Automatic Linguistic Inversion of a Fuzzy Model for Fed-Batch Fermentation Control

Andri Riid
Department of Computer Control
Tallinn University of Technology
Ehitajate tee 5, Tallinn, 19086
Estonia
andri@dcc.ttu.ee

Ennu Rüstern
Department of Computer Control
Tallinn University of Technology
Ehitajate tee 5, Tallinn, 19086
Estonia
Ennu.Rystern@dcc.ttu.ee

***Abstract* – The goal of the paper is to analyze the potential of linguistic inversion of a fuzzy model to obtain a working controller, which is tested upon a benchmark fed-batch fermentation process. For the automation of the inversion procedure a simple decision mechanism is designed that dramatically increases the efficiency of the approach and results in improved performance compared to our earlier work. Moreover, automation of the decision process allows us to conduct a larger set of experiments to provide evidential material to the statement that approximate reasoning can produce surprisingly good results in the environment resembling real life conditions, however, these resukts also indicate that for guaranteed performance level, dynamics of the systems must be taken into account in the inversion process.**

I INTRODUCTION

In 1996, Industrial Control Centre of University of Webminster announced a Modeling and Control Competition. Incidentally, the number of submitted works that could be considered successful, was rather low [1, 2], however, the process simulation software that was available at the time, has become a benchmark in subsequent years and has been utilized in several instances [3-5], including a contribution of our own [6] where we suggested that fuzzy rules of the model (specifically identified from process data for that purpose) can reveal causal relationship between model variables and thus provide information upon which it is possible to design a working controller for the process by linguistic inversion. However, at this time, the inversion procedure had to be carried out manually mainly due to certain undesirable properties of applied fuzzy modeling algorithms, which called for human intervention to make meaningful decisions.

In current paper our aim is twofold. First, compared to [6] we apply a different, simple yet an efficient fuzzy modeling method known from literature [7] in order to improve credibility and interpretability of the process model. Secondly, we develop a decision making algorithm for model inversion so that in result, the whole controller design procedure can be fully automated, is exactly determined and therefore more efficient. As our simulations show, automation not only saves us a lot of time but also results in improved control performance. Comparison with other contributions is rather favorable considering the low computational cost of the approach. Additionally, a larger set of experiments is conducted in order to evaluate the potential of the approach and analysis of these results points to further issues that need to be solved.

II PLANT DESCRIPTION

The plant is supplied as a program that simulates a fed-batch fermentation process producing a secondary metabolite as the product. The microorganism in this process needs two substrates (s_1 and s_2) for growth and production and the process has two inputs, f_1 and f_2 in terms of substrate feed rates. There are five measurements that are x - biomass, s_1 - concentration of substrate 1, s_2 - concentration of substrate 2, p - product concentration and, V - volume. Maximum feed rates and the volume of the fermentor are limited ($f_{\max} = 50$, $V_{\max} = 4000$).

It is believed that the nominal profile as provided with the assignment

$$\begin{cases} f_1 = 10 + \dfrac{25}{(1+e^{5-0.1t})} \\ f_2 = 3.5 - \dfrac{3.5}{(1+e^{10-0.15t})} \end{cases} \tag{1}$$

is not good enough for the production. The optimal feed pattern should be investigated in order to improve the process productivity. The criterion J may be selected as:

$$J = pV/T, \tag{2}$$

where T is the duration of fermentation. Other process environment variables such as temperature and pH are assumed to be constant (at theirs optimum).

From the long experience it is known that feed of substrates at too high or too low levels reduces the production. Feed rate f_2 seems to be complementary with feed rate f_1. Only feeding of s_2 does not yield any production while only feeding of s_1 yields small product.

The model representing the plant is not given explicitly (to make the exercise comparable to working on a practical plant) but supplied as a black box with a set of specified inputs. There are following features:

i. The initial state varies randomly within a subspace for each batch;
ii. The parameters of the model vary within specified limits;
iii. A set of non-measurable disturbances;
iv. A set of constraints.

