Biomass Inferential Sensor Based on Ensemble of Models Generated by Genetic Programming

Arthur Kordon[1*], Elsa Jordaan[2], Lawrence Chew[3], Guido Smits[2], Torben Bruck[3], Keith Haney[3], and Annika Jenings[3]

[1] The Dow Chemical Company, Corporate R&D, 2301 N Brazosport Blvd, Freeport, TX, 77541, USA
[2] Dow Benelux BV, Corporate R&D, 5 Herbert H Dowweg, Terneuzen, The Netherlands
[3] The Dow Chemical Company, Biotechnology, 5501 Oberlin Dr., San Diego, CA 92121, USA
* corresponding author akordon@dow.com

Abstract. A successful industrial application of a novel type biomass estimator based on Genetic Programming (GP) is described in the paper. The biomass is inferred from other available measurements via an ensemble of nonlinear functions, generated by GP. The models are selected on the Pareto front of performance-complexity plane. The advantages of the proposed inferential sensor are: direct implementation into almost any process control system, rudimentary self-assessment capabilities, better robustness toward batch variations, and more effective maintenance. The biomass inferential sensor has been applied in high cell density microbial fermentations at The Dow Chemical Company.

1 Introduction

Soft (or inferential) sensors infer important process variables (called outputs) from available hardware sensors (called inputs). Usually the outputs are measured infrequently by lab analysis, material property tests, expensive gas chromatograph analysis, etc. Furthermore, the output measurement is very often performed off-line and then introduced into the on-line process monitoring and control system. Soft sensors, on the other hand, can predict these outputs online and frequently using either data supplied from standard, and frequently cheap, hardware sensors or from other soft sensors.

Different inference mechanisms can be used for soft sensor development. If there is a clear understanding of the physics and chemistry of the process, the inferred value can be derived from a fundamental model. Another option is to estimate the parameters of the fundamental model via Kalman Filter or Extended Kalman Filter. There are cases when the input/output relationship is linear and can be represented either by linear regression or by a multivariate model. The most general representation of the inferential sensor, however, is as a nonlinear empirical model. The first breakthrough technology for soft sensor development was neural networks because of their ability to capture nonlinear relationships and their adequate

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 1078–1089, 2004.

framework for industrial use (i.e. no additional cost for fundamental model building and data collection based on design of experiments).

The common methodology of building neural net soft sensors and the practical issues of their implementation have been discussed in detail in [2]. Thousands of soft sensor applications in the chemical, petro-chemical, pharmaceutical, power generation, and other industries have been claimed by the key software vendors. However, despite their successes, most commercially available neural net packages are still based on a classical back-propagation algorithm. Some other recent approaches, like Bayesian neural networks [18] are still in the research domain and have not been accepted for industrial applications. As a result, those commercial neural networks generally exhibit poor generalization capability outside the range of training data [1]. This can result in poor performance of the model and unreliable prediction in new operating conditions. Another drawback is that such packages usually yield neural net structures with unnecessarily high complexity. Selection of the neural net structure is still an *ad hoc* process and very often leads to inefficient and complex solutions. As a result of this inefficient structure and reduced robustness, there is a necessity of frequent re-training of the empirical model. The final effect of all of these problems is an increased maintenance cost and gradually decreased performance and credibility [3].

The need for robustness toward process variability, the ability to handle industrial data (e.g., missing data, measurement noise, operator intervention on data, etc.) and ease of model maintenance are key issues for mass-scale application of reliable inferential sensors. Several machine-learning approaches have the potential to contribute to the solution of this important problem. Stacked analytical neural networks (internally developed in The Dow Chemical Company) allow very fast model development of parsimonious black-box models with confidence limits [4]. Genetic Programming (GP) can generate explicit functional solutions that are very convenient for direct on-line implementation in the existing process information and control systems [5]. Recently, Support Vector Machines (SVM) give tremendous opportunities for building empirical models with very good generalization capability due to the use of the structural risk minimization principle [6]. At the same time, each approach has its own weaknesses, which reduces the implementation space and which make it difficult to design the robust soft sensor based on separate computational intelligence techniques. An alternative, more integrated approach for a "second generation" soft sensor development is described in [4]. It combines a nonlinear sensitivity and time-delay analysis based on Stacked Analytical Neural Nets with outlier detection and condensed data selection driven by the Support Vector Machines. The derived soft sensor is generated by GP as an analytical function. The integrated methodology amplifies the advantages of the individual techniques, significantly reduces the development time, and delivers robust soft sensors with low maintenance cost. The methodology has been successfully applied in The Dow Chemical Company for critical parameter prediction in a chemical reactor [7], for interface level estimation [8], and for emission estimation [9]. However, the increased requirements for robustness towards batch-to-batch variation in inferential sensors applied on batch processes need to be addressed. A potential solution of using an ensemble of symbolic regression predictors, instead of one model, is proposed in this paper. The selected models of the ensemble occupy the Pareto front on the performance-model complexity plane. The experience from implementing the

