RapOpt - An Automation Tool for Production-Orientated Run-to-Run Model Evolution

Norman Violet, Niko Rossner, Thomas Heine, Rudibert King
Berlin Institute of Technology
Chair of Measurement and Control
Sec. ER2-1, 10623 Berlin, Hardenbergstrasse 36a, Germany

Corresponding author: Rudibert King
Rudibert.King@TU-Berlin.DE

Abstract. Mathematical models have been proven to be a key factor in optimizing production processes in recent years. However, in the case of biochemical processes the design is usually done using heuristics, since these systems show complex internal regulation mechanisms and strongly nonlinear behavior. This makes it difficult to find an appropriate model. In those cases, where a structured biochemical model has been successfully identified, the yield of the process can be increased significantly. Obtaining a suitable production model is usually a difficult and time consuming task, especially for biochemical systems. In this contribution the concept for an automation tool is presented which starts with the few noisy measurements of initial experiments to perform a model evolution from run to run. Thereby, the first unstructured model candidates are used for an optimal production-orientated process design whose realization will provide additional information about the dynamic behavior within the production area, thus, leading to new and improved model candidates. Due to the difficult measurement situation in biochemical processes many different model candidates may show a similar fit to the data why it is unwise to focus on one model candidate for process design, only. Furthermore, the use of more than one model candidate for the design procedure represents a kind of robustness for the planning. This cyclic procedure enables an optimal production design corresponding to the available measurement information at any time.

1 Introduction

Mathematical models of biological productions play an important role for process planning and optimization [1]. Here, the main task of a model is the prediction of optimal substrate feeds in order to maximize the economical yield. Usually a human modeler will plan a number of experiments using his or her experience and heuristics. To obtain a mathematical description of the system the trends of the measurements are analyzed manually and the most important state variables and reaction schemes are postulated. Then a mathematical model is formulated using balance equations and conservation laws. The velocity of each reaction step has to be described using empirical kinetic equations. After the model implementation the values of the model parameters have to be determined in a time consuming optimization-based numerical identification. In an iterative way, the human modeler changes the reaction schemes and the kinetic terms until the model shows an appropriate fit to the experimental data. Because of this tedious procedure not all promising reaction kinetics will be tested. Thus, there might exist many other different model structures which would fit the few existing measurements similarly well or even better. After the identified model was used to plan new feeding profiles, often a significant extrapolation outside the domain of identification experiments takes place. Here, this model is often no longer valid. Therefore, the modeling procedure has to be repeated. The result of this error-prone and time consuming scheme is highly depending on the expertise of the human modeler.

In the last decade many software tools have been presented to simplify the modeling procedure [2]. Commercial tools like AspenPlus^{TM}, ChemCAD^{TM} or gProms^{TM} are usually highly specialized on a certain field of application and rely on established methods, while academic institutions rather use prototypic realizations of new approaches. They often focus on the automation of major modeling steps [3, 4, 5, 6, 8, 9, 10, 11]. Besides balance equations with reaction kinetics, many recent tools like Simpathica [12, 13], TAM-B [9, 11], ProMoT [14, 7] and BioChem [20, 19] also consider temporal logic in order to integrate information from heuristic observations in the mathematical model. While TAM-B uses this information to refine the reaction scheme of an ODE system, BioChem uses the temporal logic to describe additional constraints for a canonical S-System [21]. The tool ProMoT is based on network theory and provides a graphical interface with *drag&drop* functionality, which allows to quickly build a model out of standard elements stored in a library. It offers different input and output standards, providing a wide variety of interfaces for further processing. The software tool RapOpt [22] presented in this paper focuses on a data-driven continuous model evolution starting with the measurements from the first experiments. In order to test the fitting of different models, RapOpt will interchange individual kinetics within a given basic structure,

automatically code and compile the model files and thus create a multi-model system environment. The refinement of each model in every iteration cycle is orientated towards product maximization.

The paper is organized as follows. In section 2 the progress of the run-to-run model evolution will be introduced in general, whereas only the central functionalities are described in this contribution. The final section is devoted to an example of a multi model process design and its experimental realization is presented.