Due to these features, it is impossible to obtain the same output series for each batch. This is more realistic to the practical situation. It is not quite common in real life though that all process variables can be measured on-line without problems.

III LINGUISTIC INVERSION TECHNIQUES

Arguably, the ultimate goal of controller design is to derive the inverse model of the process. In theory, the use of an inverse model possesses the advantages of open-loop control, i.e. inherent stability and perfect control with zero error. In practice, however, it is not guaranteed if the inverse configuration actually exists or if it is physically realizable. *Global inversion* of the system, where all states become the outputs of the inverted model and the output of the original system becomes the state variable (Fig. 1) has normally non-unique solution and must be given by a family of solutions. In case of *partial inversion*, only one of the states (x_1 in Fig. 1) of the original system becomes the output of the inverted model and other states together with the original output are the inputs of the inverted model. Partially inverted model can be also more easily embedded into the control system than the global inversion.

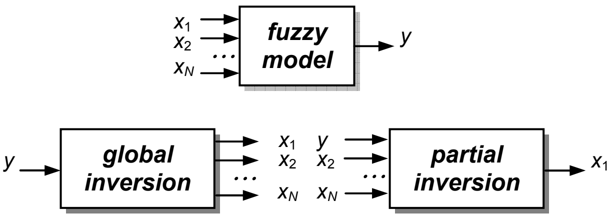

Figure 1 Fuzzy model and its inversions

The partial inversion has often an unique solution but not necessarily (the original model must be strictly monotone in respect to the inverted state to be invertible).

Numerical identification (e.g. in the form depicted in Fig. 2) of the inverse model may be computationally expensive, requiring many training epochs/samples to converge. The issue of invertibility is also of importance and not very well handled with automatic generation of the inverted model. The techniques for training the inverted fuzzy model have become known basically through neural network research. Fuzzy systems, however, can be considered different from neural networks because: (a) they can be interpreted in linguistic terms; (b) if transparent [8], their parameters can be interpreted in terms of their influence to the input-output relationship and therefore allow approximate linguistic inversion [9-11] and exact analytical inversion [12].

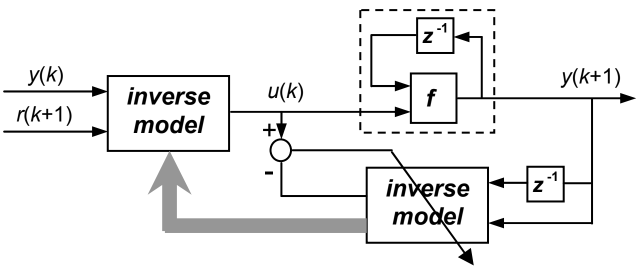

Figure 2 Fig. 2. Online inverse model based control

Linguistic inversion (causality inversion), in what we are interested in current paper, is obtained through the exchange of antecedent and consequent variables in fuzzy rules if a symmetrical operator (t-norm) represents the if-then relation. Consider a three-input fuzzy system from [12], given in Table 1.

Table 1
Rule base of the sample model

	x_1 is *small*	x_1 is *medium*	x_1 is *large*
x_2 is *low* AND x_3 is *low*	*zero*	*low*	*medium*
x_2 is *low* AND x_3 is *high*	*low*	*medium*	*high*
x_2 is *high* AND x_3 is *low*	*low*	*medium*	*high*
x_2 is *high* AND x_3 is *high*	*medium*	*medium*	*high*

The inversion procedure may have three possible results marked in Table 2:

i. The input configuration is unique. This is the ideal case.
ii. The input configuration is non-unique, meaning that the rule base is non-invertible. The approximate solution is to choose the input configuration with the lowest control energy (in linguistic sense).
iii. There are no inputs that allow one-step transition to the desired output. The reason why such situation occurs is twofold. First, the number of linguistic labels ("low", "high", etc.) given for x_1 and y is not equal, i.e. there are 16 rules in the inverted model whereas the number of rules in the original model is 12. The second reason is that the original system may not simply allow one-step transition to the desired output from the given state. The approximate solution is to choose the "nearest" output (again in linguistic sense).