proposed methodology for biomass estimation during the growth phase of fed-batch fermentation is described in this paper.

2 The Need for Biomass Estimation in Batch Fermentation Processes

Biomass monitoring is fundamental to tracking cell growth and performance in bacterial fermentation processes. In standard fed-batch fermentations of recombinant microorganisms, biomass determination over time allows for calculation of growth rates during the growth phase. Slow growth rates can indicate non-optimal fermentation conditions which can then be a basis for further optimization of growth medium, conditions or substrate feeding. Biomass monitoring is also needed to determine the most optimum time to induce recombinant product formation. During the production phase, biomass decline can forecast onset of cell lysis which, if allowed to continue, can result in decreased product yields. Biomass monitoring therefore can aid in determination of the appropriate time for process termination. In fed-batch fermentations, biomass data can also be used to determine feed rates when yield coefficients are known. Certain high cell density fermentations require growth rates to be controlled in order to prevent accumulation of by-products such as acetic acid or ethanol, which if left unchecked can be detrimental. Biomass monitoring also acts as an indicator of batch-to-batch variability.

There are several inferential sensors implementations for biomass estimation in different continuous and fed-batch bioprocesses [10], [11], [12]. Usually the biomass concentration is determined off-line by lab analysis every 2-4 hours. However, these infrequent measurements can lead to poor control and on-line estimates would be more preferred. In [10], a soft sensor for biomass estimation based on two process variables – fermenter dilution rate and carbon dioxide evolution rate (CER) successfully estimated biomass in continuous *mycelial* fermentation. The neural net model included six inputs that incorporated process dynamics for three consecutive sampling periods, two hidden layers with 4 neurons, and one output, the biomass estimate. Another successful implementation of a biomass soft sensor for a penicillin fed-batch process is described in [12]. The topology of the neural net in this case is (2-3-1), where the two inputs are the oxygen uptake rate (OUR) and the batch time, and the output is penicillin biomass estimates. Good surveys of the current state of the art of the soft sensors and data driven approaches to bioprocess modeling are given in [13] and [14]. However, all applied neural net-based soft sensors expressed the common problems of poor extrapolation, high batch-to-batch sensitivity, and frequent re-training. As a result, long term maintenance becomes the Achilles heel of neural-net-based biomass estimators that significantly reduces their credibility.

3 Robust Inferential Sensors Based on Ensemble of Predictors

Robustness toward process changes is the key requirement for industrial application of inferential sensors. It is crucial especially for batch processes where the batch-to-batch changes are guaranteed even when applying the most consistent operating

discipline [14]. One of the possible ways to improve robustness toward process changes is by using explicit nonlinear functions derived by symbolic regression. The advantages of this first approach to increase robustness in comparison to black-box models, like neural networks, are as follows: potential for physical interpretation, the ability to examine the behavior of the model outside the training range in an easy and direct way, the ability to impose external constraints in the modeling process and to relieve the extrapolation level of the final model toward process changes, and last, but not least, process engineers are more open to take the risk to implement such type of models. The applicability of symbolic-regression-based inferential sensors has already been demonstrated in several industrial applications of continuous processes [7], [8], and [9]. However, an open issue that influences robustness toward process changes is the control of empirical model complexity. As is well-known, Statistical Learning Theory in general and Support Vector Machines in particular, give explicit control over model complexity by the selection of the number of support vectors [6]. There is a defined optimal complexity for the available data, called the Vapnik-Chervonenkis (VC) dimension [6]. Unfortunately, direct application of these theoretical results to symbolic regression-based models faces difficulties. However, the idea of balancing modeling performance and complexity will be explored by selecting models on the Pareto front only and this is the second approach to increase robustness that will be tested.

