# Self-organizing Structured Modelling of a Biotechnological Fed-batch Fermentation by Means of Genetic Programming

**K.D. BETTENHAUSEN*** , **P. MARENBACH*** , **S. FREYER****, **H. RETTENMAIER**** and **U. NIEKEN****

**Darmstadt University of Technology , Institute of Control Engineering, Department of Control Systems Theory & Robotics, Landgraf-Georg-Strasse 4, D-64283 Darmstadt, Germany*

***BASF AG, D-67056 Ludwigshafen, Germany*

**Abstract.** The article at hand describes an approach for the self-organizing generation of models of complex and unknown processes by means of genetic programming and its application on a biotechnological fed-batch production.

**Key Words.** Genetic programming; modelling; system identification; biotechnology; predictive control.

## 1. INTRODUCTION

The natural biological metabolism of living organisms can be used for the production of food, medicines or basic materials for the chemical industries. The major task for optimization of those biotechnological processes is to overcome natural limitations by genetic manipulation of the organisms or variation of environmental conditions during the fermentation. This variation is part of the process engineers tasks and can be achieved by the use of advanced intelligent methods of control engineering.

Almost any approach for the design of control systems is based on a model of the process behaviour. In predictive control systems – fig. 1 –, which have proved to be especially suited for complex and non-linear plants, a model is even one essential part of the control loop. However, for biotechnological processes the classical way of systematic and empirical development of a suitable model – see e.g. (Moser, 1988) – is very difficult due to

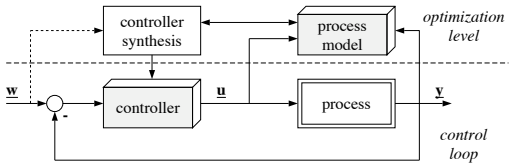


Fig. 1. Model based predictive control – see e.g. (Tolle and Ersü, 1992).

- the lack of quantitative process knowledge,
- the fact that most of the process variables relevant to the metabolism cannot be measured on line,
- the high amount of costs and time that is bound to the generation of experimental data and
- the variation of process behaviour with each different strain examined during process development.

Therefore classical control approaches use only constant setpoints for the whole process determined and judged during experiment series in laboratory scale – see e.g. (Bailey and Ollis, 1986) or (Präve *et al.*, 1987). Dynamic variations of the setpoints that provide a possibility of optimizing process performance are not used, due to the missing knowledge about how and when they have to be done.

Several approaches have been established in the last years to overcome these difficulties and to improve process control. Methods of parameter identification are used in adaptive control concepts – e.g. (Bastin and Dochain, 1990) or (Chen *et al.*, 1995) – based on the well known mass balances in the reactor. Instead of developing a model of their complex non-linear dependence on environmental conditions like temperature and pH, the time-varying specific growth rates are estimated and adapted on line. Since the system's inherent nonlinearities are interpreted by the time-varying parameters of the model and therefore an explicite representation of the mentioned dependencies does not exist, there is no chance to derive strategies for an optimal choice of setpoint profiles from the underlying model. However this can be achieved by models based on a learning ap-

proach which are capable of "remembering" the process behaviour at different operating points. The most common way to realize self-learning models are the so called neural networks. These neurally motivated memories are approximating non-linear manifolds by interpolating between information "learned" from a given training data set. It was shown in (Gehlen and Bettenhausen, 1990) and (Gehlen, 1993) that a process optimization based on a learning approach is possible and that it can lead to a significant increment of product yield. Actually Gehlen used an interpolating associative memory of the CMAC type (Albus, 1972), which is particularly suited for tasks of modelling. However there are still some disadvantages of the learning conception:

- Modelling of dynamic behaviour requires that a certain amount of process input and output history has to be considered at each timestep,
- a long term prediction leads to an error propagation due to recursive short-term prediction and
- a major problem of this approach is the missing transparency of the learned process information which cannot be visualized for an operator or the biological expert.