Table 2
Inverted rule base

x_2, x_3	y is *zero*	y is *low*	y is *medium*	*y is high*
x_2 is *low* & x_3 is *low*	*small* (i)	*medium* (i)	*large*(i)	*large*(iii)
x_2 is *low* & x_3 is *high*	*small*(iii)	*small* (i)	*medium* (i)	*large*(i)
x_2 is *high* & x_3 is *low*	*small*(iii)	*small* (i)	*medium* (i)	*large* (i)
x_2 is *high* & x_3 is *high*	*small*(iii)	*small*(iii)	*small*(ii)	*large* (i)

Besides, one must pay attention to the membership functions (MFs) of the inverted model that correspond to linguistic labels. Input and output MFs may be of different type (e.g. triangular input and singleton output MFs) and therefore singleton MFs need to be converted to triangular ones or vice versa. Even if both input and output MFs are of triangular type attention must be paid because of contradictive nature of input and output transparency constraints [8] (unless they are uniformly distributed on all variables).

For the reasons described above it may become very difficult to obtain a working linguistic inversion in practice, which is probably the main reason why the approach has found little use. However, limited version of the approach assuming a fixed control goal (pre-determined profile of y) can be performed with less effort and allows us to exclude y from the inverted model. From the rules that have linguistically identical premises in terms of x_2 and x_3 (i.e. rows in Table 1) only one that corresponds to the desired label of y is selected for inversion. E.g., if we fix the desired output profile as a sequence of y is {*low*, *medium*, *high*, *high*} (emphasized with gray background in Tables 1 and 2), the inverted model would appear as

IF x_2 is *low* AND x_3 is *low* THEN x_1 is *medium* [(i)]
IF x_2 is *low* AND x_3 is *high* THEN x_1 is *medium* [(i)]
IF x_2 is *high* AND x_3 is *low* THEN x_1 is *large* [(i)]
IF x_2 is *high* AND x_3 is *high* THEN x_1 is *large* [(i)]

It is easy to see the advantages of this approach

i. The number of rules of the inverted model is S_y times smaller than in the case of the original partial linguistic inversion (S_y is the number of linguistic labels of y).
ii. Type of consequent MFs does not present a problem, as they are never inverted. Moreover, models where each rule has an unique output MF (in fact the model we will identify in section IV is of similar type) would be extremely difficult to invert linguistically, because of the large number of consequent MFs.
iii. Less problems with [(ii)] and [(iii)] type rules.

However, the price of this simplification is that each time the desired output profile is decidedly changed, the model must be re-inverted again.

IV MODELING ALGORITHM

For inversion a suitable fuzzy model of the fermentation process is required. In this paper we have used a fuzzy model belonging to a class of 0^{th} order Takagi-Sugeno systems.

$$\begin{array}{c}\text{IF } x_1 \text{ is } A_{1r} \text{ AND } \ldots\ x_i \text{ is } A_{ir} \ldots \text{ AND } x_N \text{ is } A_{Nr} \\ \text{THEN } y_1 = b_{1r} \text{ AND } \ldots\ y_j = b_{jr} \ldots \text{ AND } y_M = b_{Mr}\end{array} \tag{3}$$

where A_{ir} denotes the linguistic label of the i^{th} input variable x_i and b_{jr} is the consequent singleton value for output variable y_j, respectively, associated with the r^{th} rule ($i = 1 \ldots N, j = 1 \ldots M, r = 1 \ldots R$).

0^{th} order TS systems possess a computationally inexpensive inference algorithm that gives numerical relationship between the system (3) variables

$$y_j = \sum_{r=1}^{R} \tau_r b_{jr} \Big/ \sum_{r=1}^{R} \tau_r \ , \ j = 1, \ldots, M, \tag{4}$$

where τ_r is the activation degree of the r^{th} rule, given by

$$\tau_r = \bigcap_{i=1}^{N} \mu_{ir}(x_i)\,, \tag{5}$$

where μ_{ir} denotes the membership function MF of the i^{th} input variable (representing A_{ir}) associated with the r^{th} rule.