2 Run-to-Run model evolution with RapOpt

2.1 Definition of a Model Family

In order to initialize RapOpt, the user has to define the system's states that should be considered. For the first crude, unstructured model it is assumed that all substrates may influence the reaction rates of the specified states. Additionally, measured data is required which can either be obtained from initial experiments in Erlenmeyer flasks and/or from the database of a process control system of a fermenter. Furthermore, the user has to define those dependencies in the reaction rates which are supposed to be interchanged by RapOpt as well as the permitted kinetics for this process (see Figure 1). Choosing all dependencies as changeable will cause a huge number of model candidates as it will be explained later in this section. At this point, the model family is completely defined and the investigation of its individual members concerning the available data will follow. To clarify the definition of a model family, a short example is introduced:

The growth of the biomass m_x is an autocatalytic process whose specific growth rate μ_x is influenced by three substrate concentrations c_{S1}, c_{S2} and c_{S3}. The balance equation of a (fed-)batch fermentation without cell death then reads

$$\dot{m}_x = \underbrace{\mu_{max}\, g_1(c_{S1})\, g_2(c_{S2})\, g_3(c_{S3})}_{\mu_x}\, m_x \tag{1}$$

In this basic structure of the model family, the reaction rate μ_x is a product of the three kinetics g_1, g_2 and g_3 depending on the different substrates c_{S1}, c_{S2} and c_{S3}. The a priori unknown kinetics for the specific growth rate - and analogously of every other unknown reaction rate - can be described using empirical kinetic expressions as shown in Figure 1. Besides the name of the kinetic and the mathematical expression, the library also contains meaningful initial values for the parameters in order to guarantee a typical behavior during simulation. Moreover, minimal and maximal values are given. These avoid the degeneration of kinetics. To create all members of a model family, all interchangeable kinetic terms will be permutated automatically by RapOpt using the list of the permitted kinetics, beginning with the least parameterized terms. In order not to create senseless models, a simple logic is implemented that for example avoids the use of strictly inhibiting kinetics in a growth rate when substrate is the dependency. Methods from TAM-B [9, 18, 11] shall be used to eliminate inappropriate models in future. Nevertheless, the kinetic library contains more than 50 different empirical expressions to describe building rates, which can lead to a huge number of models. In the example, see eq. (1), $50^3 = 125\,000$ candidates for μ_x can be generated, which have to be identified.

In the case that only very few measurements are available initially, the tool just activates the three most often used kinetic terms which contain at most two parameters to create different initial model candidates. In most cases the few initial measurements can be fitted similarly well with different kinetics, even if some reaction rates contain only one parameter. All created models constructing the model family and their corresponding parameter files will be coded automatically by RapOpt in a MATLAB m-file for easy interpretable documentation as well as coded and compiled in *C* for accelerated simulation. A short *select_model* command allows the user to change between different models, whereby a multi-model environment can be easily embedded in existing MATLAB programs which will be detailed below.

2.2 Parameter identification for all members of a model family

As pointed out in the previous section, the permutation of kinetics can lead to a large number of model candidates, for each of which the parameters have to be identified in a nonlinear optimization procedure. The numerical burden for a nonlinear parameter identification depends on the degree of the nonlinear couplings, the optimization algorithm, and on the quality of the initial values and measured data. In RapOpt, the time-consuming calculations for all model candidates are accelerated using three short-cut methods beside a multiple shooting approach [17].

- **Sequential Parameter Identification**
  The dependencies among the design parameters can be cut off with a sequential identification procedure. Normally in an identification, the ODE system has to be simulated to calculate the value of the maximum likelihood (ML)-objective. In a nonlinear system, each equation of the ODE system is usually coupled to many other equations. By using interpolated measurements instead of the simulations for all measured states, these equations can be decoupled, such that a parameter in one equation does not affect the results of other equations. Thus, the identification problem is partitioned in a series of identifications. The first problem of this sequence only

name	kinetic term w/o μ_{max}	parameter min	parameter max	parameter initial
Monod	$\frac{c_s}{c_s+k}$	k=0.001	k=50	k=0.75
Ming	$\frac{c_s^2}{c_s^2+k}$	k=0.001	k=1000	k=1
Aiba	$\frac{c_s}{c_s+k}\ e^{-\frac{c_s}{k}}$	k=0.05	k=20	k=1
Sokol Howell	$\frac{c_s}{c_s^2+k}$	k=0.02	k=10	k=1
Haldane	$\frac{c_s}{c_s+k_m+\frac{1}{k_i}c_s^2}$	k_m=0.001 k_i=0.001	k_m=100 k_i=100	k_m=1 k_i=0.1
Moser	$\frac{c_s^\lambda}{c_s^\lambda+k}$	k=10^{-3} λ=0.2	k=1 λ=25	k=0.5 λ=1

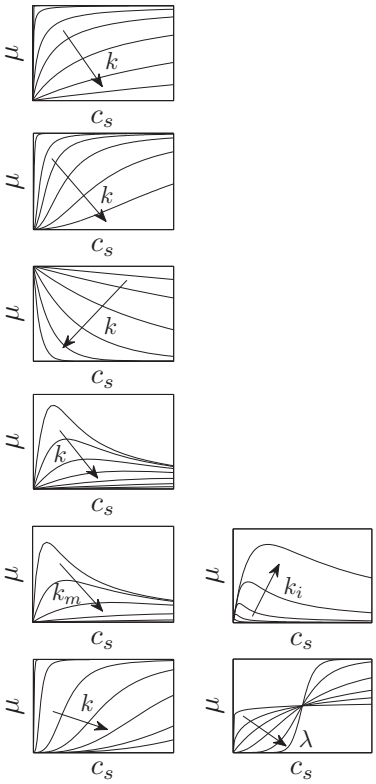

Figure 1: Examples of commonly used rate equations for biological systems that are stored in the kinetic library without the corresponding maximal specific growth rate μ_{max}. The library also contains information about the parameter bounds and the initial parameter values. To the right, a set of curves is shown to illustrate the shape of the kinetic when varying its parameter(s).

