# Data Driven Structured Modelling of a Biotechnological Fed-batch Fermentation by Means of Genetic Programming

**P Marenbach**, Dipl.-Ing.
Institute of Control Engineering, Darmstadt University of Technology, Darmstadt, Germany

**K D Bettenhausen**, Dr.-Ing.
Hoechst AG, Corporate Research and Technology, Frankfurt am Main, Germany

**S Freyer**, Dr. rer. nat. and **U Nieken**, Dr.-Ing. and **H Rettenmaier** Dr. rer. nat.
BASF AG, Ludwigshafen, Germany

*The article at hand describes an approach for data driven generation of structured models of complex and unknown processes by means of Genetic Programming. The basic approach which is used to generate and to modify symbolic model descriptions represented as block diagrams is introduced and an application for modelling of an industrial biotechnological fed-batch fermentation process is presented.*

*Key words: genetic programming, system identification, biotechnology, fermentation processes.*

## NOTATION

| | |
|---|---|
| $C_i$ | complexity value of the $i$-th solution |
| $E_i$ | error value of the $i$-th solution |
| $F_i$ | fitness value of the $i$-th solution |
| $P_i$ | selection probability of the $i$-th solution |
| $S$ | substrate concentration |
| $X$ | biomass |
| $\vartheta$ | temperature |
| OTR | oxygen transfer rate |
| CTR | carbon dioxide transfer rate |
| RQ | respiratory quotient |

## 1 INTRODUCTION

The natural metabolism of living organisms can be used for the production of food, medicines or basic materials for the chemical industries. The major task for optimisation of those biotechnological processes is to overcome natural limitations either by genetic manipulation of the organisms or variation of environmental conditions during the fermentation. This variation is part of the process engineers task 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 (**2**) – is very difficult due to

1. the lack of quantitative process knowledge,

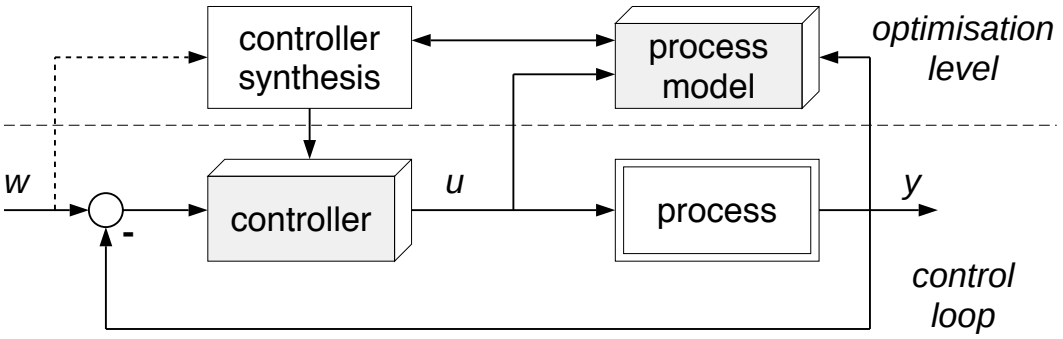


Figure 1: Model based predictive control – see e.g. textbook of Tolle and Ersü (**1**).

2. the fact that most of the process variables relevant to the metabolism cannot be measured on line,
3. the high amount of costs and time that is bound to the generation of experimental data and
4. 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 series of experiments in laboratory scale – see e.g. textbooks of Bailey and Ollis (**3**). Dynamic variations of the setpoints that provide the possibility of optimising process performance are not used, due to the missing knowledge about how and when to occur.
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. by Bastin and Dochain (**4**) – based on the well known mass balances in the reactor. Instead of developing a model of the 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 explicit representation of the mentioned dependencies does not exist, there is no chance ance of deriving strategies for an optimal choice of setpoint profiles from the underlying model. However this can be achieved by models based on a learning approach 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* (NN). These neurally motivated memories approximate non-linear manifolds by interpolating between information "learned" from a given training data set. It was shown by Gehlen (**5**) that a process optimisation based on a learning approach is possible and that it can lead to a significant increment of product yield. Gehlen used an interpolating associative memory of the CMAC (*Cerebellar Model Articulation Controller*) type as introduced by Albus (**6**), which is particularly suited for tasks of modelling. However there are still some disadvantages of the learning conception:

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

## 2 DATA DRIVEN STRUCTURED MODELLING

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 approaches in industrial applications as well as the governmental approval.
The new approach of the data driven structured modelling by means of *Genetic Programming* (GP) 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 data driven structured modelling, which is indeed very similar to the way models are developed by a process engineer – see e.g. Moser (**2**). Starting with a collection of elementary transfer

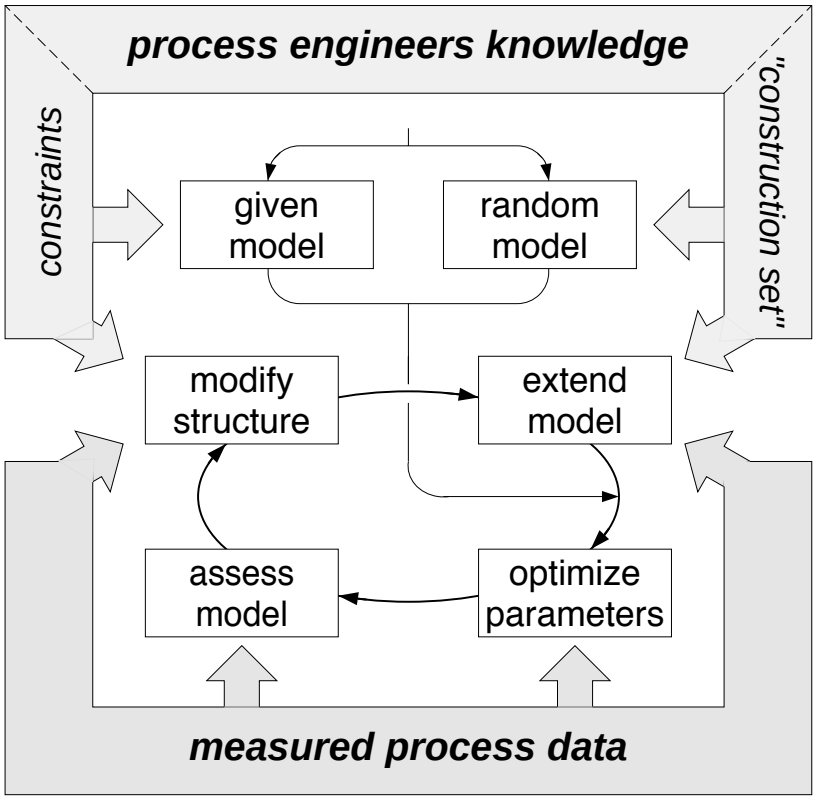


Figure 2: Scheme of automatic generation of structured process models.

elements like time-delay or Monod kinetics placed in a so called "model construction set", a number of models is created. The degree to which this is done at random depends on how much a-priori knowledge is available. In an evolutionary process the following three steps are performed iteratively. First each model of a generation is adapted to measured process data by optimising its internal parameters, using well known parameter search methods – e.g. direct search algorithm by Hooke and Jeeves (**7**). 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 – introduced as *Genetic Algorithm* (GA) by Holland (**8**) – finally leads to models that combine high accuracy and low complexity, which are needed for most kinds of control and further process development purposes.
As can been seen from the description above, the algorithm distinguishes between two tasks: One is the optimisation 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 GP.

## 2.1 Symbolic model representation

In the fundamental textbook on GP by Koza (**9**) 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: 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 is that the approach of symbolic regression does not provide a methodology to introduce dynamic behaviour to the generated expressions.
In order to overcome these problems it was decided not to let GP generate mathematical expressions but to build up block diagrams as they are commonly used in control theory to describe a system and its inner structure. A further advantage of this representation is that block diagrams are much closer to the way human experts think of a process due to the focus on interactions between input and output values instead of on mathematics.
Therefore the *function set* consists of the basic arithmetic functions and transfer blocks such as

1. dynamic elements like time-delay of different order or dead time,
2. non-linear elements like switches or limiters,
3. feedforward or feedback loop creating blocks and
4. domain specific elements like Monod kinetics or bell-shaped characteristics.