In current application our primary concern is model ability to reveal valid causal relationships between system variables and therefore both transparency and reliability of the model are important requirements. Transparency (validity of linguistic interpretation) has been discussed extensively in our previous works [8] and it has been found out that transparency can be preserved by imposing certain restrictions to MF parameters. Things are not so simple with reliability, which is sometimes rather difficult to evaluate. Often fuzzy model is identified from scarce (and possibly noisy) data where the modeling algorithm has to generalize on the basis of existing samples. This situation is quite common in practical applications because it is usually difficult to get good coverage of the input space as the number of inputs increases (it also may be difficult to obtain adequate validation data sets).

Fuzzy modeling methods (neural network inspired learning algorithms, in particular) generally rely on global learning techniques driven by numerical approximation error and tend to obtain the parameter values by drawing conclusions through the extrapolation of existing data samples often resulting in fuzzy rules that are unrealistic or simply untrue for the given application.

Our answer to this problem is to compute consequent parameters with the simple method (6) proposed by Nozaki *et. al.* in [7]. Given a set of training data [$x_1(k)$, $x_2(k)$, …, $x_i(k)$, …, $x_N(k)$, $y_1(k)$, $y_2(k)$, …, $y_j(k)$, …, $y_M(k)$], $k = 1, \ldots, K$, the consequent parameters of the model can be computed by

$$b_{jr} = \sum_{k=1}^{K} \tau_r(k) y_j(k) \Big/ \sum_{k=1}^{K} \tau_r(k)\,,\ j = 1, \ldots, M\ . \tag{6}$$

The basic important characteristic of (6) is that consequent parameters for a given rule are computed as the weighted average of relevant (measured by rule activation degree τ_r in (6)) output samples that gives the algorithm interpolating rather than extrapolating character, which improves rule credibility.

The drawbacks of Nozaki's method are its relatively modest numerical accuracy and the assumption that input MF parameters are available (otherwise it is impossible to calculate τ_r for (6)). Moreover, no known automatic input MF identification procedure guarantees us optimal results, so in current paper we typically rely on expert-defined or template input partitions that satisfy

$$\forall x_i \in X_i : \sum_{s=1}^{S_i} \mu_i^s(x_i) \le 1\,, \tag{7}$$

that is the transparency condition for input MFs of the model (output MFs are transparent by default in 0^{th} order TS architecture), where $\mu_i^s(x_i)$ is the s^{th} MF defined for i^{th} input variable.

V TECHNICAL DETAILS

Controller design is a two step procedure (the transformation sequence is depicted in Fig. 3). First, fuzzy model having the structure [Δx, Δp] = $f(x, s_1, s_2)$ is identified from measured (and preprocessed) data from 10 process runs with nominal feeds (similar approach in respect to data acquisition was taken in [4] and more details about this and the reasoning behind the model structure can be found from [6]), where Δx and Δp are the biomass and product concentration growth rates, respectively ($\Delta x = x(k) - x(k-1)$, $\Delta p = p(k) - p(k-1)$, $k = 2, \ldots, K$). The model can be validated on a separate set of validation data.

In existing applications of linguistic inversion, the following step - controller extraction - is typically a manual task. For example, in [6], first, all rules of the model were grouped into S_x subsets according to the biomass label in the antecedent part of the rules (S_x is the total number of MFs (or linguistic labels) defined for biomass). From each such group, containing the rules of format:

$$\begin{array}{c} \text{IF } x \text{ is } A_{1r} \text{ AND } s_1 \text{ is } A_{2r} \text{ AND } s_1 \text{ is } A_{3r} \\ \text{THEN } \Delta x \text{ is } b_{1r} \text{ AND } \Delta p \text{ is } b_{2r}, \end{array} \tag{8}$$