## 2. SELF-ORGANIZING GENERATION OF STRUCTURED MODELS

The disadvantage of missing transparency of neural models pointed out above, means that since the input/output behaviour is approximated by a black box approach no direct insight into the process and its underlying relationships can be gained. This usually leads to problems concerning the acceptance of these approches in industrial applications.

The new approach of the self-organizing generation of structured models by means of genetic programming described in this paper is an attempt to overcome these disadvantages. The general idea is to automate the iterative methodology of empirical modelling used by a process engineer. Therefore existing mathematical knowledge on structural properties of biochemical experts should be taken into account. Fig. 2 shows the basic scheme of self-organizing model generation, which is indeed very similar to the way models are developed by an process engineer – see e.g. (Moser, 1988). Starting with a collection of elementary transfer elements like time-delay or Monod kinetics placed in a so called "model construction set" a number of models are created. The degree of how much this is done by random depends on how much a-priori knowlegde is available. In an evolutionary process the following three steps are performed iteratively. First each model of a generation is

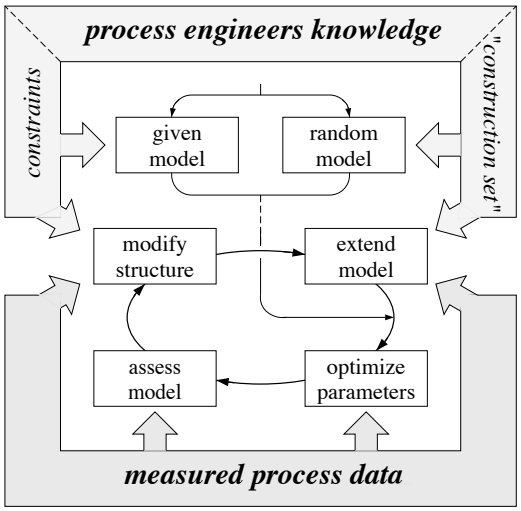


Fig. 2. Scheme of automatic generation of structured process models.

adapted to measured process data by optimizing its internal parameters, using well known parameter search methods – e.g. direct search (Hooke and Jeeves, 1961). After that a fitness value is evaluated for each model by assessing its accuracy and complexity. Directed by this fitness value new models are created by modifying and extending the actual model's structure. This iterative methodology which is imitating the principles of natural selection and reproduction (Holland, 1975) finally leads to models that combine high accuracy and low complexity, which are needed for most kinds of control purposes. A-priori knowledge on structural properties can be taken into account in this process by constraining the elements in the model construction set and by influencing their selection frequency. Furthermore elements can be combined to "super-blocks" that are treated as if they where single elements and therefore cannot be divided by genetic operators.

As can been seen from the description above the algorithm distinguishes between two tasks: One is the optimization or identification of the structure's inherent set of parameters which is achieved by well known conventional methods. The other even more interesting task is the symbolic generation of an appropriate model structure which is done by means of genetic programming.

In Koza's fundamental book on genetic programming (Koza, 1992) the term *symbolic regression* is introduced standing for the process of discovering both the functional form of a target function and all of its necessary coefficients, or at least an approximation to these. Within a number of different examples Koza did show that using the approach of symbolic regression the generation of mathematical expressions approximating a functional

relation described by a given set of input/output data is possible. On first sight this task is very closely related to the development of a structured process model. However there remain some major problems: Although Koza studied only simple examples, the complexity of the expressions generated by symbolic regression was much bigger than that of the minimal solution. That was due to the fact, that only a few constant numbers that could be used as coefficients in the expressions where given as part of the *terminal set*. Second the approach of symbolic regression does not provide a methology to introduce dynamic behaviour to the generated expressions.

In order to overcome these problems we decided not to let the genetic algorithm generate mathematical expressions but to build up block diagramms as they are commonly used in control theory to describe a system and its inner structure. Therefore the *function set* consists of the basic arithmetic functions and transfer blocks such as

- dynamic elements like time-delay of different order or dead time,
- non-linear elements like switches or limiter,
- elements that let the system interprete subexpressions as feedforward or feedback loops and
- domain specific elements like Monod kinetics or bell-shaped functions.