contains a few design parameters. By replacing the data interpolations with the model simulations step by step, the forthcoming identifications grow piecewise until the original identification problem is solved.

- **Determination of Initial Values**
  The sequential identification is used for the first model candidate only and provides a well fitted initial model. For all further model candidates the similarity between the different models is used to generate initial parameter values for the next identification. This presumes a designated order, in which consecutive models only differ in one kinetic term. The parameters of the current identification are initialized with the optimal parameters of the former model which ensures a converging identification procedure. The new parameters of the changed expression (e.g. Figure 1) are determined within the given bounds such that the kinetic term is as close as possible to that of the former candidate. For this process no time-consuming simulation of the ODE system is necessary.

  As an example, Figure 2 shows how some kinetics from Figure 1 are equalized to a Monod equation with given parameter $k = 1$ within the given range $0 \leq c_s \leq 1$. In RapOpt this range will be determined according to the experimental data used for parameter identification. The kinetic *Moser* is not shown in this figure, because by setting $\lambda = 1$ it can be exactly transformed in a Monod kinetic. As a result of this kinetic equalization, the initial simulation of the new model is very close to the final simulation of the previous model candidate.

- **Sequence of Identification**
  Depending on the number of model candidates, there might not be enough time to identify all of them. Therefore, most promising models have to be identified first. In RapOpt this is achieved using a hierarchical tree structure. The top level consists of the model candidate with the most simple kinetic terms (usually all *Monod*) as a parent for further variations. The second layer is derived from the first by replacing one dependency with each of the activated reaction equations and thus creating several children. Therefore, in the appearing tree structure, adjoining models along the branch differ in one term only. Moreover, all children of a model only differ in one term as well. The following heuristic is used to choose the most promising model candidates for the next identifications. After the identification of the first model, all of its direct children will be identified. The child with the best fit to the measured data will be the new parent, whose children will then be tested.

name	parameter	
Monod	$k = 1$	
Ming	$k = 0.655$	
Sokol Howell	$k = 1.185$	
Haldane	$k_m = 0.988$	$k_i = 38.126$
Moser	$k = 1$	$\lambda = 1$

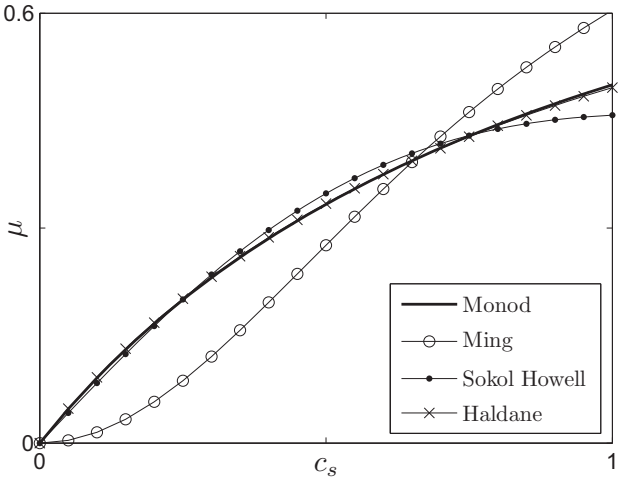

Figure 2: Parameters of different kinetics adjusted such that they fit the Monod kinetic ($k = 1$) in a least squares sense.

This identification procedure continues until all leaves of one branch of this systematic tree have been reached. At this point, every dependency was interchanged with all activated kinetics from the library even though not all permutation have been identified yet. Then, the procedure continues with the model in the data tree which shows the second best fit to the measurements, and so on. This strategy is based on the assumption, that a better model always arises from a good model by further changes in individual dependencies and therefore many promising models are identified at an early stage. However, the best model can be located somewhere in the tree. Still a process design with appropriate model candidates can be started already as explained next while the identification procedure continues to find even better models.