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 simple model depicted as a

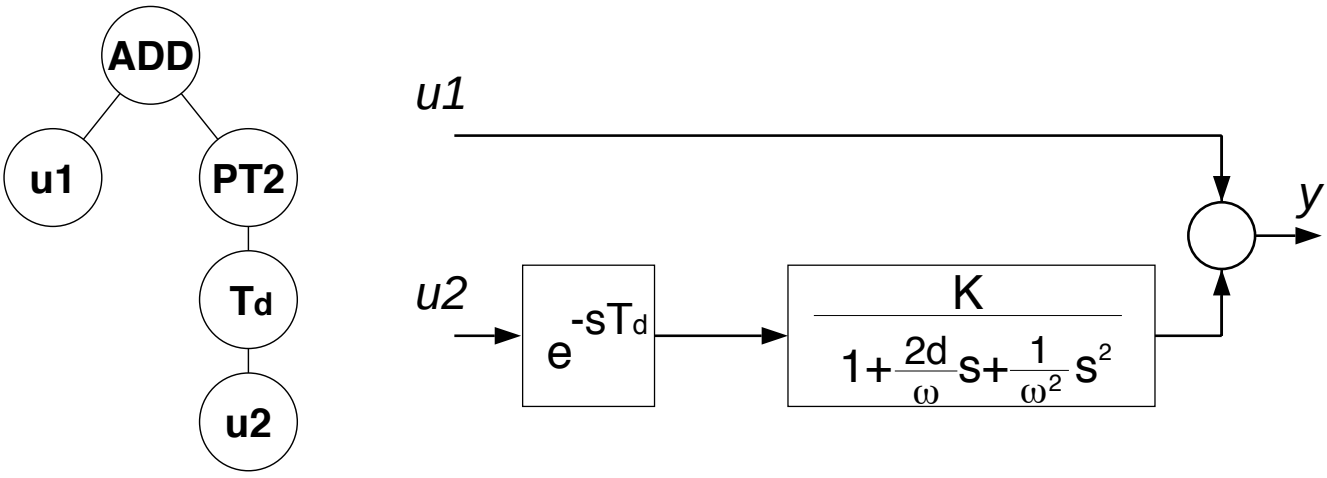


Figure 3: Structured process model; depicted as a tree (left) and as a block diagram (right).

block diagram and as the equivalent genetic representation in a tree structure. The model given in this example corresponds to the following system of first-order differential equations:

$$\dot{x}_1(t) = x_2(t)$$

$$\begin{aligned} \dot{x}_2(t) &= K\omega^2 \cdot u_2(t - T_d) - \omega^2 \cdot x_1(t) - 2d\omega \cdot x_2(t) \\ y(t) &= u_1(t) + x_2 \end{aligned} \tag{1}$$

In the second example in Fig. 4 the realization of feedforward or feedback loops is described. Within the tree structure the node labeled *Fb* stands for the beginning of a feedback loop and its left branch defines the blocks within the loop. The terminal of this left branch – no matter of what kind (e.g. an input variable) it was and therefore labeled ** here – marks the end of the loop. Due to this mechanism the implemented

