is the major process in the pulp mill and its proper control is very important to the pulp production. The aim of chemical pulping is to remove enough lignin from wood so that the fibers are free and give them the required characters at the lowest possible cost. Variation in the quality parameters is minimised by providing the similar cooking history for all chip particles. Chip quality variations, measurement problems and long process delays make the control difficult. The main problem is that the quality variables cannot be measured while the cooking goes on. Changing operating conditions introduce challenges also for mechanistic modelling. Pulp cooking has used computerised control since the early 1960s [1].

Cooking reactions start when chips reach the cooking temperature, about 150 - 170 ^oC, depending on the wood species and grade requirements. The active chemicals of the cooking liquor react with lignin in chips and convert it chemically into the compounds that dissolve in the cooking liquor. Fibers are separated into the mass since the bonding material of the chips is dissolved. Cooking chemicals also react with fibrous material (cellulose and hemicellulose) and break down their molecular bonds. These reactions are to be avoided as much as possible because they lead to losses in the fiber yield and decrease the pulp strength.

There are two main types of process equipment for cooking: batch and continuous digesters. This paper deals with the batch cooking process, more specifically *SuperBatch* cooking, where the chips are preimpregnated and preheated with warm black liquor (Fig. 1) to make batch cooking more energy efficient.

The cooking liquor is circulated from the middle to top and bottom to ensure a uniform cooking. When the target H-factor has been reached, the circulation is stopped and the cool displacement liquor is pumped into the bottom part of the digester. The delignification process stops when the temperature decreases under 100 ^oC. The used cooking liquor is taken out from the top and directed to the accumulators to be reused.

For continuous cooking, many cooking models are based on physical-chemical phenomena. Most of them consider that the delignification happens in three consecutive steps, labeled as initial, bulk and residual delignification, respectively [2]. Some models divide the lignin into high- and low-reactivity lignin [3, 4] and assume that the reactions take place simultaneously. Aguiar and Filho compared phenomenological models, neural models and hybrid models, which combine both approaches [5]. *Kappa number* predictions are good but the networks are rather large.

Cooking liquor analyser CLA 2000 is an advanced measurement device developed for analysing the chemical pulping processes (Fig. 1). It utilises a continuous sampling technique that means also, in practice, a continuous measurement if the analyser is applied for one sample point, only [6]. In batch cooking, the cooking liquor analyser enables the analysis of alkali concentration and concentrations of total dissolved solids and dissolved lignin during individual batches.

The widely used quality variable, the Kappa number, can be predicted on the basis of these measurements before the end of the cook. As the digester process is far from linear and simple input-output system, the analysis must be nonlinear. Different approaches have been used for mathematical modelling of the cooking result [7]: fuzzy logic, partial least squares method (*PLS*), artificial neural networks (*ANN*) and linguistic equations (*LE*). In modelling the Kappa number in a continuous digester, all these methods seem to learn the process behaviour in a similar manner, but the *LE* models are the best in process environment since they can be adapted to various operating conditions in an understandable way. *ANN* and *PLS* models are sensitive for changes in process conditions, and fuzzy models need a large number of membership functions and rules that are too time-consuming to adapt.

In batch cooking, dynamic modelling and simulation is necessary. Dynamic *LE* models have been used in tuning *LE* controllers since they have provided accurate prediction and good performance in continuous processes, e.g. a lime kiln and a solar collector field [8]. For batch cooking, the dynamic *LE* modelling was started in 2000 [9]. According to the on-line tests, dynamic *LE* models are well suited for forecasting the cooking result.

The approach is generic for batch processes: in [10] modelling of a Super Batch cooking process was compared to the modelling of a fluidised bed granulator, and later the approach has been applied in fed-batch fermentation [11]. A set of practical and interactive intelligent systems can be combined with other modelling and simulation methodologies to build practical simulators for industrial processes [12].

The *FuzzEqu Toolbox* includes routines for developing, tuning and testing linguistic equation systems [13]. The LE systems can be modified interactively to adapt the models to changing operating conditions. Links to *Matlab* toolboxes facilitate comparisons with various modelling approaches, e.g. fuzzy set models, artificial neural networks, decision trees, regression models and system identification. It provides routines for building *LE* systems from large fuzzy systems including various ruleblocks implemented in FuzzyCon [14] or *Matlab* FuzzyLogic Toolbox. Other fuzzy modelling approaches can be used as channels for combining different sources of information [8].