Most of these elements include certain parameters – e.g. gain and delay time of a first-order time-delay element –, that are initialized by random. During the evolutionary process these parameters are not explicitly modified by the genetic operators but – as described above – adapted by a conventional search algorithm which is applied to the entire model. Fig. 3 shows an example of a

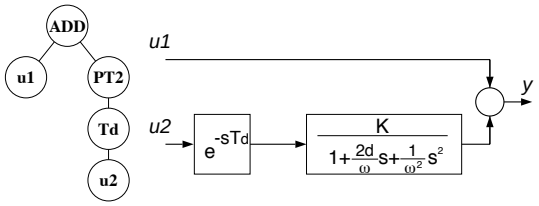


Fig. 3. Structured process model; depicted as a tree (left) and as a block diagramm (right).

simple model depicted as a block diagramm and as the equivalent genetic representation in a tree structure.

For most kinds of control purposes models are needed that combine high accuracy and low complexity. That means the objective for the genetic programming algorithm is not only to find a model that perfectly approximates the given training data, but also to take the complexity of the model's structure into account. Therefore fitness of a model or entity $i$ is calculated by

$$F_i = \frac{1}{1 + E_i \cdot C_i} \tag{1}$$

where $E_i$ is the root-mean-square error and $C_i$ is a heuristic complexity value.

As it has been shown by (Goldberg and Deb, 1991) for standard genetic algorithms the *tournament selection* scheme, where the statistical probability for the selection of an entity $P_i$ can be calculated from its position $p_i$ in an imaginary fitness ranking list by

$$P_i = \sum_{j=1}^{k} \left(\frac{1}{n}\right)^j \left(\frac{n - p_i}{n}\right)^{k-j} \binom{n}{j} \tag{2}$$

,where $n$ is the size of the population and $k$ that of the tournament, provided much better evolutionary performance than the proportionate reproduction proposed in (Holland, 1975) and it was therefore used in the experiments decribed below.

The digital simulation of the dynamic behaviour of each model which is needed for parameter adaption and the determination of its fitness value appears to be the major time consuming factor of this approach. Due to the block oriented representation of the models the dynamic elements can be simulated using their $\mathcal{Z}$-transformation. Compared to the use of numerical integration methods which is necessary when choosing a equation oriented representation this drastically reduces the requirements of computation time.

The concept of self-organizing generation of process models introduced here was implemented in C++ for UNIX workstations. The evolutionary structure search and the task of parameter adaption were realized in separate programs. Due to the inherent parallelism of genetic algorithms this provides a simple but powerful and flexible way to parallel computation.

## 3. FERMENTATION PROCESS

For a fermentation process which is currently developted at the laboratories of the BASF AG, Germany, the determination of the various concentrations of the biomass (bacteria), substrate and product is desired for the further process optimization.

In this fermentation process plant oil is used as substrate. Plant oils are common substrates for microbial fermentations: they are cheap, clean and have a high energy content. The insolubility of plant oil in water leads to the formation of a sec-

ond fluid phase. In this oil-water emulsion a determination of the desired process variables is impossible and time consuming. By standard methods it is therefore impossible to reach an optimal yield by setting up a control strategy for an optimal supply of the bacteria with substrate at every stage of the fermentation.

An estimation of the state variables is therefore desired. The classical approach needs a physical process model correlating measured variables and state space variables. In our case the formulation of physical based models is – as often in fermentations – not possible. Based on publications of (Stephanopoulos and San, 1984) a simple stoichiometric model was built. The model uses the concentration of $CO_2$ and $O_2$, of the exhaust gas of the fermenter measured by a mass spectrometer, to solve the balances for Carbon and Oxygen. This global process model was implemented as a Kalman-Bucy filter. Measured and estimated values were in good agreement.

However, because the estimates are solely based on the results of the mass spectrometer in practice analytical drifts were not detected and are leading to bad state estimates. To reach higher redundance further on line measurable parameters should be considered in the process model. Due to the lack of physical correlations of these parameters this is not possible. As an alternative approach neural networks were considered but discarded because of their specific disadvantages which were discussed in the introduction. Instead the approach of a data driven generation of structured process models described above was applied to this process.