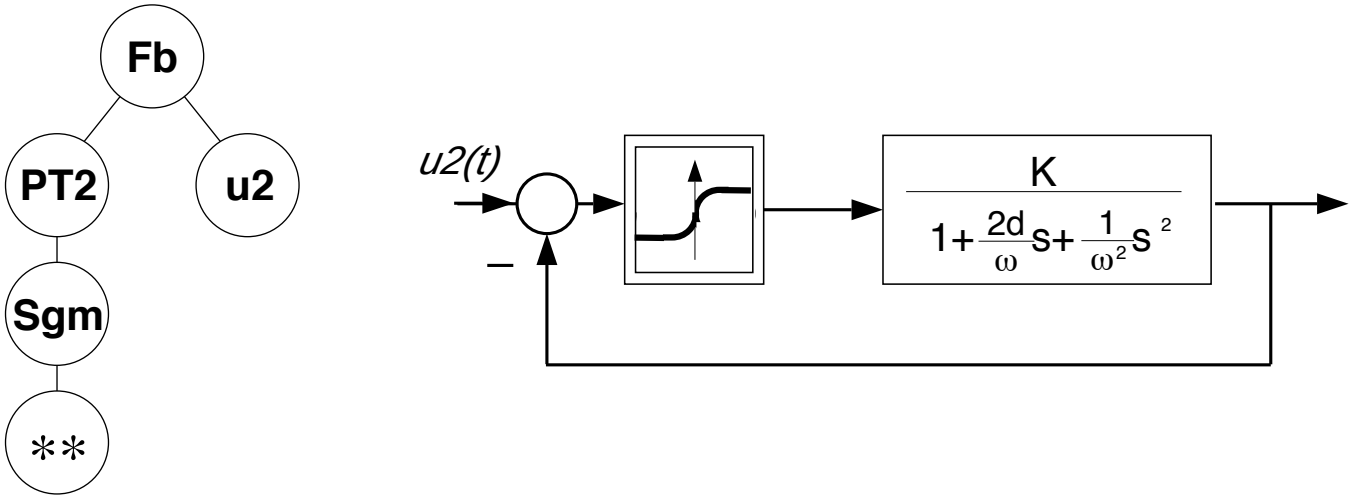


Figure 4: Realization of a feedback loop; depicted as a tree (left) and as a block diagram (right).

genetic operators do not have to distinguish whether they are applied inside or outside of such a loop.

## 2.2 Use of a-priori knowledge

One of the most important advantages of the concept of data driven model structure generation presented here compared to the usage of NNs is, that it provides several ways to take a-priori knowledge on structural properties into account. First of all the elements within the model construction set can be selected from a variety of different blocks and their selection frequency can be influenced. Furthermore elements can be combined into *super blocks* that are treated as if they where single elements and therefore cannot be divided by genetic operators. If a model is partially known or if the system shall be forced to take certain parts as fixed, a so-called *fixed sub-model* can be defined. As a result each individual starts with the same subtree which must not be changed by genetic operators.

## 2.3 Implementation details

For most kinds of control purposes models are needed that combine high accuracy and low complexity. That means the objective for the GP 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 the accuracy model or entity $i$ is assesses by an error value $E_i$ which is defined as the root"=mean"=square error between given process behaviour and simulated model response and a model complexity $C_i$ is evaluated as the sum of a heuristically defined complexity value of each block within the model. Based on these two objectives, *raw fitness* $F_i$ respectively *standardized fitness* $f_i$ is calculated by

$$F_i = E_i \cdot C_i \tag{2}$$

$$f_i = \frac{1}{1 + F_i} = \frac{1}{1 + E_i \cdot C_i} \tag{3}$$

where the standardization formula 3 was suggested by Koza (**9**).
As it has been shown by Goldberg and Deb (**10**) for standard GAs 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

$$P_i = \sum_{j=1}^{k} \left(\frac{1}{n}\right)^j \left(\frac{n - p_i}{n}\right)^{k-j} \binom{n}{j} \quad , \tag{4}$$

where $n$ is the size of the population and $k$ the size of the tournament, provided much better evolutionary performance than the proportionate reproduction proposed by Holland (**8**) and it was therefore used in the experiments described below.
The whole system 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 GAs this provides a simple but powerful and flexible way to parallel computation by distributing the time consuming fitness evaluation within a heterogeneous workstation cluster and collecting the results in a single program which performs the symbolic model modification.

# 3 FERMENTATION PROCESS

For a fermentation process which is currently developed at the laboratories of the BASF AG, Germany, the determination of the various concentrations of the biomass (bacteria), substrate and product is desired for further process optimisation (cf. Fig. 5).