The third approach that could improve robustness toward process changes is to use an ensemble of predictors. By combining diverse symbolic regression models it is possible to use the statistical characteristics of the ensemble for more reliable prediction and for model self-assessment [15].

3.1 Integrated Methodology for Robust Inferential Sensors Development

The objective of the integrated methodology is to deliver successful industrial inferential sensors with reduced development time, better generalization capability, and minimal implementation and maintenance cost. The main blocks of the methodology (data pre-processing, nonlinear variable selection based on analytic neural networks, data condensation based on SVM, automatic model generation based on GP, model selection, on-line implementation and maintenance) are described in detail in [4]. The key idea is to optimize the synergy between three methods for empirical model building: neural networks, Support Vector Machines, and Genetic Programming. Special types of neural networks, called analytic neural networks, are used for nonlinear sensitivity analysis and variable selection [4]. The data set is further reduced by using SVM with selected level of complexity (% support vectors). As a result, only relevant variables and information-rich data points are used for GP model generation. In this way the efficiency of symbolic regression, which is computationally-intensive, is increased significantly. In addition, the probability for finding parsimonious solutions increases due to the reduced search space. The integrated methodology has been successfully implemented for several soft sensors on continuous industrial processes [4]. However, the increased requirements for robustness in batch processes require some improvements in the areas of proper model selection and using ensemble of predictors.

3.2 Pareto-Front Based Model Selection

Several thousand empirical models are generated in a typical GP run with at least 20 simulated evolutionary processes of 200 generations. Most of the generated models are with similar performance and proper model selection is non-trivial. The direct approach is to use the R^2-statistic as model selection criterion and to select the "best" model based on the fitness measure at the end of the run.

However, the fitness measure does not take complexity or smoothness of the function into account. Furthermore, it is possible that for a slight decrease in the measure a far less complex function may be obtained that may have higher robustness. For this the experience of the analyst is needed. Therefore it is necessary to extract a manageable number of models to inspect.

One indicator of the complexity of the models in a GP-run is the number of nodes used to define the model. The measure may be misleading for it does not discern between the types of operators used in each node. For example, no distinction is made between an operator that is additive and an operator that is an exponential function. Clearly there is a huge difference in complexity. However, using the number of nodes as an indicative measure can help reduce the number of models to inspect to a reasonable size.

In order to find the right trade-off between complexity and accuracy, the Pareto-front is constructed. The Pareto-front is a concept commonly used in multi-objective optimization [16]. In multi-objective optimization, apart from the solution space, which is constructed by the constraints in terms of the input variables, there is also an objective space. The objective space is a mapping of the solution space onto the objectives. In classical multi-objective optimization, the problem is cast into a single objective optimization problem by defining an *a priori* weighted sum. The solution to the single objective optimization problem is one point in the objective space. However, as the optimal weighted sum is seldom known *a priori*, it is often better to make the final decision from a set of solutions which is independent of the weights. This set of solutions is given by the Pareto-front. The Pareto-front thus represents a surface in the objective space of all possible weighted combinations of the different objectives that optimally satisfy the constraints.

Since the model selection task is in principle a multi-objective problem (i.e. accuracy vs. complexity), the fundamentals of the Pareto-front can be applied. Using the Pareto-front for GP-generated models has many advantages [17]. Firstly, the structural risk minimization principle [6] can be easily applied to GP-generated models. Secondly, it effectively displays the trade-off between the measures, which enables the analyst to make an unbiased decision. Thirdly, as only a small fraction of the generated models in GP will end up on the Pareto-front, the number of models that need to be inspected individually is decreased tremendously. Finally, additional considerations such as variety in input variables used for ensemble construction can be taken into account. For example, if a Pareto-optimal model uses an undesirable transformation or input variable, one could look for an alternative model among the models close to the Pareto-front.

