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™, ChemCAD™ or gProms™ 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 \( \mu_x \) 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

\[
m_x = \frac{\mu_{\text{max}} g_1(c_{S1}) g_2(c_{S2}) g_3(c_{S3})}{\mu_x} m_x
\]  

(1)

In this basic structure of the model family, the reaction rate \( \mu_x \) 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 permuted 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), \( 50^3 = 125000 \) candidates for \( \mu_x \) 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 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
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 \( \text{Moser} \) 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.
<table>
<thead>
<tr>
<th>name</th>
<th>parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monod</td>
<td>$k = 1$</td>
</tr>
<tr>
<td>Ming</td>
<td>$k = 0.655$</td>
</tr>
<tr>
<td>Sokol Howell</td>
<td>$k = 1.185$</td>
</tr>
<tr>
<td>Haldane</td>
<td>$k_m = 0.988$ $k_i = 38.126$</td>
</tr>
<tr>
<td>Moser</td>
<td>$k = 1$ $\lambda = 1$</td>
</tr>
</tbody>
</table>

**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

\[
X \cdot m^d \cdot (m + 1)^d < (m + n)^d
\]

or

\[
X < \frac{(m + n)^d}{(m \cdot (m + 1))^d}
\]
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.

\[
\dot{m}_x = \mu_x \cdot m_x \quad (4)
\]

\[
\dot{m}_{am} = -Y_{am/x} \cdot \mu_x \cdot m_x + c_{am,feed} \cdot u_{am}(t) \quad (5)
\]

\[
\dot{m}_{ph} = -\left(Y_{ph/x} \cdot \mu_x + Y_{ph/pp} \cdot \mu_{pp}\right) \cdot m_x + c_{ph,feed} \cdot u_{ph}(t) \quad (6)
\]

\[
\dot{m}_c = -\left(Y_{c/x} \cdot \mu_x + Y_{c/ML} \cdot \nu_{pp} \cdot \mu_{ML} + Y_{main}\right) \cdot m_x + c_{c,feed} \cdot u_c(t) \quad (7)
\]

\[
\dot{m}_{pp} = \left(\mu_{pp} - Y_{pp/ML} \cdot \nu_{pp} \cdot \mu_{ML}\right) \cdot m_x \quad (8)
\]

\[
\dot{m}_{ML} = \mu_{ML} \cdot m_x \quad (9)
\]

\[
\dot{V}_x = u_{am}(t) + u_{ph}(t) + u_c(t) \quad (10)
\]

\[
\mu_x = \mu_{max,x} \cdot g_1(c_{am}) \cdot g_2(c_{ph}) \cdot g_3(c_c) \quad (11)
\]

\[
\mu_{ML} = \mu_{max,ML} \cdot h_1(c_{am}) \cdot h_2(c_c) \cdot h_3(c_{pp}) \quad (12)
\]

\[
\mu_{pp} = \mu_{max,pp} \cdot ming(c_{ph}) \cdot aiba(c_{pp}) \quad (13)
\]

\[
\nu_{pp} = aiba(c_{ph}) \quad (14)
\]

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 \( \Phi_{MLE} \) 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.](image)
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.

<table>
<thead>
<tr>
<th>No.</th>
<th>( V_{PP} )</th>
<th>( h_{PP} )</th>
<th>Yield coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>( \mu_{\text{max}} )</td>
<td>( g_{1}(c_{\text{am}}) )</td>
<td>( g_{2}(c_{\text{pp}}) )</td>
</tr>
<tr>
<td>1 m.</td>
<td>0.2503</td>
<td>4.140</td>
<td>1.9254</td>
</tr>
<tr>
<td>361</td>
<td>0.0500</td>
<td>0.005</td>
<td>0.01090</td>
</tr>
<tr>
<td>316</td>
<td>0.0883</td>
<td>0.025</td>
<td>0.1183</td>
</tr>
<tr>
<td>2 m.</td>
<td>0.0624</td>
<td>0.048</td>
<td>0.1113</td>
</tr>
<tr>
<td>3 m.</td>
<td>0.0573</td>
<td>0.005</td>
<td>0.10999</td>
</tr>
<tr>
<td>4 m.</td>
<td>0.0728</td>
<td>0.001</td>
<td>0.08834</td>
</tr>
<tr>
<td>5 m.</td>
<td>0.0251</td>
<td>0.005</td>
<td>0.01184</td>
</tr>
<tr>
<td>568</td>
<td>0.0161</td>
<td>0.001</td>
<td>0.06892</td>
</tr>
<tr>
<td>424</td>
<td>0.0344</td>
<td>0.049</td>
<td>0.09400</td>
</tr>
</tbody>
</table>

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.

### 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.

### 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.
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