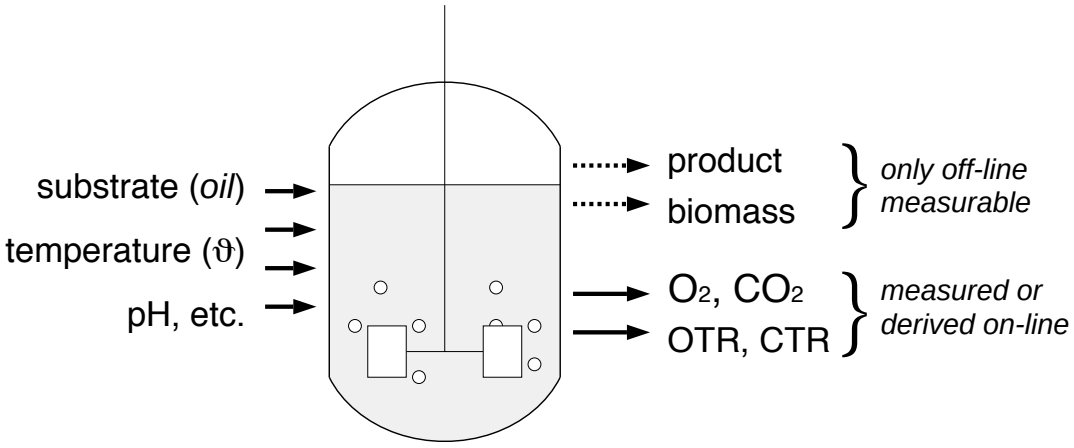


Figure 5: Fermentation process; on-line measured values and only off-line measurable state variables.

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 second 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 achieve 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 physically based models is – as often in fermentations – not possible. Based on publications of Stephanopoulos and San (**11**) a simple stoichiometric model was built. The model uses the concentration of $CO_2$ and $O_2$, of the exhausted 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 lead to bad state estimations. To reach higher redundancy 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 of this, 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 workstation cluster at Darmstadt University of Technology (Fig. 6). A single run of testing and comparing 700 generations of 700 entities each was

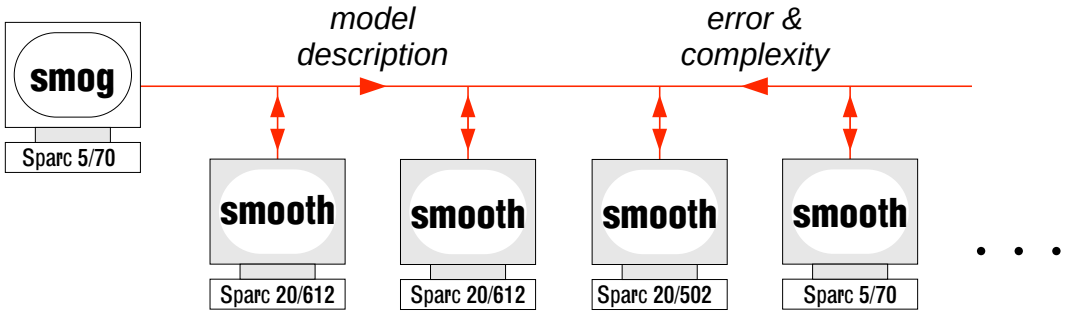


Figure 6: Parallel processing; evolutionary structure search (smog) and parameter adaption (smooth).

Table 1: *Model Construction Set*: List of all used elements.

| Name | Function | Name | Function |
|---|---|---|---|
| $Add$ | summer | $PDT_2$ | 2nd-order time-delay with differential component |
| $Sub$ | subtractor | | |
| $Mult$ | multiplicator | $M$ | Monod kinetics |
| $Div$ | divider | $Off$ | offset |
| $PT_1$ | first-order time-delay | $Lim$ | limiter |
| $PT_2$ | second-order time-delay | $Exp$ | exponential characteristic |
| $P$ | proportional-action | $Gauss$ | bell-shaped characteristic |
| $DT_1$ | 1st-order time-delay with differential component | $Sig$ | sigmoid characteristic |
| | | $F_b$ | feedback loop (negative) |
| $I$ | integrator | $F_f$ | feedforward loop |
| $Td$ | dead time | | |

computed within 24 hours. These 490,000 models can hardly be derived and coordinated by a human operator.

A couple of linear and non-linear elements were selected to be part of the construction set – 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.