Modelling and simulation results of a *SuperBatch* pulp cooking have been presented in [15]. The simulation study concentrated on the heating and cooking sequence was done in co-operation with *ABB* and *UPM-Kymmene* in an industrial pulp mill.

This paper presents more details of the solution and extends the models to uncertain environment.

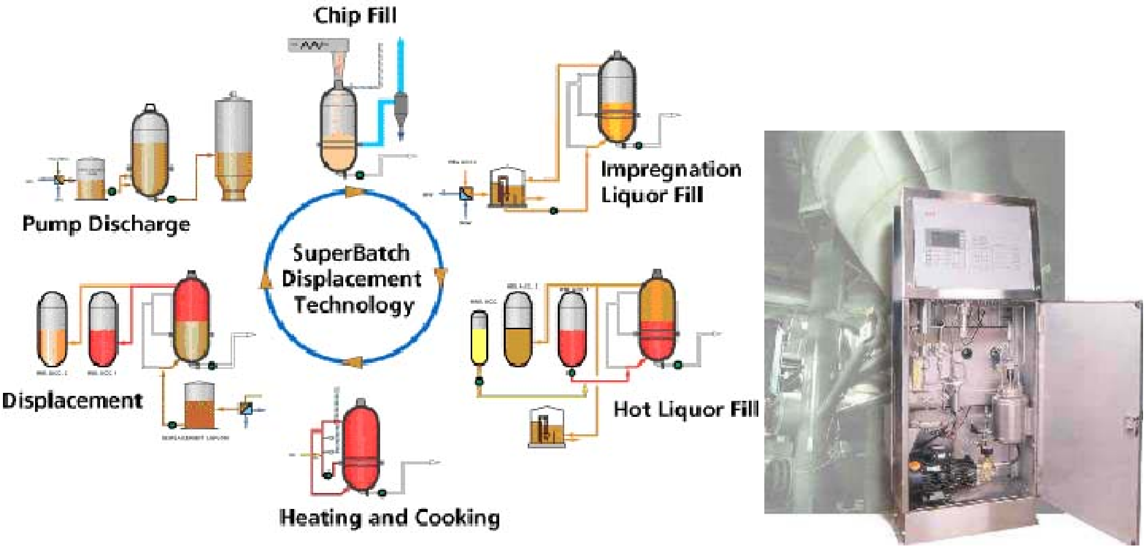

Fig. 1 *SuperBatch* sequences and cooking liquor analyser *CLA 2000*.

2 Measurements

Alkali concentration and concentrations of total dissolved solids and dissolved lignin give information showing how cooking reactions go on during individual batches. With batch digester operation it is possible to terminate the cooking process just when the required cooking degree has been achieved. Measurements are carried out in the actual process concentration and no dilution is needed. Measurements are indirect: alkali components are based on conductivity, solid contents on refractive index and dissolved lignin on UV-absorbency [16].

The models presented in [15] have been tested with an on-line simulator in an industrial digester house. A large set of experimental data from various dissimilar operating conditions was collected during this study.

3 Modelling with linguistic equations

Linguistic equations introduced in [17] have been used in various applications [8, 18]. Data-driven steady state modelling has been used in development of LE model to represent interactions between three measurements of the cooking liquor analyzer. Dynamic structures extend the models to dynamic simulation of these measurements during the cooking period.

3.1 Steady state modelling

Linguistic equation models consist of two parts: *interactions* are handled with linear equations, and nonlinearities are taken into account by *membership definitions* [8]. The basic element is a compact equation

$$\sum_{j=1}^{m} A_{ij} X_j + B_i = 0, \qquad (1)$$

where X_j is a linguistic level for the variable j, $j = 1...m$. Linguistic values very low, low, normal, high, and very high correspond to integer numbers -2, -1, 0, 1 and 2. The direction of the interaction is represented by interaction coefficients A_{ij}. The bias term B_i was introduced for fault diagnosis systems. A *LE* model with several equations is represented as a matrix equation. Linguistic equations can be used to any direction.