one rule was selected, based on the values of b_{1r} and b_{2r} (that serve as the estimates of growth rates of x and p provided by the levels of biomass and substrate concentrations that trigger this rule) and rewritten in the format

$$\text{IF } x \text{ is } A_{1r} \text{ THEN } s_1 = p_{1r} \text{ AND } s_2 = p_{2r}\,, \tag{9}$$

where p_{1r} = core(A_{2r}), p_{2r} = core(A_{3r}). Manual approach in [6] was further necessitated by the properties of adopted modeling algorithms, e.g. with one of the methods – fuzzy template modeling algorithm [13] - each rule was assigned a rule weight which made interpretation of rules difficult at times even for an human eye and with Gustafson-Kessel clustering in combination with the least squares method [14], many clearly unrealistic b_{1r} and b_{2r} turned up in identification process. With the current method (6), the rules are much more interpretable and reliable therefore rule selection can be supported by a simple decision mechanism (Fig. 4). All numerical values of (b_{1r}, b_{2r}) corresponding to the rules of one subset are fed into respective inputs of the decision making mechanism that computes the corresponding pair (τ_1, τ_2). As our previous work has shown, the process can be divided into two phases: biomass formation phase in which biomass is grown to the level it is capable to give high production rates and production phase in which product concentration rapidly increases. It is easy to see that rules with a high value of τ_1 have a high potential for being selected for production phase and that rules with a high value of τ_2 would be good candidates for biomass formation phase.

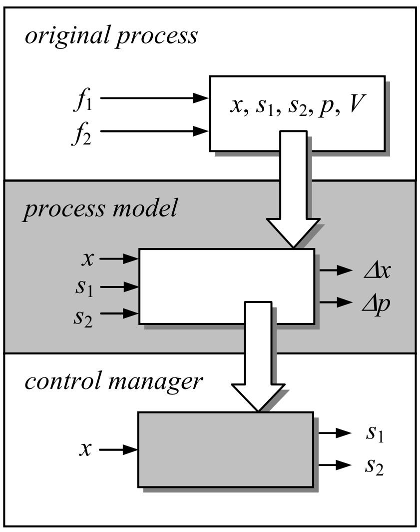

Figure 3 Controller design procedure

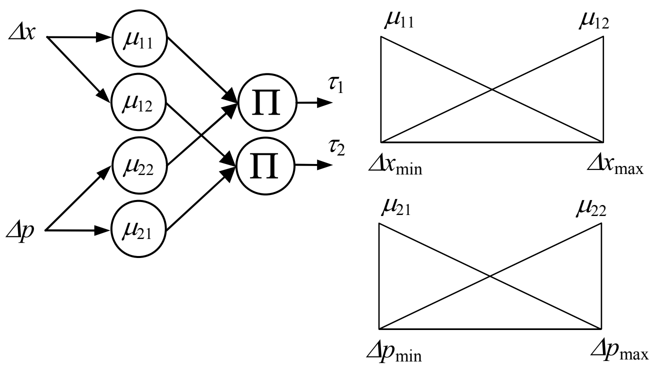

Figure 4 Decision making mechanism and its parameters

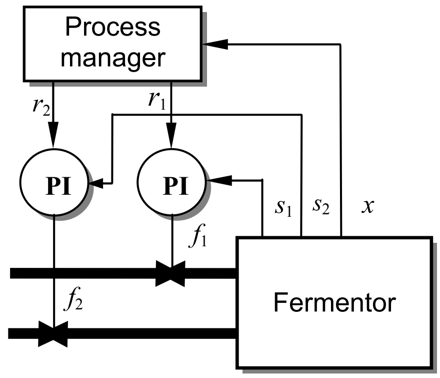

Figure 5 The control system

Of the subset of rules the one with maximum τ_1 or τ_2 (whichever happens to be larger) is declared the winner and chosen for inversion. Note that this way the principal decision of choosing production or biomass growth phase will be determined automatically. After that we proceed with the next subset of rules until we have analyzed the last one.