Table 2: Configuration of run A.

| | |
|---|---|
| Data sets: | $S_1 = \{D_1, D_2, D_4\}$ |
| Population size: | 700 |
| Generations: | 700 |
| Function set: | A,B,E,I,M,U,T,Y,X |
| Input variables: | $oil$, $pH$, $\vartheta$, $OTR$, $CTR$, $RQ$ |

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. 7. As Fig. 8 illustrates, the trained data sets $D_1$, $D_2$ and $D_4$ are well

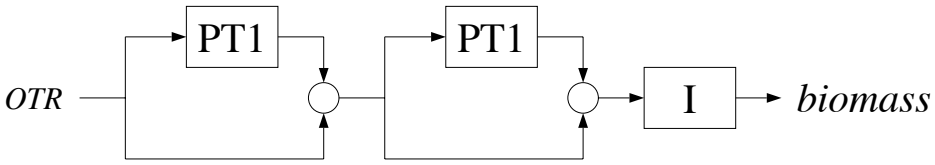


Figure 7: Best entity of run A.

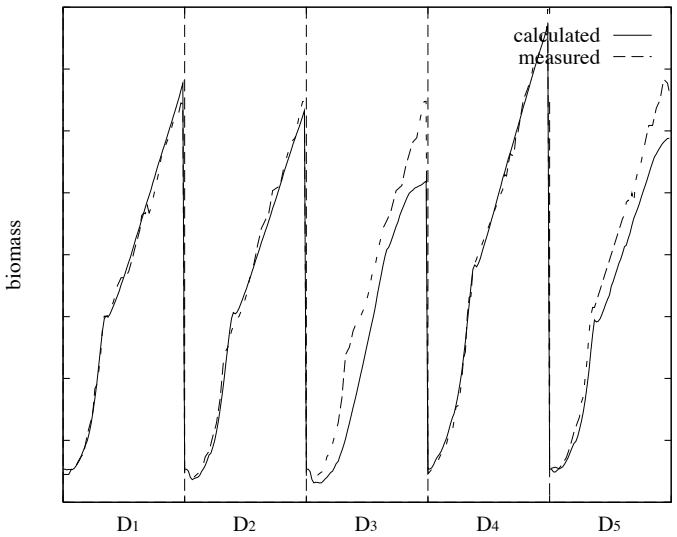


Figure 8: Comparison of measured and calculated shapes using the best entity of run A.

fitted but medium to large differences are obtained estimating the output behaviour of the untrained data sets $D_3$ and $D_5$.
One of the next experiments – configuration according to table 3 – produced a sensible non"-linear model which is shown in Fig. 9. However the reproduction test (Fig. 10) shows, that the generalization towards the untrained data sets is absolutely insufficient. Overfitting as specialization on the training samples occured like it is well known in the NN and the adaptable pattern classification domain.

Table 3: Configuration of run B.

| | |
|---|---|
| Data sets: | $S_2 = \{D_3, D_5\}$ |
| Population size: | 700 |
| Generations: | 700 |
| Function set: | A,B,E,M,U,T,Y,X |
| Input variables: | $oil$, $pH$, $\vartheta$, $OTR$, $CTR$, $RQ$ |

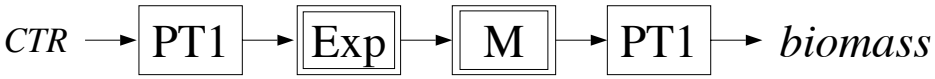


Figure 9: Best entity of run B.

One of the most interesting results is the model – shown in Fig. 11 – gained with the configuration in table 4. A cascade of three fed back non-linear Monod kinetics standing for a substrate limited growth described by the formula

$$\dot{X} = \mu_{max} \cdot \frac{S}{K+S} X \tag{5}$$

with the input variable feed rate estimate the output variable biomass concentration. The signals between these modules can be interpreted as intermediate products. Fig. 12 shows, that the trained as well as the untrained data sets are well reproduced and estimated respectively. Additional experiments concerning the generalization capabilities of this model have shown, that the input feed rate can be varied within an interval from $-20\%$ until $+10\%$ while the output produces still a sensible prediction of the biomass.
A further improvement – for details see Bettenhausen (**12**) – can be achieved by interactions of the user. In order to demonstrate this a run was started in which promising results from earlier runs each of them with certain advantages and disadvantages were combined. A model structure that gives a very good approximation of the lag phase but is not able to represent the behaviour during the whole fermentation was linked to the models shown in Fig. 9 and 11 which are advantageous within other physiological process phases. The weighted integration of these three models, shown in Fig. 13, leads to a detailed model with a very good representation of the whole fermentation behaviour as can be seen from Fig. 14.