2.3 Multi-Model Trajectory Planning

At an early stage of process development only few measurement information is available. Many of the model candidates created in section 2.1 and 2.2 will fit the measurements similarly well with differences in the objective values in the order of the measurement noise. Nevertheless, they all have different structures, with different parameter sensitivities according to the measurements given. A design procedure that is based on the best model only, runs the risk of showing a bad extrapolation of the model behavior around the planned trajectory either caused by a wrong mathematical structure or by parameters that had been insensitive during identification and have now a significant effect in process design. Moreover, judging a model by its objective value is delusive, due to the fact that a gradient-based optimization could have stopped in a local minimum. Not rarely, a better objective can be found if the optimization is restarted at the last minimum, because of an untrained Hessian matrix. By considering more than one model in the planning procedure, these problems can be addressed. The extrapolation to an extreme dynamical behavior, that a single model could predict, is now partly covered by the others. Moreover, the difficulty of finding the best model is circumvented by optimizing the feeding profiles according to the yield predicted by all models. The simplest objective function for a multi-model trajectory planning would be to maximize the mean or median of the product amount. More robust trajectories can be obtained if the objective is formulated using the minimal product yield obtained over all models.

2.4 Preparing the next evolution

As more measurement information becomes available from run to run, two different evolutions can take place. At first, an automated data analysis will be done in order to investigate whether or not the ODE structure has to be refined by intra-cellular storages of nutrients and products to slowly obtain a more and more structured compartment model. If new states are postulated, the procedure will restart using the simplest kinetics. Otherwise, more complex kinetic terms will be tried out for the best models of the last cycle. The percentage of models that should be carried over to the next cycle can be investigated as follows:

Let m be the number of permitted kinetics of the last cycle and n the number of new kinetics for the forthcoming. If d is the number of dependencies wherein the kinetics are inserted, then m^d models were formerly considered and $(m+n)^d$ would have to be identified in the next cycle. Reusing a certain fraction X of old structures can reduce the number of models if

$$X \cdot m^d \cdot (n+1)^d < (m+n)^d \tag{2}$$

$$X < \frac{(m+n)^d}{(m \cdot (n+1))^d} \tag{3}$$

holds. It has to be observed that reusing models will lead to several identical models, when new kinetics are inserted for terms that formerly distinguished the models from each other. Therefore, a routine has to be implemented that eliminates all duplicated models. Nevertheless, this method promotes the actual evolution, since further modifications are based on the properties of the fittest models only.

3 Experimental Part

The development of the RapOpt software-tool for a run-to-run optimization was followed in parallel to the synthesis of an optimization, based on a multi-model approach. Therefore, the first experimental results obtained from a multi-model planning which is presented here, did not make use of all possibilities concerning automatic modeling as described above. Instead, some modeling steps had been done manually to describe the growth and production behavior of the bacteria *Paenibacillus polymyxa*, see below. Early experiments with this organism have pointed out that a simple unstructured model has to be refined by a storage term for phosphate, giving rise to the following low-structured model family.

$$\dot{m}_x = \mu_x \, m_x \tag{4}$$
$$\dot{m}_{am} = -Y_{am/x} \, \mu_x \, m_x + c_{am,feed} \, u_{am}(t) \tag{5}$$
$$\dot{m}_{ph} = -\left(Y_{ph/x} \, \mu_x + Y_{ph/pp} \, \mu_{pp}\right) m_x + c_{ph,feed} \, u_{ph}(t) \tag{6}$$
$$\dot{m}_c = -\left(Y_{c/x} \, \mu_x + Y_{c/ML} \, \nu_{pp} \, \mu_{ML} + Y_{main}\right) \, m_x + c_{c,feed} \, u_c(t) \tag{7}$$
$$\dot{m}_{pp} = \left(\mu_{pp} - Y_{pp/ML} \, \nu_{pp} \, \mu_{ML}\right) \, m_x \tag{8}$$
$$\dot{m}_{ML} = \mu_{ML} \, m_x \tag{9}$$
$$\dot{V}_x = u_{am}(t) + u_{ph}(t) + u_c(t) \tag{10}$$

$$\mu_x = \mu_{max,x} \quad g_1(c_{am}) \, g_2(c_{ph}) \, g_3(c_c) \tag{11}$$
$$\mu_{ML} = \mu_{max,ML} \; h_1(c_{am}) \, h_2(c_c) \; h_3(c_{pp}) \tag{12}$$
$$\mu_{pp} = \mu_{max,pp} \quad \text{ming}(c_{ph}) \, \text{aiba}(c_{pp}) \tag{13}$$
$$\nu_{pp} = \text{aiba}(c_{ph}) \tag{14}$$

The balance equations for biomass m_x and the product macrolactin m_{ML} contain variable unknown kinetics g_i, h_j that depend on the concentrations of the substrates glucose (index 'c'), ammonium ('am'), phosphate ('ph') and polyphosphate ('pp'). For each of these variable kinetics one of the three allowed kinetics *Monod*, *Moser*, *Ming* for the growth and *Monod*, *Haldane*, *Ming* for the production from Figure 1 was inserted by RapOpt to define an individual member of the model family. Since there are 6 variable kinetic terms in the model family and three different permitted kinetic terms, $3^6 = 729$ different models had been set up automatically. After the automatic computational implementation of all models had been completed as described in the previous section, a parameter identification for every single model had been carried out. The result of these identifications are shown in Figure 3 as a histogram, where the number of models with a certain amount of the objective Φ_{MLE} is given.