In Fig. 1 the Pareto-front is displayed for a set of GP-generated models in terms of two objectives, ratio of nodes and R^2. The ratio of nodes is a measure of complexity and needs to be minimized. The second objective, R^2 is a measure of the performance of the models. Using 1-R^2 instead of R^2 allows easier interpretation as both objectives

are minimized. The Pareto-front models are models for which no improvement on one objective can be obtained without deteriorating another objective. The optimal model will therefore lie somewhere on the Pareto-front. Its position will depend on the problem at hand. For example, if the complexity and performance have equal importance then the optimal model would lie in the lower left corner of the Pareto front. This example shows how the structural risk minimization is used in finding the optimal model as well as the trade-off between complexity and accuracy.

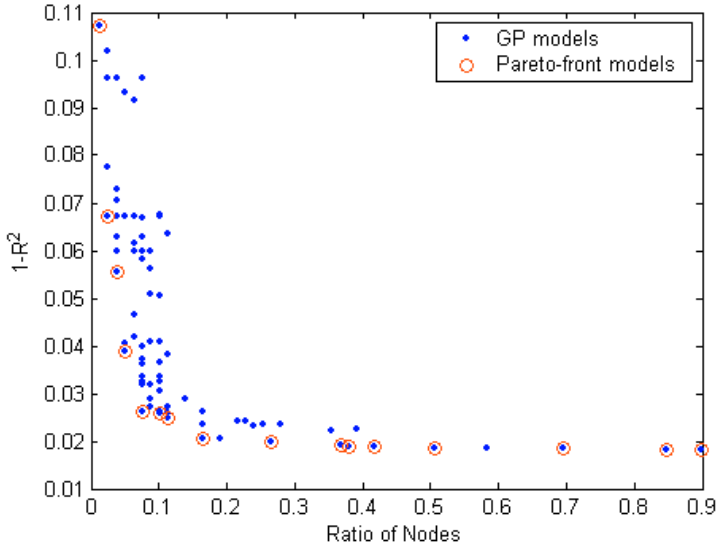

Fig. 1. Pareto-front based on the performance of the training data

Recall that another advantage of using the Pareto front is the reduced number of models to inspect. This is clearly seen in Fig. 1. In total there are 88 models depicted in the figure. However, only 18 of them are identified as Pareto-front models. Furthermore, one can clearly see that using models with a ratio of nodes higher than 0.3 does not result in a significant improvement of R^2. Therefore, the number of models to consider may even be less.

Finally, if the analyst is interested in a set of models to be used in the ensemble, the pool of interesting models can be easily extracted. It is possible to perform model selection in an interactive way such that the analyst can request a model's functional form, error statistics, and response surfaces by clicking on the particular model depicted in the figure.

3.3 Ensemble Design

It is often preferred to not to develop a single-model soft sensor, but a soft sensor that uses an ensemble of models. An ensemble consists of several models that will be used to predict future measurements. The average of the various models will be used as the final prediction.

One advantage of using an ensemble soft sensor is that the standard deviation of the different models in the ensemble can be used as model disagreement measure. This measure indicates how much the performance of the models differs from each other within a given estimate. The smaller the difference, the more certain the operators can be of the prediction made by the soft sensor. This indicator is of special importance for automatically detecting that the inferential sensor goes outside the training range. It is assumed that in the extrapolation mode the different models will diverge more strongly than in interpolation mode, i.e. the standard deviation of their predictions will significantly increase. There are several methods to design an ensemble of predictors [15]. The key issue is that one should be careful not to use models in the ensemble that are too similar in performance, because then a false sense of trust can be created as the standard deviation will be very small. On the other hand, the models should not be too different in performance, because then the predictive power of the soft sensor will be lost. Ideally, the selected models are diverse enough to capture uncertainties, but similar enough to predict well. The final selection of the models to be used in the ensemble depends mainly on the expertise of the analyst.

Another advantage of using an ensemble of models is that it enables redundancy. Since soft sensors are mainly used in processing conditions it often occurs that one or more of the instruments measuring the input variables can fail. If the ensemble consists of models that have different input variables, there will be another model available in the ensemble that still can predict. This prediction may not be the most accurate one, but at least there is a prediction instead of nothing.

In order to select the final models for the ensemble, we inspect the models on the Pareto-fronts based on their performance with both the training and test data. Model selection of the ensemble is based on the following error statistics: correlation coefficient, standard deviation, relative error, R^2-statistic, Root mean square error prediction (RMSEP) and the ratio of nodes. A matrix of variables used by the models is also taken into account. This is needed to identify models that don't use certain variables when ensembles with redundancy are constructed.