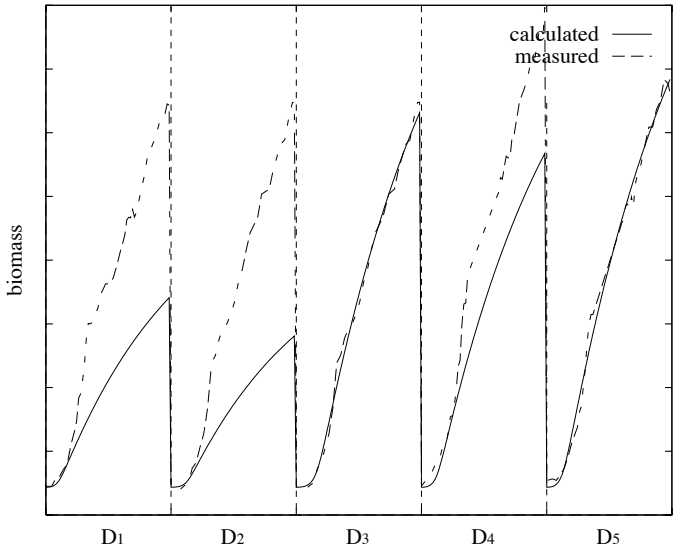


Figure 10: Comparison of measured and calculated shapes using the best entity of run B.

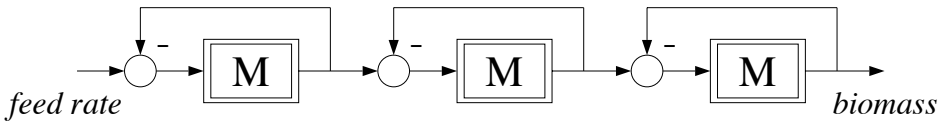


Figure 11: Best entity of run C.

Table 4: Configuration of run C.

| | |
|---|---|
| Data sets: | $S_2 = \{D_3, D_5\}$ |
| Population size: | 700 |
| Generations: | 700 |
| Function set: | A,B,G,M,U,T,Y,X |
| Input variables: | $oil$, $pH$, $\vartheta$, $RQ$ |

# 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 diagrams. 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 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 this approach of data driven generation of structured models cannot make the engineer unnecessary 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 remains in his hand. But he has the possi-

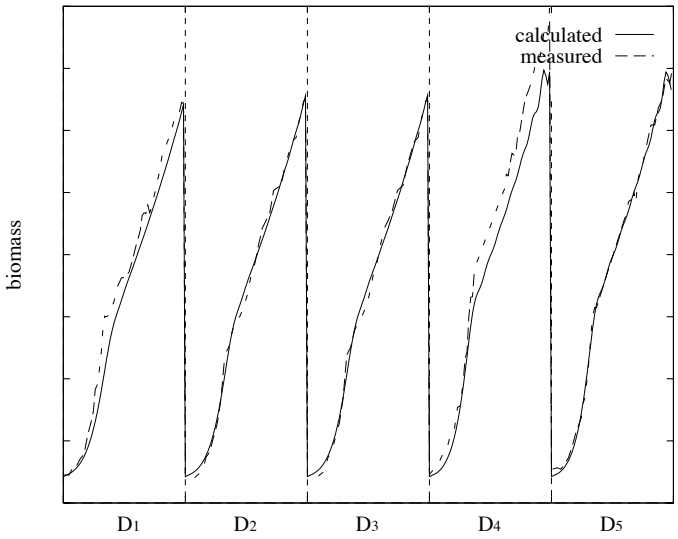


Figure 12: Comparison of measured and calculated shapes using the best entity of run C.