The histogram clearly illustrates that many models can describe the measurements with similar quality. As argued in the previous section, using the model with the best objective value ignores the fact that the measured data might not be informative enough to clearly discriminate one model from the others as well as the problem of local minima in the objective of the parameter identification. Nevertheless, it is obvious that a lot of model structures are not able to fit the underlying measurement information. The question remains how many of the suitable model candidates should be used for the upcoming process design. In this case, the best 4 model structures according to

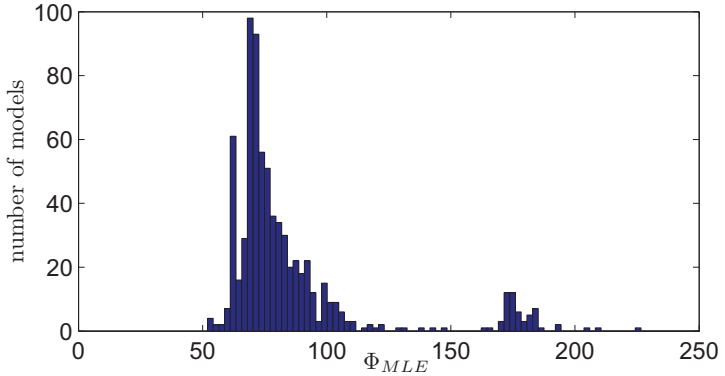

Figure 3: Histogram of the final ML-objective for the parameter identifications of all 729 models.

no.	growth					production					Φ_{ML}
	$\mu_{max,x}$	$g_1(c_{am})$	$g_2(c_{ph})$		$g_3(c_c)$	$\mu_{max,m}$	$h_1(c_{am})$	$h_2(c_c)$	$h_3(c_{pp})$		
1 *m.*		Ming	Moser		Ming		Monod	Ming	Sokol Howell		
	0.249	k=0.0158	k=0.088	λ=0.909	k=0.0059	0.005	k=0.0413	k=0.037	k=0.897		51.9
361		Monod	Moser		Monod		Monod	Monod	Haldane		
	0.347	k=0.786	k=0.095	λ=0.783	k=0.012	0.115	k=0.025	k=0.031	$k_m = 12.06$	$k_i = 0.018$	52.5
316		Monod	Moser		Monod		Ming	Monod	Haldane		
	0.335	k=0.724	k=0.091	λ=0.812	k=0.007	0.095	k=0.003	k=0.037	$k_m = 6.717$	$k_i = 0.018$	52.7
2 *m.*		Monod	Haldane		Monod		Ming	Monod	Haldane		
	0.403	k=0.470	k_m=0.078	k_i=1.150	k=0.003	0.110	k=0.002	k=0.041	$k_m = 9.433$	$k_i = 0.018$	54.2
3 *m.*		Monod	Haldane		Monod		Monod	Monod	Haldane		
	0.410	k=0.398	k_m=0.068	k_i=0.940	k=0.002	0.099	k=0.010	k=0.043	$k_m = 9.592$	$k_i = 0.020$	55.8
4 *m.*		Monod	Moser		Monod		Ming	Monod	Sokol Howell		
	0.326	k=0.594	k=0.100	λ=0.867	k=0.004	0.057	k=0.001	k=0.040	k=14.477		56.6
5 *m.*		Monod	Haldane		Ming		Ming	Monod	Haldane		
	0.398	k=0.379	k_m=0.068	k_i=0.970	k=0.001	0.099	k=0.001	k=0.051	$k_m = 10.75$	$k_i = 0.021$	57.9
568		Ming	Moser		Monod		Ming	Monod	Haldane		
	0.262	k=0.376	k=0.087	λ=0.544	k=0.001	0.090	k=0.019	k=0.041	$k_m = 10.27$	$k_i = 0.030$	59.0
424		Monod	Moser		Monod		Ming	Ming	Haldane		
	0.317	k=0.469	k=0.082	λ=0.826	k=0.004	0.105	k=0.005	k=0.048	$k_m = 11.06$	$k_i = 0.019$	60.3

Table 1: Inserted kinetic terms of nine model candidates that were used for multi-model trajectory planning. For the mathematical expressions of the kinetic terms see Figure 1. Model numbers denoted by *m.* had been built manually.