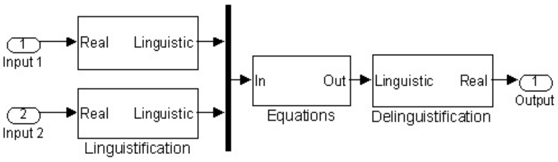

Fig. 2 A steady state LE model for two inputs and one output.

A membership definition is a nonlinear mapping of the variable values inside its range to a certain linguistic range, usually $[-2, 2]$. The mapping is represented with two monotonous, increasing functions,

$$\begin{array}{ll} f_j^- = a_j^- X_j^2 + b_j^- X_j + c_j, & X_j \in [-2, 0), \\ f_j^+ = a_j^+ X_j^2 + b_j^+ X_j + c_j, & X_j \in [0, 2], \end{array} . \qquad (2)$$

which must overlap in the center at the linguistic value 0. In the present system, these functions are second order polynomials. Coefficients are extracted from data or defined on the basis of expert knowledge. [18]

In the LE models, the nonlinear scaling is performed twice: first scaling from real values to the interval [-2, 2] before applying linguistic equations, and then scaling from the interval [-2, 2] to real values after applying linguistic equations (Fig. 2). The linguistic level of the input variable j is calculated the inverse functions of the polynomials (Eq. (2)).

The model is represented by

$$x_{out} = f_{out}\left(- \frac{\sum_{j=1, j\neq out}^{m} A_{ij}\, f_j^{-1}(x_j) + B_i}{A_{i\ out}} \right) \tag{3}$$

where the functions f_j and f_{out} are membership definitions.

Real-valued approach is now the main method in applications because of efficient tuning techniques. A neural network based tuning can be done for selected variables. A genetic tuning method can handle several variables at a time by varying parameters of membership definitions.

For case based model development, the training data consist of several data sets. Fuzzy C-Means Clustering is used for finding these overlapping operating areas [13]. Alternatively the operating areas can be obtained by subtractive clustering, self-organizing maps (SOM) or radial basis functions. The interaction matrix is normally the same for all working areas, which is quite reasonable since the directions of interactions do not change considerably between different working points. The differences between the models are handled with membership definitions.

The modelling technique can be extended to several equations as well, e.g. by using Takagi-Sugeno (TS) type fuzzy models [19] together with ANFIS method [20] for development of local linear models for different operating areas. As *LE* models are nonlinear, also these local models are nonlinear [9]. In these models, fuzzy approach is used for combining gradually changing interaction coefficients.

Steady-state models represent interactions between alkali, lignin and dissolved solids. A single equation model shown in 3 was very accurate but test campaign specific [9]. The retuning was necessary because of drastic changes in the operating conditions, especially in properties of the cooking liqueur. Two equation models based modified TS-models had similar a similar performance. Later only one model area has been used since the operating conditions have been stabilized.

3.2 Dynamic modelling

The basic form of the linguistic equation (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. External dynamic models provide the dynamic behaviour. The models are developed for a defined sampling interval in the same way as in various identification approaches [21].

Rather simple input-output LE models, where 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. Nonlinear scaling reduces the number of input and output signals needed for modelling of nonlinear systems.

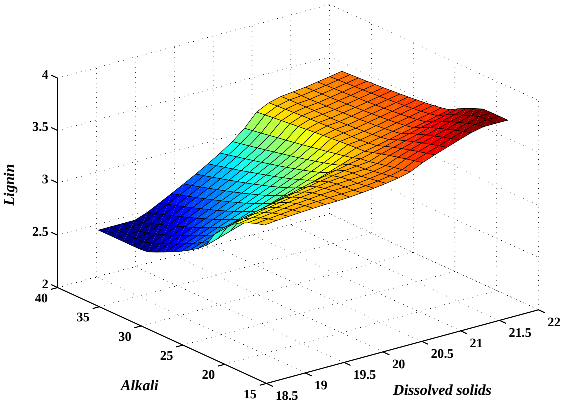

Fig. 3 A single equation *LE* model for alkali, lignin, and dissolved solids [15].

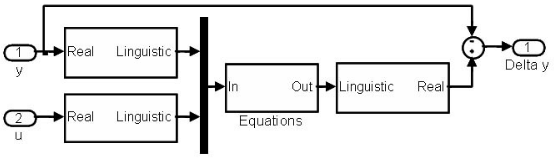

Fig. 4 Dynamic LE model of Δy.