## 4. EXPERIMENTS AND RESULTS

For the experimental examinations only five data records $D_i$ were available. They are arranged as data set $S_1 = \{D_1, D_2, D_4\}$ operated at constant setpoints temperature $\vartheta_1$ and $pH_1$ and data set $S_2 = \{D_3, D_5\}$ operated at temperature $\vartheta_2 < \vartheta_1$ and $pH_2 > pH_1$.

First experiments were prepared in order to achieve continuously available estimations of dominant process states – which can be measured only with a time delay and not during the night – based on the on line measured data. The calculations occured in a part of the local *Sun* workstation cluster at Darmstadt University of Technology (fig. 4). A single run of testing and comparing 700 generations of 700 entities each was computed within 24 hours. These 490,000 models can hardly be derived and coordinated by a human operator. A

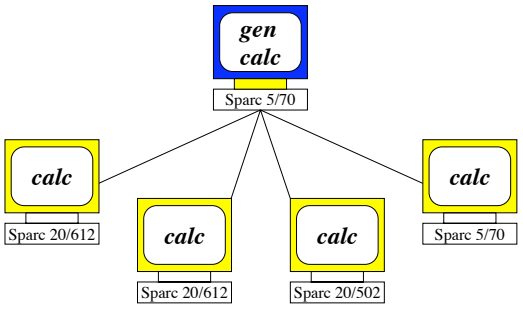


Fig. 4. Parallel processing; evolutionary structure search (*gen*) and parameter adaption (*calc*).

| | Name | Function |
|---|---|---|
| A | *Add* | summer |
| B | *Sub* | subtractor |
| C | *Mult* | multiplicator |
| D | *Div* | divider |
| E | $PT_1$ | first-order time-delay |
| G | *P* | proportional-action |
| I | *I* | integrator |
| J | *Td* | dead time |
| M | *M* | Monod kinetic[1] |
| T | *Exp* | exponential function |
| U | *Gauss* | bell-shaped function |
| V | *Sig* | sigmoid function |
| X | $F_b$ | feedback loop (negative) |
| Y | $F_f$ | feedforward loop |

Table 1 List of all used elements.

couple of linear and non-linear elements were chosen for the function set – see table 1 – and a careful selection was applied to the modelling process and changed step by step.

In this paper we will concentrate on three experimental results out of a large number of carefully judged models generated by this approach, estimating the off line measurable biomass concentration.

| | |
|---|---|
| Data sets: | $M_1 = \{D_1, D_2, D_4\}$ |
| Population size: | 700 |
| Generations: | 700 |
| Function set: | A,B,E,I,M,U,T,Y,X |
| Input variables: | *oil*, *pH*, $\vartheta$, $OTR^2$, $CTR^3$, $RQ^4$ |

Table 2 Configuration of run A.

[1] Here the Monod kinetic is realized as a substrate limited growth described by

$$\dot{X} = \frac{\mu_{max} \cdot S}{K + S} X \qquad (3)$$

with S as the input and X as the output value.

The first of all experiments was configured according to table 2. The computer generated a simple linear model using one input variable, two first-order time-delay elements with a parallel (unfeasible) feedthrough and a final integrator – see fig. 5. As fig. 6 illustrates, the trained data sets $D_1$,

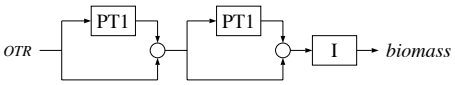


Fig. 5. Best entity of run A.

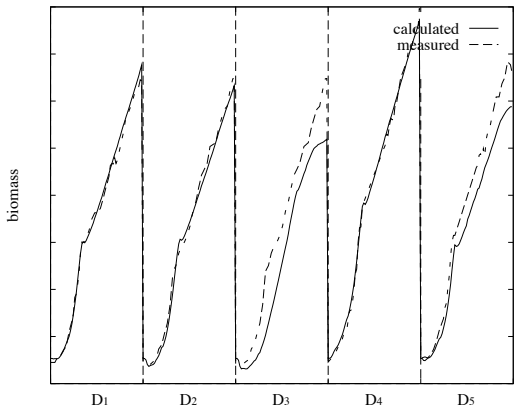


Fig. 6. Comparison of measured and calculated shapes using the best entity of run A.