no.	ν_{pp}	μ_{pp}			yield coefficients						
	Aiba	μ_{max}	*Ming*	*Aiba*	$Y_{am/x}$	$Y_{ph/x}$	$Y_{ph/pp}$	$Y_{c/x}$	$Y_{c/main}$	$Y_{c/ML}$	$Y_{pp/ML}$
1 *m.*	0.2503	4.340	1.9254	2.9309	0.1848	0.0228	12.936	1.7753	0.0782	15.232	0.3716
361	0.0500	0.005	0.1090	22.095	0.1891	0.0468	10.237	1.9250	0.0626	19.048	0.1401
316	0.0883	0.025	0.1183	21.638	0.1897	0.0471	8.0091	1.8984	0.0642	19.104	0.1442
2 *m.*	0.0624	0.048	0.1113	22.034	0.1932	0.0475	10.753	1.6069	0.0642	21.728	0.1445
3 *m.*	0.0573	0.005	0.1099	22.450	0.1969	0.0484	11.778	1.8260	0.0702	18.953	0.1419
4 *m.*	0.0728	0.061	0.0834	19.965	0.1892	0.0467	13.746	1.6376	0.0639	21.134	0.1306
5 *m.*	0.0561	0.054	0.1104	22.682	0.1958	0.0487	11.705	1.8195	0.0700	19.046	0.1395
568	0.0616	0.061	0.0892	22.752	0.1868	0.0476	11.223	1.7779	0.0636	20.289	0.1318
424	0.0534	0.049	0.0940	22.308	0.1940	0.0488	12.344	1.6834	0.0701	20.000	0.1442

Table 2: Parameter values of those kinetics that had not been interchanged (eq.(13) and (14)) as well as the identified yield coefficients of all nine models. Model numbers denoted by *m.* had been build manually.

the objective value were added to the 5 manually built models that already existed. Hence, in this point we left the route given in section 2, as RapOpt missed some functionalities when the experiment was scheduled. In Table 1 the kinetics and their corresponding parameter values for the interchanged dependencies of these models as well as its objective value after parameter identification are listed. Table 2 shows the yield coefficients and the parameters of the kinetics that had not been interchanged.

During the identification of the parameters these 9 models showed a very similar outcome as shown in Figure 4 for one of the experiments used in the identification process. It points out that different model structures show a similar fit to the measurements after parameter identification. When using these models for a multi-model trajectory planning, the individual trajectories differ a lot, as can be seen in Figure 5 as grey solid lines. The underlying process design is based on an objective that maximizes the mean of all nine predicted products, shown by the black solid line. When the process was run (measurements shown as open circles) none of the models at this early state could fully describe the behavior. Especially the predicted yield, here the antibiotic macrolactin, differs among the candidates. However, the obtained yield of macrolactin was higher than the one obtained by our partners of the biological department in first tests and taking into account that macrolactin is a secondary metabolized product these first results are promising. Moreover, these data are now used in the aforementioned model evolution.

Furthermore it is noticeable that the simulated phosphate concentrations show differences between the individual model and a bad fit in the first part of the experiment. The range can be partly explained by the obviously different implementations of the assumed polyphosphate reactions which can not be fitted since there are no measurements available. Beside of that it is reasonable to assume that phosphate is involved in a reaction not covered by these early model candidates. The model evolution will take care of that later on. The difference between planned trajectories of c_{am} and the measurements at the end of the experiment can be explained by the violation of the glucose concentration constraint; the models were not able to cope with this condition. It has to be noted that the feeding container for the glucose substrate run dry during the last 10 hours of the experiment. This explains the feeding profile of u_c in Figure 5 and the violation of the lower boundary of c_c. The last part of the experiment was re-calculated in this Figure to reflect the realized feeding profile to allow a better comparison between planned trajectories and measurements.