Consequently, this simple mechanism allows us to go through the whole sequence in Fig. 3 without human intervention. The resulting controller or process manager will then be embedded into the control system (Fig. 5).

VI RESULTS

First, the new approach is tested with the input partition depicted in Fig. 6 that belongs to one of the successful controllers from [6]. Application of (6) results in a 38–rule (so few remain of 140 potential rules after filtering out the unused ones based on the cumulative value of τ_r computed on training data) model that has root-mean-squared-error (RMSE) 10.75 and 0.18 for Δp and Δx, respectively. On a separate set of validation data of similar size, RMSE values are 5.54 and 0.04, indicating that model generalization properties are rather good and the model is thus reliable. In the next step the model is inverted automatically into a four-rule process manager Averaged J (J_{av}) over ten process runs obtained with this controller as well as typical process duration (T_{av}), average product concentration (p_{av}) and improvement (↑) in respect to nominal profiles are given in Table 3 along with the results from other sources [1, 4-6]. Moreover, in Fig. 7, four time diagrams of J_{av} are depicted, where it is demonstrated that our controller is quite similar in character to the one from [4]. It must taken into account, however, that other authors' contributions violate at least one of the premises of the original task, e.g. [1, 5] ignore the requirement that fermentation process should be stopped at the moment when fermentor volume reaches 4000 (violation of this condition means that we are dealing with a continuous fermentation process rather than a fed-batch one). So, when taking the volume limitation into account the true J_{av} of [1] would be slightly lower (about $9.2\cdot10^4$) than the reported one and in [5] the actual outcome would be below $3\cdot10^4$, which, incidentally, is just a failure. The application [4] on the other hand makes explicit use of equations hidden in the black-box model and has therefore access to supposedly forbidden information.

Table 3
Comparison of results

	nominal	DISOPE[1]	Riid [6]	Zhang[5]	Liang [4]	current
J_{av}	$8.33\cdot10^4$	$9.5\cdot10^4$	$1.02\cdot10^5$	$1.07\cdot10^5$	$1.53\cdot10^5$	$1.32\cdot10^5$
T_{av}	116	120	122	120	74	82
p_{av}	2415	2850	3110	3210	2829	2714
↑	0%	14%	22.5%	28.5%	83%	59%

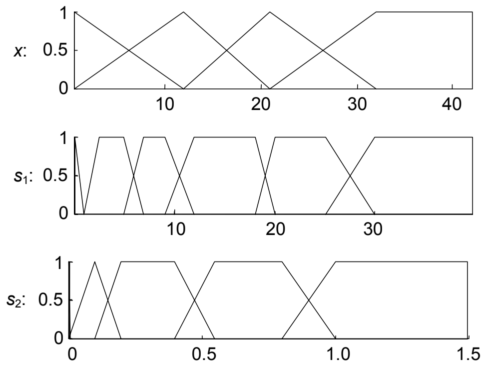

Figure 6 Input partition of the fuzzy model (from [6])