$D_2$ and $D_4$ are well fitted but medium until large differences are obtained while estimating the output behaviour of the untrained data sets $D_3$ and $D_5$.

| | |
|---|---|
| Data sets: | $M_2 = \{D_3, D_5\}$ |
| Population size: | 700 |
| Generations: | 700 |
| Function set: | A,B,E,M,U,T,Y,X |
| Input variables: | $oil$, $pH$, $\vartheta$, $OTR$, $CTR$, $RQ$ |

Table 3 Configuration of run B.

One of the next experiments – configuration according to table 3 – produced a sensible nonlinear model which is shown in fig. 7. But the reproduction test (fig. 8) shows that the generalization towards the untrained data sets is absolutely insufficient. An overfit as a specialization on the training samples occured like it is well known in the neural network and the adaptable pattern classification domain.

[2] oxygen transfer rate

[3] carbon dioxide transfer rate

[4] respiratory quotient

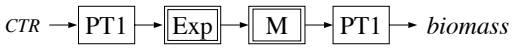


Fig. 7. Best entity of run B.

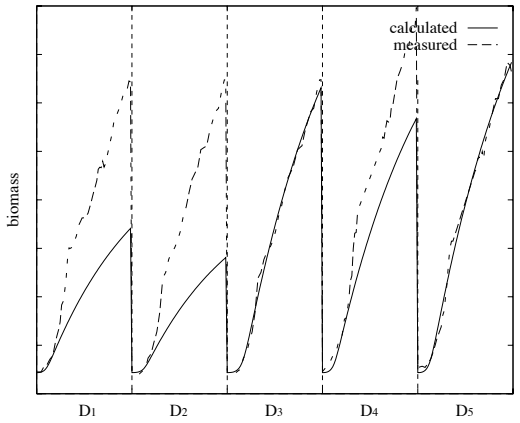


Fig. 8. Comparison of measured and calculated shapes using the best entity of run B.

One of the most interesting results is the model – shown in fig. 9 – gained with the configuration in table 4. A cascade of three fedback non-linear Monod kinetics (eqn. 3) with the input variable feed rate estimates the ouput variable biomass concentration. The signals between these modules can be interpreted as intermediate products built in the fermenter. Fig. 10 shows, that the trained as well as the untrained data sets are well reproduced and estimated respectively. This basic model is actually deeper examined integrating temperature and $pH$ dependencies gained from further experiments.

## 5. CONCLUSION

In this paper we have presented an application of the genetic programming paradigm for the modelling of a biotechnological fed-batch process. The approach described here combines novel results of computer science – genetic programming – with well known and proven techniques of control and system theory – block diagramms and $\mathcal{Z}$-transformation. The synthesis of these approaches is a powerful tool for data-driven modelling that offers a large number of possibilities to integrate existing knowledge e.g. on submodels or expected elements. The models received by the use of this tool provide a transparent insight into the structure of the process and a basis for long-term prediction of the process behaviour and therefore

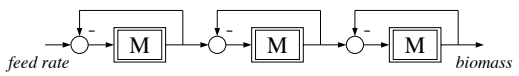


Fig. 9. Best entity of run C.

| Data sets: | $M_2 = \{D_3, D_5\}$ |
|---|---|
| Population size: | 700 |
| Generations: | 700 |
| Function set: | A,B,G,M,U,T,Y,X |
| Input variables: | $oil$, $pH$, $\vartheta$, $RQ$ |

Table 4 Configuration of run C.