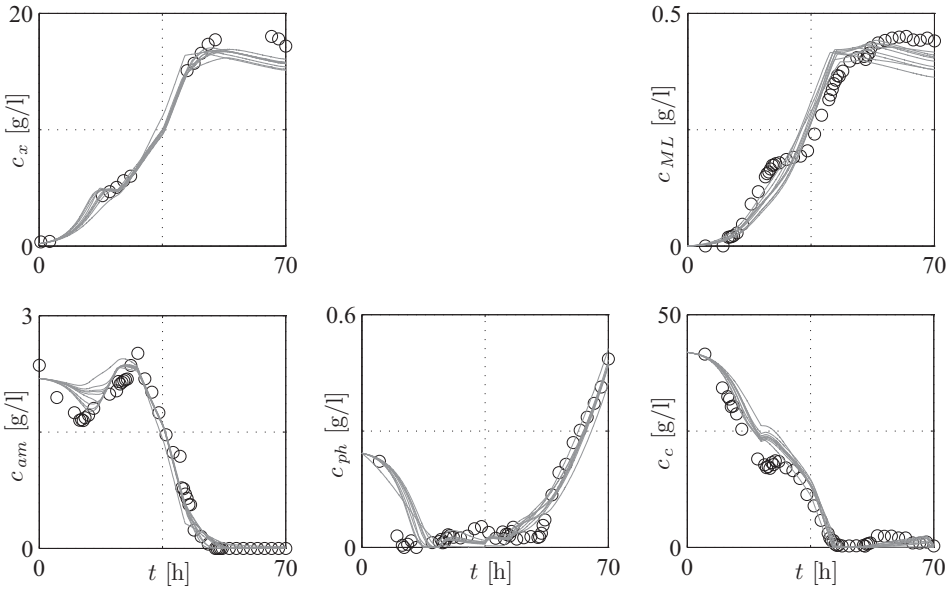

Figure 4: Simulation run of all nine models for one of the experiments used for parameter identification. Measurements of the concentration of biomass c_x, macrolactin c_{ML}, ammonium c_{am}, phosphate c_{ph} and glucose c_c are given by open circles, the different model simulations by grey solid lines.

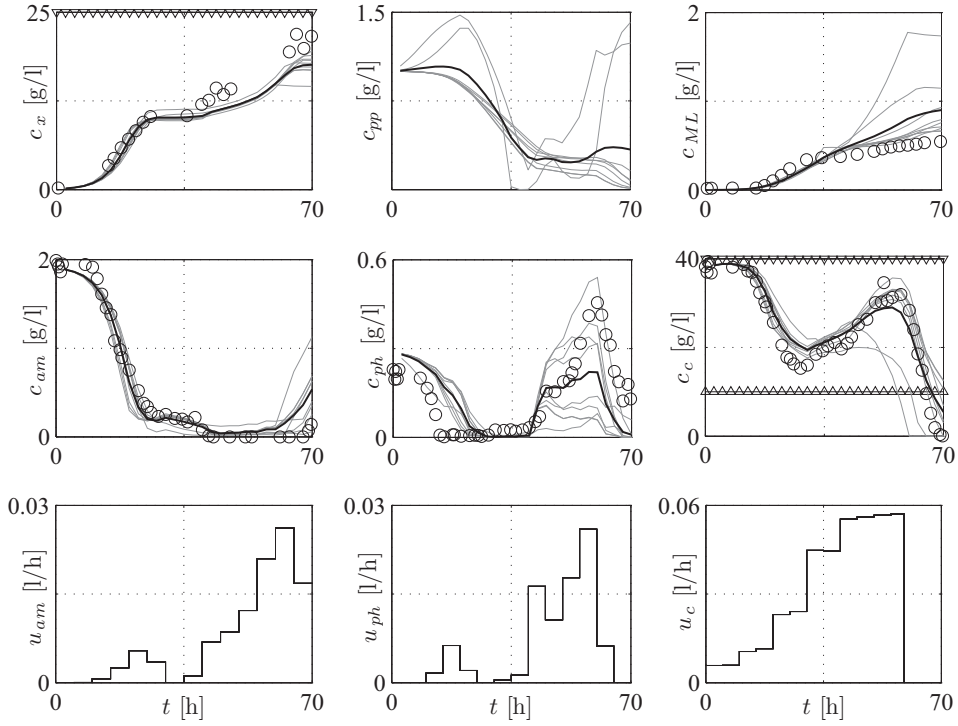

Figure 5: Optimized trajectories of the different models in a process design that considers all nine models. First and second row as in Figure 4. Third row: optimal feeding profiles used for ammonium, phosphate and glucose feed. Constraints are shown as lines with triangulars.

4 Conclusion

In this contribution, it was shown how RapOpt enables an automatic modeling based on a multi-model approach. Moreover, by connecting this automated modeling procedure with a multi-model process design an evolutionary model development was established that focuses from the beginning on the product output of the process. Therefore, the modeling procedure can be performed in parallel to the actual production process leading quickly to a certain amount of product while the model will be continuously refined.

Acknowledgment:
This work was partly supported by AIF, project 14775N. Additional financial support from the Deutsche Forschungsgemeinschaft and the Cluster of Excellence Unifying Concepts in Catalysis (EXC 314/1) are gratefully acknowledged.

5 References

[1] Roubos J.A.: *Bioprocess Modelling and Optimization: Fed-Batch Clavulanic Acid Production by Streptomyces*. PhD Thesis, Delft University of Technology, Netherlands, 2002.

[2] von Watzdorf R., Allgöwer F., Helget A., Marquardt W., and Gilles E.D.: *Dynamische Simulation verfahrenstechnischer Prozesse und Anlagen - Ein Vergleich von Werkzeugen*. In: Simulationstechnik 9. ASIM-Symposium, Wiesbaden, 1994, pp. 83–88.