Need for higher order models can be tested by applying classical identification with different polynomial degrees to the data after scaling with membership definitions. For the default LE model, all the degrees of the polynomials in parametric models become very low, i.e. all the parametric models become the same

$$y(t) + a_1 y(t-1) = b_1 u(t - n_k) + e(t). \tag{4}$$

This model is a special case of Eq. (1) with three variables, $y(t), y(t-1)$ and $u(t-n_k)$, the interaction matrix $A = [1 \;\; a_1 \;\; -b_1]$ and the bias term $B = 0$.

For small systems, delays can be taken into account by moving the values of input variables correspondingly, e.g. n_k sampling intervals in Eq. (4). Initial estimates of the delays can be developed by correlation analysis, but similarities detected by the correlation analysis can be accidental in some cases. The delays should be assessed against process knowledge, especially if normal on-line process data is used [8]. An appropriate handling of delays extends the operating area of the model considerably.

Local time delays can be analysed with different parametric models used in system identification: the number of delay n_k in Eq. (4) is the effective time delay.

Effective delays depend on the working conditions (process case), e.g. the delays are closely related to the production rate in many industrial processes. In the block shown in Fig. 5, the delay of the variable *Var1* depends on the variable *Var2*: the linguistic level of the

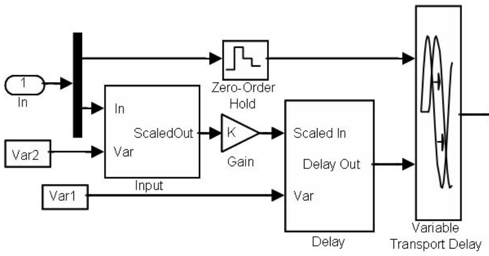

Fig. 5 Variable delay.

variable *Var2* is multiplied by *1* or *-1* to get the linguistic level of the delay for the variable *Var1*, coefficient *1* means that the delay increases when the variable *Var2* increases. The real value of the delay is obtained by the delinguistification block.

These models are used for calculating the change Δy for a sampling interval (Fig. 4): the linguistification and delinguistification blocks are the same as in Fig. 2. The membership definition of the variable y does not depend on time. The membership definitions have captured the real meaning of the variables if higher degrees do not improve the model considerably. An alternative approach is to make membership definitions also for the change Δy and obtain the derivative directly from the corresponding LE model.

The output, the derivative of the variable y, is integrated with numerical integration methods:

$$y = \int_0^T F(t, y, u)dt + y_0 \qquad (5)$$

where T is the time period for integration, and y_0 *the initial condition*. Usually, several values from the integration step or the previous steps are used in evaluating the new value. Step size control adapts the simulation to changing operating conditions.

In batch cooking, the dynamic models of alkali, lignin and dissolved solids are interacting. The overall model is a set of three equations, in the present these equations are used for calculation of changes for alkali, lignin and dissolved solids. For example the alkali concentration is decreasing during the cooking, and the change in absolute figure is decreasing with increasing H-factor and decreasing alkali level as shown in Fig. 6. For batch cooking, each equation is represented by a model block (Fig. 7).

In batch cooking, specific submodels were needed because of variations in the quality of chips and the properties of the incoming cooking liquor [15]. As the differences in operating conditions affect to the speed of change, the models were adapted with a speed factor that depends on the H-factor and alkali level. The clustering approach could be used if the properties of the chips and incoming cooking liquor are measured. Since these measurements are not available, the subareas must be identified by comparing the performance of the normal model. Later only one model area has been used since the operating conditions have been stabilized.

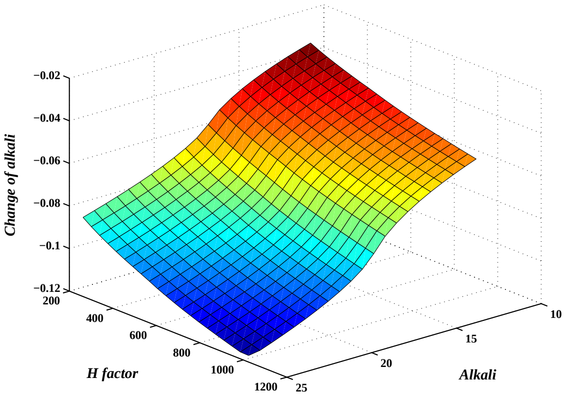

Fig. 6 Model surface of the alkali model [15].