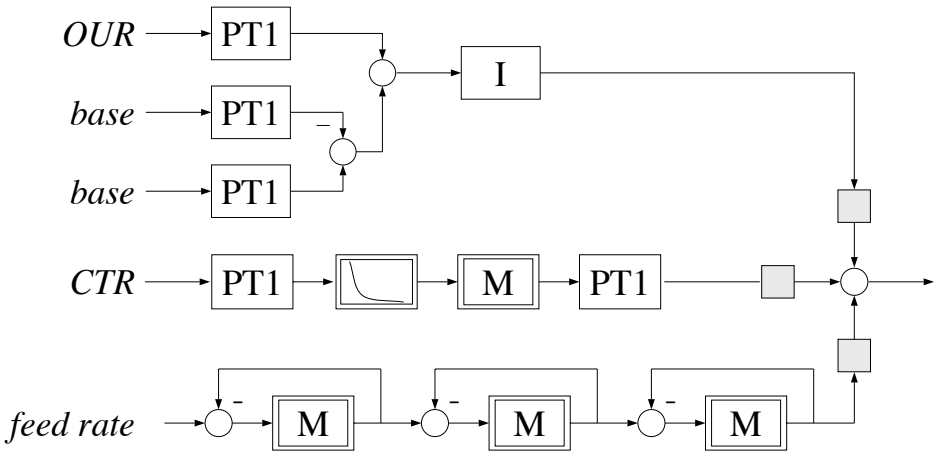


Figure 13: Integration of phase-specific partial models.

bility 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 data driven generated model structures. Therefore 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 and of different techniques of model structure validation.

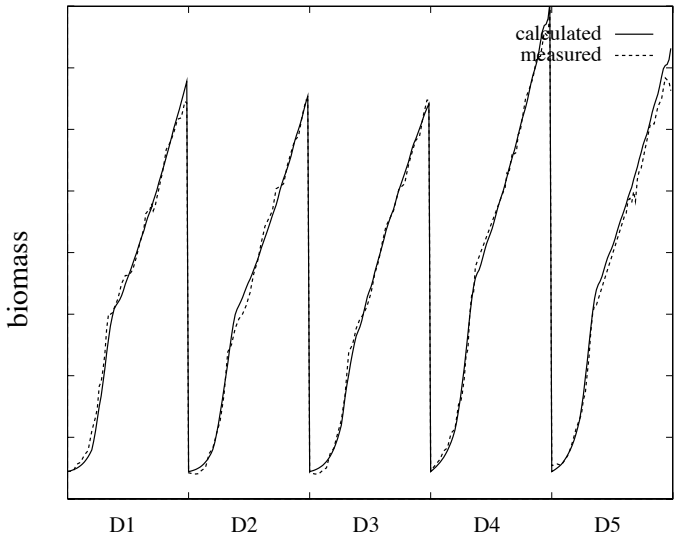


Figure 14: Comparison of measured and calculated shapes using the integration of phase-specific partial models.

## References


**1** Tolle, H. and Ersü, E. *Neurocontrol*, Vol. 172 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, Berlin, 1992.

**2** Moser, A. *Bioprocess technology*, Springer-Verlag, Wien, 1988.

**3** Bailey, J. E. and Ollis, D. F. *Biochemical engineering fundamentals*, 2 edn, McGraw-Hill, New York, 1986.

**4** Bastin, G. and Dochain, D. *On-line Estimation and Adaptive Control of Bioreactors*, Elsevier Science, New York, 1990.

**5** Gehlen, S. *Untersuchungen zur wissensbasierten und lernenden Prozeßführung in der Biotechnologie*, PhD thesis, TH Darmstadt, 1993. (In German)

**6** Albus, J. *Theoretical and Experimental Aspects of a Cerebellar Model*, PhD thesis, University of Maryland, Maryland, 1972.

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

**8** Holland, J. H. *Adaptation in natural and artificial systems*, The University of Michigan Press, 1975.

**9** Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Massachusetts, 1992.

**10** Goldberg, D. E. and Deb, K. A comparative analysis of selection schemes, *in* G. J. R. Rawlins (ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishing, 1991.

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

**12** Bettenhausen, K. D. *Automatische Struktursuche für Regler und Strecke: Beiträge zur datengetriebenen Analyse und optimierenden Führung komplexer Prozesse mit Hilfe evolutionärer Methoden und lernfähiger Fuzzy-Systeme*, PhD thesis, TH Darmstadt, 1996. (In German)