4 Development of Biomass Inferential Sensor for the Growth Phase in a Batch Fermentation Process

The described methodology for design of inferential (soft) sensors based on an ensemble of symbolic regression-type predictors will be illustrated with an industrial application of biomass estimator for the growth phase in a batch fermentation process at the Dow Biotech facilities in San Diego.

4.1 Data Collection and Pre-processing

The growth phase data were selected from eleven repeat fermentation runs on different 20L fermentors. Each experiment produced a batch of time series data that included seven input variables (air, pressure, OUR, time elapsed, agitation, nutrient, and total volume) and one output – Optical Density (OD), a measurement for

biomass. The selected inputs have been recommended by the process experts and were sampled every 15 min. The output measurement, which is highly correlated to the biomass, was sampled every two hours. The data from each batch was preprocessed in order to remove outliers, faults and missing values. From the eleven batches eight batches were chosen for training purposes and three batches were set aside for testing purposes.

4.2 GP Models Generation and Selection

The symbolic regression-type models have been derived on a MATLAB toolbox, developed internally in The Dow Chemical Company. The key parameters of a typical Genetic Programming run are given in Table 1. Several GP simulated evolution processes of 20 runs were made varying the values for the number of generations and the parsimony pressure.

Table 1. Genetic Programming Parameter Settings

Parameter	Value/Setting
Random subset selection [%]	100
Number of runs	20
Population size	100
Number of generations	30, 100
Probability for function as next node	0.6
Optimization function	Correlation Coefficient
Parsimony pressure	0.1, 0.08, 0.05
Probability for random vs. guided crossover	0.5
Probability for mutation of terminals	0.3
Probability for mutation of functions	0.3