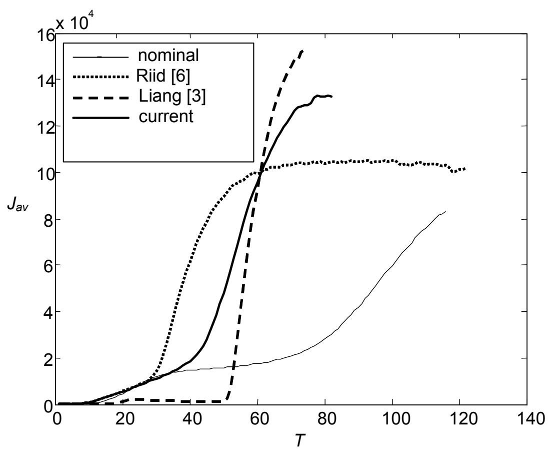

Figure 7 Control results

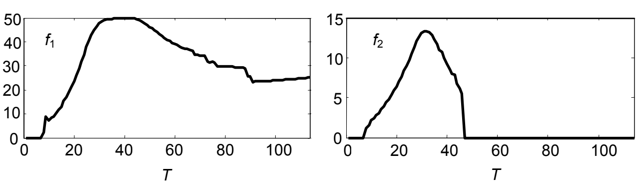

Figure 8 Averaged feed profiles

We can see that in these circumstances our approach compares rather favorably; however, the variation of these preliminary results is rather high. E.g. process time varies from 74 to 114 time units and product concentration from 2000 to 3000. In order to fix that, averaged feed profiles (Fig. 8) on the basis of the feed profiles provided by the control system (Fig. 5) from 10 runs are used rather than applying the control system directly, which leads to much more consistent (as can be seen from Fig. 9) and further improved ($J_{\text{av}} = 1.38\cdot10^5$, $T_{\text{av}} = 83$, $p_{\text{av}} = 2862$) results.

Secondly, and even more importantly, automation of the inversion process opens the way for large-scale analysis of the applicability of linguistic inversion for the given problem. For this purpose we have constructed 180 different process models with varying number of input MFs, ranging from 4 to 8 in case of biomass concentration and from 5 to 10 in case of those of substrates, including all possible combinations. For input partitions we assume that smaller concentrations of substrates need to modeled with greater care (it is unlikely to maintain higher concentrations in the second phase of the process as the volume grows), therefore uneven (logarithmic) distribution of MFs was used for these variables (Fig. 10), where the center b_i^s of s^{th} input MF of i^{th} input variable can be calculated as

$$b_i^s = x_i^{\min} + (x_i^{\max} - x_i^{\min})((s-1)/(S_i-1))^m,\ s = 1,\ldots,S_i\,. \quad (10)$$

The results of this experiment set are depicted as histograms of J_{av}, p_{av} and T_{av} (Fig. 11).

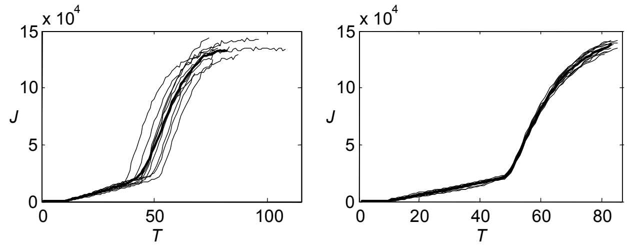

Figure 9 Variance of results. Left: productivity with the control system from Fig. 5. Right: productivity with averaged feed profiles

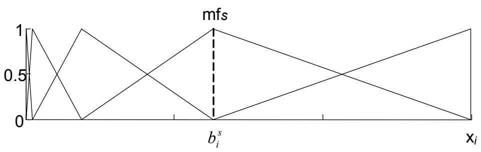

Figure 10 Logarithmic MF distribution ($m = 3$)

It turns out that only 57% of these resulting controllers outperform the nominal controller in terms of J_{av} (average productivity over all experiments is $8.95\cdot10^4$). However, at the closer look over 90% of process managers are able to bring p to 2500 and higher so in general the inversion fulfills its original goal (there are just a couple of downright failures where product concentration reaches nowhere). Apparently, the main problem and the reason behind low productivity is that computed feed profiles take generally a long time to raise the fermentor volume to 4000 (over 21% of them require more than 200 time units for that!).