3.3 Fuzzy simulation

Universal approximators for fuzzy functions can be constructed as extension principle extensions of continuous real-valued functions which continuously map fuzzy numbers into fuzzy numbers [22, 23]. LE models can extended to fuzzy inputs with this approach if the membership definitions, i.e. functions f_j^- and f_j^+ and the corresponding inverse functions, are replaced by corresponding extension principle extensions of these functions.

The argument of the function f_{out} in Eq. (3) is obtained by Fuzzy arithmetics. Only addition and subtraction are needed if the interaction coefficients are crisp. Fuzzy LE models with fuzzy inputs can be constructed by using multiplication and division as well. Fuzzy extension of the classical interval analysis [24] suits very well to these calculations.

4 Dynamic simulation

The steady-state model developed for interactions between alkali, lignin and dissolved solids was first included to a dynamic model, i.e. lignin and dissolved solid were used in the calculation of new alkali level [9]. The dynamic model shown in Fig. 7 includes two additional models: one for calculating the change of lignin, and one for calculating the change of dissolved solids. The structure of all these submodels was similar, and they were developed from the measurement data on several overlapping operating areas. The dynamic effects were handled with integration of changes.

Multimodel aspects were taken into account by adapting the model to different operating conditions with a speed factor that depends on the H-factor and alkali level. The level of the speed factor is modified to get the simulated results close to the measured values (Fig. 8). Each submodel had its own model for the speed factor.

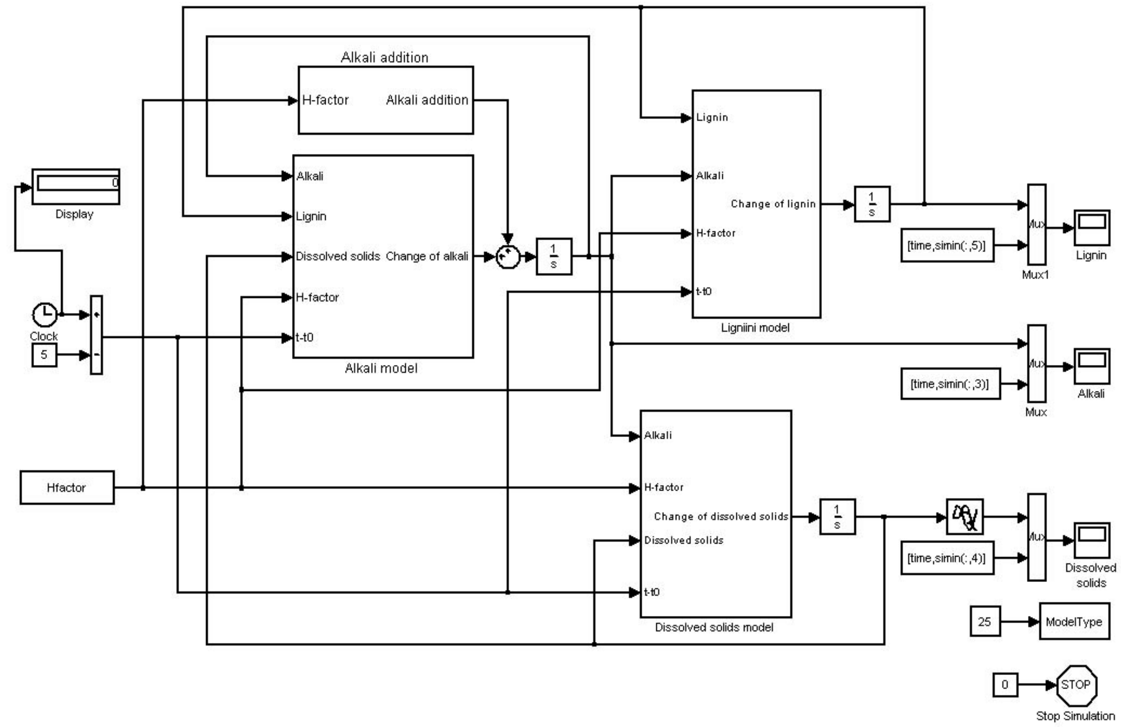

Fig. 7 Simulator [15].