[3] Bonvin D., and Rippin D. W. T.: *Target Factor Analysis for the Identification of Stoichiometric Models*. In: Chemical Engineering Science, Vol. 45, No. 12, pp. 3417–3426, 1990.

[4] Bettenhausen K. D., Marenbach P., Freyer S., Rettenmaier H., and Nieken U.: *Self-Organizing Structured Modelling of a Biotechnological Fed-Batch Fermentation by Means of Genetic Programming*. In: First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA, Vol. 414, pp. 481–486, 1995.

[5] Bogaerts Ph., and Vande Wouwer A.: *Systematic Generation of Identifiable Macroscopic Reaction Schemes*. In: 8th International Conference on Computer Applications in Biotechnology (CAB8), Quebec City, pp. 13–18, 2001.

[6] Bardow A., Marquardt W.: *Incremental and simultaneous identification of reaction kinetics: methods and comparison*. In: Chemical Engineering Science, Vol. 59, Issue 13, pp. 2673–2684, 2004.

[7] Saez-Rodriguez J., Mirschel S., Hemenway R., Klamt S., Gilles E. D., and Ginkel M.: *Visual setup of logical models of signaling and regulatory networks with ProMoT*. In: BMC Bioinformatics, Vol. 7, No. 506, 2006.

[8] Mandenius C.-F.: *Recent developments in the monitoring, modeling and control of biological production systems*. In: Bioprocess and Biosystems Engineering, Vol. 26, No. 6, pp. 347–351.

[9] Leifheit J., and King R.: *(Semi-)automatic modeling of biological reaction systems with TAM-B*. In: 9th International Conference on Computer Applications in Biotechnology (CAB9), Nancy, France, 2004.

[10] Mangold M., Angeles-Palacios O., Ginkel M., Kremling A., Waschler R., Kienle A., and Gilles E.D.: *Computer Aided Modeling of Chemical and Biological Systems: Methods, Tools and Applications*. In: Industrial & Engineering Chemistry Research, Vol. 44, No. 8, pp. 2579–2591, 2005.

[11] Leifheit J., Heine T., Kawohl M., and King R.: *Rechnergestützte halbautomatische Modellierung biotechnologischer Prozesse*. In: at Automatisierungstechnik, Vol. 55, Issue 5, pp. 211–218, 2007.

[12] Mishra B. *A Symbolic Approach to Modeling Cellular Behavior*. In Proceedings of HiPC 2002, pp. 725-732, Springer-Verlag, 2002.

[13] Antoniotti M., Policriti A., Ugel N., and Mishra B.: *Model Building and Model Checking for Biological Processes*. In: Cell Biochemistry and Biophysics, Vol. 38, No. 3, pp. 271–286, 2003.

[14] Ginkel M., Kremling A., Nutsch T., Rehner R., and Gilles E. D.: *Modular modeling of cellular systems with ProMoT/Diva*. In: Bioinformatics, Vol. 19, No. 9, pp. 1169–1176, 2003.

[15] King R., Leifheit J. and Freyer S.: *Automatic identification of mathematical models of chemical and biochemical reaction systems*. In: CHISA, pp. 495–510, Prague, Czech Rep., 2002.

[16] Pohlheim H., and Marenbach P.: *Generation of Structured Process Models Using Genetic Programming*. In: Lecture Notes in Computer Science, AISB Workshop on Evolutionary Computing, Vol. 1143, pp. 102–109, Springer-Verlag, 1996.

[17] Bock H.G., Diehl M.M., Leineweber D.B., and Schlöder J.P.: *A direct multiple shooting method for real-time optimization of nonlinear DAE processes*. Nonlinear Model Predictive Control, pp. 245–268, 2002.

[18] Leifheit J., and King R.: *Systematic structure and parameter identification for biological reaction systems supported by a software-tool*. In: 16th IFAC World Congress, Vol. 5, pp. 211–218, Prague, Czech Rep., 2005.

[19] Fages F., and Soliman S.: *Abstract interpretation and types for systems biology*. Theoretical Computer Science, Vol. 403, Issue 1, pp. 52–70, 2008.

[20] Calzone L., Chabrier-Rivier N., Fages F., and Soliman S.: *Machine learning biochemical networks from temporal logic properties*. In: Transactions on Computational Systems Biology VI, Vol. 4220, pp. 68–94, Springer-Verlag, 2006.

[21] Voit E. O.: *Canonical Nonlinear Modeling, S-system Approach to Understanding Complexity*. Van Nostrand Reinhold, New York, 1991.

[22] Violet N., Heine T., Rossner N., Valentin M., Böckelmann U., Szewzyk U. and King R. *Usage of a software tool for automatic modeling and application of the results to the design of fermentations with Paenibacillus polymyxa*. European BioPerspectives, 2008.