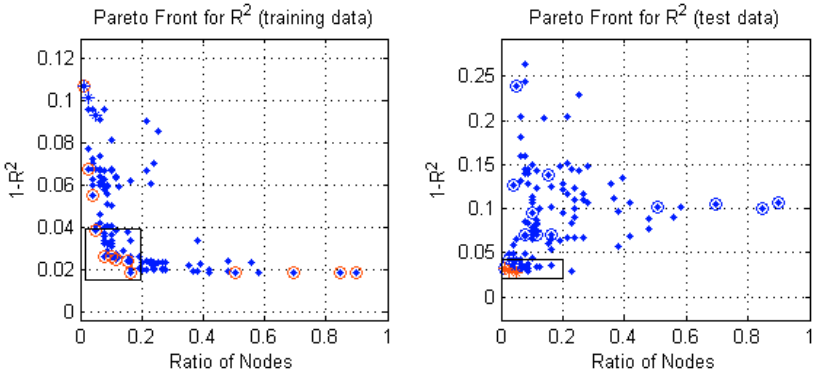

Fig. 2. Performance and Pareto-fronts for combined results of all GP runs

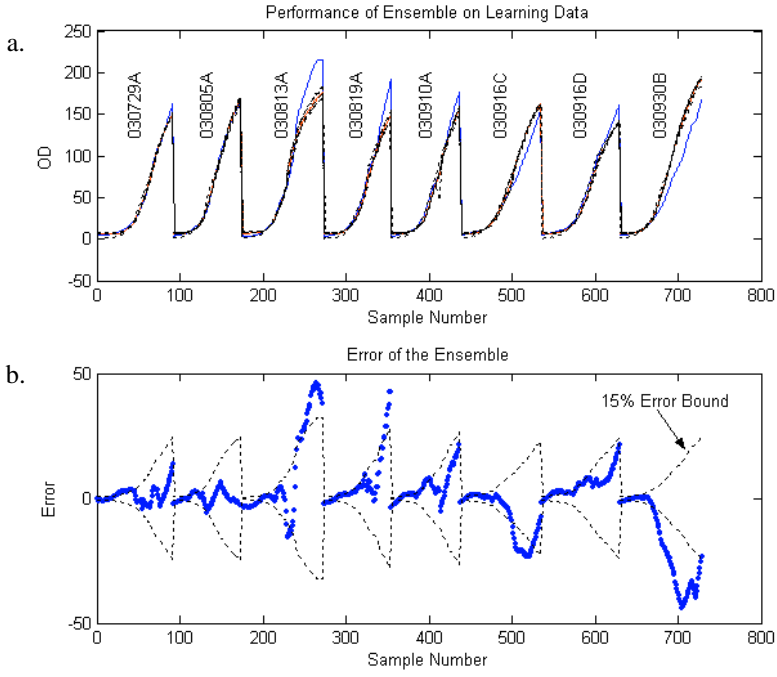

Fig. 3. Performance of the ensemble on the training data

The results from the different GP runs were combined and all duplicate models with equal symbolic strings were removed. In Fig. 2, the performance of the various models on the training and testing data are shown as well as the Pareto-fronts based on both data sets.

For the biomass inferential sensor, it was important that the models not be too complex and have an R^2–performance above 0.94. The models of interest were therefore those in the lower left corner of the Pareto-fronts of both training and test data. Furthermore, it was desirable that all the models have similar performance on the test data. The set of models that satisfy these conditions is within the compounding boxes of the training and test data, as shown in Fig. 2. The first set of the potential members of the ensemble were identified and inspected from the models that appear within both the boxes. Several models turned out to be duplicate models and were consequently removed. The final set of models for the ensemble is as follows:

$$f_1 = -116.471 + 329.8129\, e^{-e^{(-x_6)}}$$

$$f_2 = 7.4698 + 6.296 \frac{x_5 x_6}{{x_7}^2}$$

$$f_3 = 7.6537 + 343.9166\frac{x_2 x_6}{{x_7}^2}$$

$$f_4 = -37.7896 + 647.9714\, e^{-e^{\left(\exp(-x_6)+\exp\left(-\exp\left(x_7^{1/4}\right)\right)\right)}}$$

$$f_5 = -1.3671 + 0.12025\sqrt{x_6}\,(x_5 - x_7)$$

All of them are simple enough and provide significant diversity (in terms of selected inputs and nonlinear relationships) to allow a proper design of the ensemble of predictors. There is no direct physical or biological interpretation of the models but the experts agree that the most often used variable in all models x_6 is of critical importance to the cell growth.

4.3 Ensemble Performance

The prediction of the ensemble is defined as the average of the predictions of the five individual models obtained in the previous section. The accuracy requirements for the ensemble were to predict OD within 15% of the observed OD-level at the end of the growth phase. The performance of the ensemble on the training data used by GP can be seen in Fig. 3.

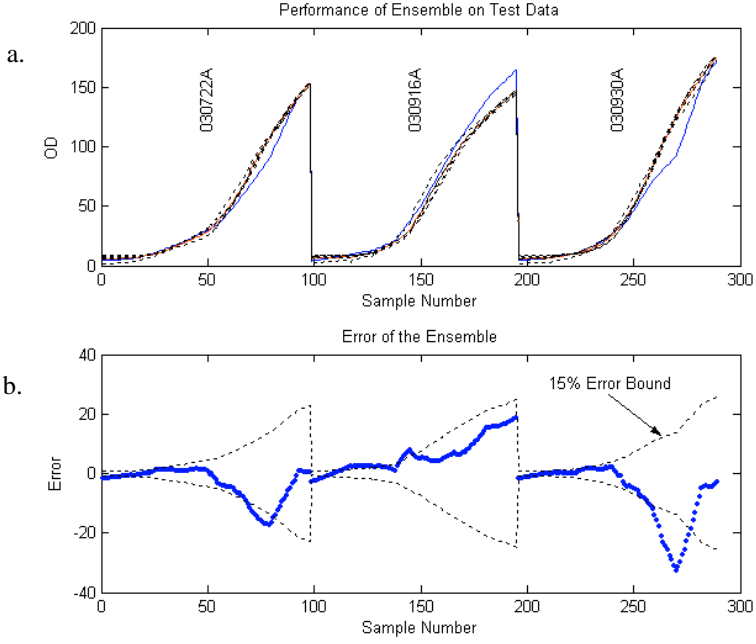

Fig. 4. Performance of the ensemble on the test data

In Fig. 3 (a) the OD-level is plotted against the sample number, which corresponds to the time from the beginning of the fermentation. The solid (blue) line indicates the observed OD-level. The dashed lines represent predictions of the individual models. The solid (red) line corresponds to the prediction of the ensemble. In Fig. 3(b) the residuals of the ensemble's prediction with respect to observed OD is shown. The 15% error bound is also shown. For the training data one sees that for three batches (030813A, 030819A and 030930B), the ensemble predicts outside the required accuracy. For batch 030813A it was known that the run was not consistent with the rest of the batches. However, this batch was added in order to increase the range of operating conditions captured in the training set.