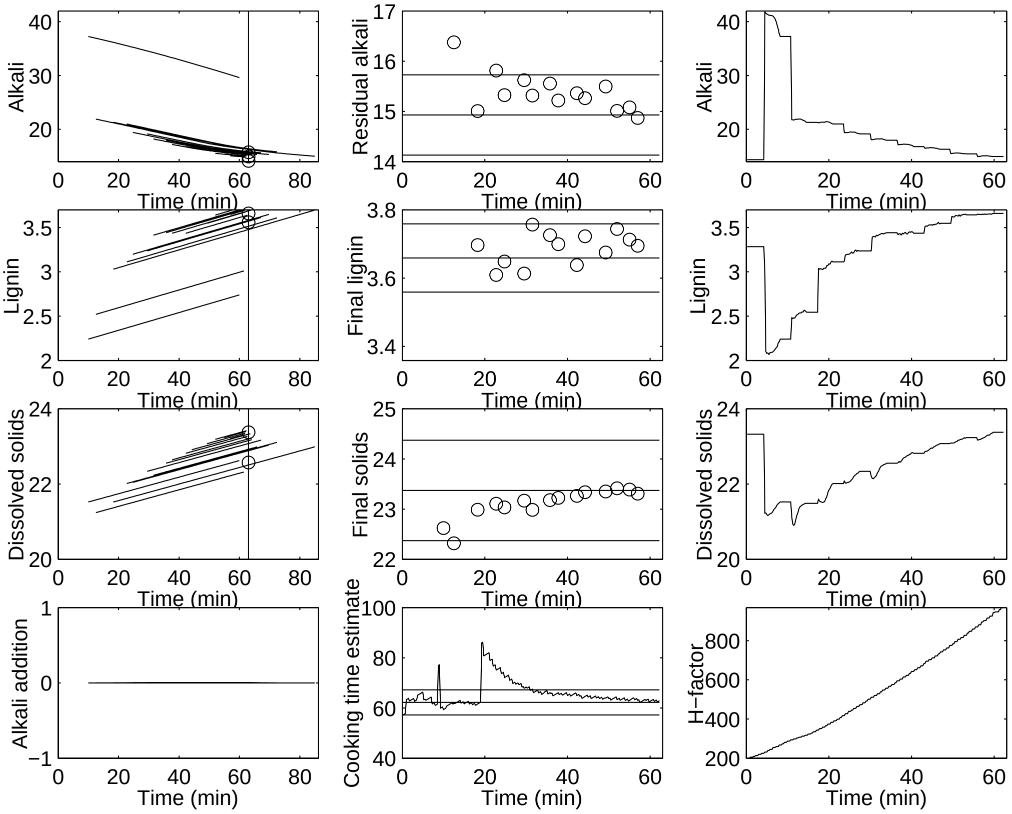

Fig. 8 Test results of the simulator [15].

The simulation program was used to test the modelling performance. All the submodels were built with special blocks implemented in *Matlab-Simulink*. Simulation results were shown in the same display together with the real data values (Fig. 8). In this way, the real error between simulated and batch data was clearly seen. In the simulation, the integration methods were principally Runge-Kutta and Ode15s(stiff/NDF) with variable step size.

According to the first tests, the original model structure operated in a reasonable way [10]. Each test campaign required a new tuning for the membership definitions as each operating area was always very different from previous ones. However, the parameters generated on the basis of one day on each campaign operated fairly well for all the days of that specific campaign. The results of the test campaigns have been clustered into data sets corresponding to different operating conditions.

The analysis of the on-line operation changed the submodels completely, although the high level model shown in Fig. 7 remained unchanged [15]. The LE model shown in Fig. 3 was not a feasible model in this case as there were all the time considerable differences in the starting levels of both the lignin and the dissolved solids. First lignin and dissolved solids were excluded in the calculation of the new alkali level, and the simulation was based direct calculation of the alkali change. Later all the simulations have been based on calculating the changes directly.