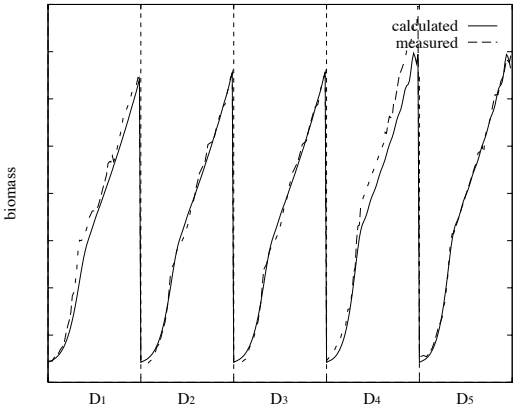


Fig. 10. Comparison of measured and calculated shapes using the best entity of run C.

for the determination of optimal setpoint profiles. That means that this approach may overcome the specific difficulties that are bound to the use of adaptive or learning – in the sense of neural networks – methods.

However it appears that the generation of models by means of genetic progamming cannot make the engineer unnessesary who is entrusted with the task of modelling. Instead he is given a powerful tool that allows him to concentrate on creative considerations while it creates and assesses a huge number of structures and models. The final decision about which model appears to be a plausible approximation of the physical reality and to be well suited for control purposes stays in his hand. But he has the possibility to guide the evolutionary process in an interactive way by introducing concrete ideas and modifying experimental constraints.

First results of the application of this approach to the task of finding a model of a biotechnological process presented here have shown that the concept is working. Compact models capable of a good approximation of trained and not trained data were achieved. Furthermore these first results stimulated a new series of experiments on the real process in order to validate new ideas that were inspired by the self-organized generated models. Therfore the actual work is concentrated on the model extension and the generation of dynamic process control strategies based on these models as well as on the examination of the applicability of newest results on genetic programming.

## 6. REFERENCES


Albus, J.S. (1972). Theoretical and Experimental Aspects of a Cerebellar Model. PhD thesis. University of Maryland. Maryland.

Bailey, James E. and David F. Ollis (1986). *Biochemical engineering fundamentals*. 2 ed.. McGraw-Hill. New York.

Bastin, Georges and Denis Dochain (1990). *On-line Estimation and Adaptive Control of Bioreactors*. Elsevier Science Publishers B.V., ISBN 0-444-88430-0. New York.

Chen, Libei, Georges Bastin and Vincent van Breusegem (1995). A case study of adaptive nonlinear regulation of fed-batch biological reactors. *Automatica* **31**(1), 55–65.

Gehlen, Stefan (1993). Untersuchungen zur wissensbasierten und lernenden Prozeßführung in der Biotechnologie. PhD thesis. TH Darmstadt, FG Regelsystemtheorie & Robotik. Fortschritt-Berichte VDI, Reihe 20, Rechnerunterstützte Verfahren, Nr. 87, VDI-Verlag.

Gehlen, Stefan and Kurt Dirk Bettenhausen (1990). Modelling of biotechnological processes with interpolating associative memories. In: *International Symposium on Mathematical and Intelligent Models in System Simulation*. Brüssel.

Goldberg, David E. and Kalyanmoy Deb (1991). A comparative analysis of selection schemes. In: *Foundations of Genetic Algorithms* (Gregory J. R. Rawlins, Ed.). Morgan Kaufmann Publishing.

Holland, John H. (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press.

Hooke, Robert and T. A. Jeeves (1961). Direct search: Solution of numerical and statistical problems. *Journal of the Association of Computing Machinery* pp. 212–224.

Koza, John R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press. Cambridge, Massachusetts. ISBN 0-262-11170-5.

Moser, Anton (1988). *Bioprocess technology*. Springer-Verlag. Wien.

Präve, Paul, Uwe Faust, Wolfgang Sittig and Dieter A. Sukatsch (1987). *Handbuch der Biotechnologie*. R. Oldenbourg Verlag. München.

Stephanopoulos, G. and K.-Y. San (1984). Studies on the on-line bioreactor identification. *Biotechnology and Bioengineering* **26**, 1176—1218.

Tolle, Henning and Enis Ersü (1992). *Neurocontrol*. number 172 In: *Lecture Notes in Control and Information Sciences*. Springer-Verlag.