The performance of the ensemble on the test data can be seen in Fig. 4. Again, in Fig. 4(a) the OD-level is plotted against the sample number for the observed data, the individual model predictions and the ensemble. In Fig. 4(b), the error with respect to the test data can be seen. We see that the performance of the ensemble at the end of the run for all the batches of the test data is within the required error bound.

5 Conclusions

In this paper we have shown a successful application of a novel type of biomass estimator based on GP. Furthermore, it is one of the rare applications of inferential sensors to batch processes.

We have also improved the model selection by using the Pareto-front approach. The main advantages of this approach are that the complexity of the generated models is taken into account and an unbiased decision is made. Furthermore, the number of interesting models to inspect manually is decreased to a manageable number.

Finally we have successfully implemented an ensemble-based inferential sensor of symbolic regression-generated functions for a notoriously difficult batch process.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, New York (1998)
2. Qin, S.: Neural Networks for Intelligent Sensors and Control - Practical Issues and Some Solutions. In: Neural Systems for Control, Academic Press, New York (1996)
3. Lennox B. G. Montague, A. Frith, C. Gent, V. Bevan: Industrial Applications of Neural Networks – An Investigation, Journal of Process Control, **11**, (2001) 497 – 507
4. A. Kordon, G. Smits, A. Kalos, E. Jordaan: Robust Soft Sensor Development Using Genetic Programming, In: R. Leardi (ed): Nature-Inspired Methods in Chemometrics, Elsevier,Amsterdam (2003)
5. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA (1992)
6. Vapnik, V.: Statistical Learning Theory, Wiley, New York (1998)
7. Kordon A., G. Smits: Soft Sensor Development Using Genetic Programming, Proceedings of GECCO'2001, San Francisco, (2001) 1346 – 1351
8. Kalos A., A. Kordon, G. Smits, S. Werkmeister: Hybrid Model Development Methodology for Industrial Soft Sensors, Proc. of the ACC 2003, Denver, CO, (2003) 5417-5422

9. Kordon A.K, G.F. Smits, E. Jordaan, E. Rightor, Robust Soft Sensors Based on Integration of Genetic Programming, Analytical Neural Networks, and Support Vector Machines: Proceedings of WCCI 2002, Honolulu (2002) 896 – 901
10. Di Massimo C. et al.: Bioprocess Model Building Using Artificial Neural Networks, Bioprocess Engineering **7** (1991) 77-82
11. Tham M, A Morris, G Montague, P Lant: Soft Sensors For Process Estimation and Inferential Control, J. Process Control **1** (1991) 3-14
12. Willis M et al.: Solving Process Engineering Problems Using Artificial Neural Networks, In: Mc Ghel, M. Grimble, and P. Mowforth (eds): Knowledge-Based Systems for Industrial Control, Peter Peregrinus, London, (1992) 123-142
13. Cheruy A.: Software Sensors in Bioprocess Engineering, Journal of Biotechnology **52** (1997) 193-199
14. Hodge D., L. Simon, M. Karim: Data Driven Approaches to Modeling and Analysis of Bioprocesses: Some Industrial Examples, Proc. of the ACC2003, Denver (2003) 2062-2076
15. Sharkey A. (Editor): Combining Artificial Neural Nets, Springer-Verlag, London (1999)
16. Deb K.: Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, Chichester UK (2001)
17. Bleuer S., M. Brack, L. Thiele, E. Zitzler: Multi-Objective Genetic Programming: Reducing Bloat bt Using SPEA-2, In Proceedings of CEC 2001, (2001) 536-543
18. Lampinen J. and A. Vehtari: Bayesian Approach for Neural Networks – Review and Case Studies, Neural Networks, **14** (2001 7-14