In most cases predictions are fairly accurate already 20-25 minutes from the beginning of the cooking phase. As fluctuations related to the quality of chips or cooking liqueur can be seen within first 30 minutes, there are possibilities to make alkali additions if required.

In the present simulator, uncertainty is taken into account by repeating the calculations in a fuzzy framework based on the extension principle. Uncertainty of the results is clearly seen when the resulting residual alkali, final lignin and final solids are presented as fuzzy numbers. The width of these fuzzy numbers depends on the length of the calculation period, i.e. the fuzziness of the result is smaller for the simulations performed later during the cooking sequence. This is reasonable since also the accuracy of the prediction is improved when the cooking progresses as can be seen in Fig. 8.

5 On-line simulator

The dynamic simulator of the batch cooking has been in on-line testing at an industrial digester house. The simulator was first tested with previous measurements. To obtain several estimates, the simulation is done to the end of cooking sequence several times during cooking process [15]. A result window similar to Fig. 8 is updated after each simulation run.

The measurements are not continuously updated all the time as the analyser needs a washing sequence. Therefore, the data set must be checked before starting the simulation. To emulate these features in the off-line tests, the feeding of the data was done with another *Matlab* program running simultaneously. The off-line tests are very fast: one cooking sequence takes some minutes and most of the time goes for showing the intermediate results similar to the on-line operation.

According to the test results, the speed of the change depends strongly on the operating conditions. For the first test periods, differences were too high for a reliable prediction, but later the speed factor did not need any changes for most of the cases, i.e. changes of the speed factor level mean considerable differences in the quality of chips.

The on-line simulator calculates every time the values of alkali, lignin and dissolved solids to the end of the cooking sequence on the basis of the *CLA 2000* measurements and various process measurements collected by the Damatic automation system. The simulation is started on appropriate time intervals. Fig. 8 shows an example of the simulation results in the end of the cooking sequence: simulations and in the first column, predictions of the end result in the second, and actual measurements in the third. Predictions are shown after each simulation.

Good results at the early stage of the cooking sequence are promising for extending the use of the *CLA 2000* analyser and the on-line simulator to several digesters. Three periods with several measurement sequences would be needed for each digester to be able to get first predictions, quality tests and final levels in the end of the cooking phase. More frequent simulation runs do not cause problems since the simulator operates fast. Four digesters could be handled with one cooking liqueur analyzer.

The on-line simulator of the cooking process is planned to be used in the control of alkali additions during the cooking sequence on the basis of the estimates of the residual alkali. The other predictions are planned to be used forecasting the quality of the pulp, e.g. the well-known Kappa number or a related property. In the future the aim is also to tune the controllers of the whole cooking process.

The fuzzy LE simulator is now in the simulation test stage. As the computer resources required for the extended version are only slightly higher than the deterministic version, the new version should be feasible for on-line use. However, additional tests with the extensive set of measurements will be continued before starting the on-line use.

6 Conclusions

Linguistic equations have been applied in the dynamic simulation of the batch cooking process. The dynamic *LE* model handles interactions between the values of alkali, lignin and dissolved solids and can be adapted to different operating conditions with configurable parameters. The graphical tools of *FuzzEqu* and *Simulink* have been essential in testing the quality of the dynamic *LE* models. Predicted alkali concentrations are directly useful, and predictions of lignin and dissolved solids can be used in assessing the pulp quality. Results are promising for control applications in the digester house. Extension to fuzzy LE models provides useful information about uncertainties of the forecasted cooking results. The complexity of the models is increased only slightly with the new system based on the extension principle and fuzzy interval analysis.

7 References

[1] K. Leiviskä. Process control in fiber line. In K. Leiviskä, editor, *Process Control. Papermaking Science and Technology, Book 14*, pages 73–99. Fapet Oy, Jyväskylä, 1999.

[2] R. R. Gustafson, G. G. Jiménez, and W. T. McKean. Theoretical model of kraft pulping process. *Industrial Engineering & Chemistry Process Design & Development*, 22(1):87, 1983.

[3] J. F. Saltin. A predictive dynamic model for continuous digesters. In *Proceedings of the Pulping conference*, pages 261–268. Tappi, 1992.

[4] S. Mirams and K. L. Nguyen. Application of predictive kinetic models in eucalypt kraft pulping. In *Proceedings of the International Pan Pacific conference*, pages 73–82. Tappi, 1994.

[5] H. C. Aguiar and R. M. Filho. Neural network and hybrid model: a discussion about di!erent modeling techniques to predict pulping degree with industrial data. *Chemical Engineering Science*, 56:565–570, 2001.

[6] R. Sutinen. New analyzer for more accurate cooking liquor measurements. *ABB Review*, (6):39–46, 1999.

[7] K. Leiviskä, E. Juuso, and A. Isokangas. Intelligent modelling of continuous pulp cooking. In K. Leiviskä, editor, *Industrial Applications of Soft Computing*, pages 147–158. Springer, Heidelberg, 2001.

[8] E. K. Juuso. Fuzzy control in process industry: The linguistic equation approach. In H. B. Verbruggen, H.-J. Zimmermann, and R. Babuska, editors, *Fuzzy Algorithms for Control, International Series in Intelligent Technologies*, pages 243–300. Kluwer, Boston, 1999.

[9] E. K. Juuso and E. Jäväjä. Intelligent dynamic simulation of batch cooking. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET'01 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 19-20, 2001*, pages 118–129, Oulu, 2001. Oulun yliopistopaino.

[10] E. K. Juuso. Intelligent dynamic simulation of batch processes. In *Proceedings of SIMS 2001 - 42nd Conference on Simulation and Modeling, Porsgrunn, Norway, October 8-9, 2001*, pages 155–163, 2001.

[11] U. Saarela, E. Juuso, and K. Leiviskä. *Modelling of a fed-batch fermentation process. Report A 21, June 2003.* Control Engineering Laboratory, University of Oulu, Oulu, 2003.

[12] E. K. Juuso. Modelling and simulation with intelligent methods. In *White paper of the Virtual Institute for Simulation (Sim-Serv): www.sim-serv.com*. Sim-Serv, 2004. 17 pp.

[13] E. K. Juuso. Linguistic equations for data analysis: Fuzzequ toolbox. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET 2000 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 13-14, 2000*, pages 212–226, Oulu, 2000. Oulun yliopistopaino.

[14] E. K. Juuso, J. Myllyneva, and K. Leiviskä. Fuzzy logic controller and adaptive tuning. In A. Guasch and R. M. Huber, editors, *Proceedings of the Conference on Modelling and Simulation 1994 Barcelona, Spain, June 1 -3, 1994*, pages 475–479, San Diego, USA, 1994. SCS International.

[15] E. K. Juuso. Intelligent dynamic simulation of batch cooking. In *44th Scandinavian Conference on Simulation and Modeling SIMS 2003, September 18 -19, 2003*, pages 163–169. Malardalen University, Västeras, Sweden, 2003.

[16] S. Murtovaara, K. Leiviskä, E. Juuso, and J. Ikäheimonen. Kappa number prediction based on cooking liquor measurements. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET'99 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 15-16, 1999*, pages 210–216, Oulu, 1999. Oulun yliopistopaino.

[17] E. K. Juuso and K. Leiviskä. Adaptive expert systems for metallurgical processes. In S.-L. Jämsä-Jounela and A. J. Niemi, editors, *Expert Systems in Mineral and Metal Processing, Proceedings of the IFAC Workshop, Espoo, Finland, August 26-28, 1991, IFAC Workshop Series, 1992, Number 2*, pages 119–124, Oxford, UK, 1992. Pergamon.

[18] E. K. Juuso. Integration of intelligent systems in development of smart adaptive systems. *International Journal of Approximate Reasoning*, 35:307–337, 2004.

[19] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Syst., Man, & Cybern.*, 15(1):116–132, 1985.

[20] J.-S. R. Jang. Anfis: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.

[21] L. Ljung. *System Identification - Theory for the User*. Prentice Hall, Upper Saddle River, N.J., 2nd edition, 1999.

[22] J. J. Buckley and Y. Hayashi. Can neural nets be universal approximators for fuzzy functions? *Fuzzy Sets and Systems*, 101:323–330, 1999.

[23] J. J. Buckley and T. Feuring. Universal approximators for fuzzy functions. *Fuzzy Sets and Systems*, 113:411–415, 2000.

[24] R. E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.