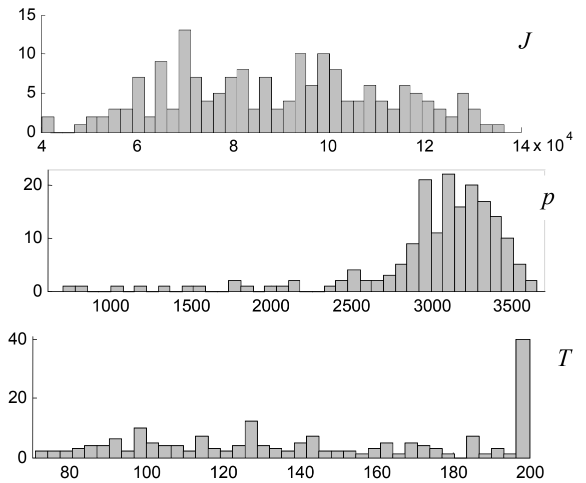

Figure 11 Summary of experimental results

In the analysis of the results, one must first acknowledge that in order to detect important dependencies between system variables such dependency must first be present in training data, which ultimately, imposes certain upper limit to controller performance. Second factor that must be taken into account is that whole approach is highly approximate by nature. Third, high values of T must not come as a complete surprise because, in fact, identified model contains no information of the dynamics of the process (except perhaps in indirect way). Another issue is if we are able to find or develop a method that would assist us in placing input MFs so that the final outcome would be favorable, because, apparently, it is not always feasible to conduct a huge set of trial-and-error experiments. It must be noted that presently, it is rather impossible to detect any correlation between the number of MFs or their positions and controller performance despite a large number of different controllers that were tested. Summarizing, it is actually quite remarkable that the resulting controllers are that good, taking into account highly approximate (model has approximation error, linguistic inversion is approximate and even MF conversion following (9) is of approximate nature) nature of the approach, limited amount of information (training data) and highly stochastic working environment.

VII CONCLUSIONS

In current paper linguistic inversion based control method for benchmark fed-batch fermentation has been investigated. With the proper choice of the modeling algorithm and preserving model transparency we have managed to improve model reliability so it can be inverted automatically using a simple decision mechanism. Automated decision-making in turn leads to better performance despite a very approximate nature of the whole approach. However, as the original assignment makes assumptions about process dynamics, it is recommended for guaranteed performance level that information about process dynamics must be included in the process model and utilized in inversion procedure as the performance test with a large family of controllers clearly indicates. This, along with a few minor issues remains a matter of future research.

REFERENCES

[1] B. McKay, C.S. Sandersson, M.J. Willis, J.P. Barford, G.W. Barton, "Evolving a hybrid model of a fed-batch fermentation process," *Trans. Inst. MC*, Vol. 20, No. 1, pp. 4-10, 1998

[2] V. M. Becerra, P. D. Roberts, "Application of a novel optimal control algorithm to a benchmark fed-batch fermentation process," *Trans. Inst. MC*, Vol. 20, No. 1, pp. 11-18, 1998

[3] M. Soufian, M. Soufian, "Parallel genetic algorithms for optimised fuzzy modelling with application to a fermentation process," in *Proc. 2nd Int. Conf. on Genetic Algorithms In Engineering Systems: Innovations and Applications*, pp. 123-128, 1997

[4] J. Liang, Y. Chen, "Optimization of a fed-batch fermentation process control competition problem using the NEOS server," *Proc. Inst. Mech Engrs Part I: J. Systems Control Eng.*, Vol. 217, Part 5, pp. 427-432, 2003

[5] H. Zhang, B. Lennox, "Multi-way optimal control of a benchmark fed-batch fermentation process," *Trans. Inst. MC*, Vol. 25, No. 5, pp. 403-417, 2003

[6] A. Riid, E. Rüstern, "Supervisory fed-batch fermentation control on the basis of linguistically interpretable fuzzy models" *Proc. Estonian Acad. Sci. Eng.*, Vol. 6, No. 2, pp. 96-112, 2000

[7] K. Nozaki, H. Ishibuchi, H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets and Systems,* Vol. 86, pp. 251-270, 1997

[8] M. Braae, D.A. Rutherford, "Theoretical and linguistic aspects of the fuzzy logic controller," *Automatica*, Vol. 15, No. 5, pp. 553-577, 1979

[9] C. Fantuzzi, "Linguistic rule synthesis of a fuzzy logic controller," in *Proc.IECON'94*, pp. 1354-1358

[10] C. Raymond, S. Boverie, A. Titli, "Fuzzy multivariable control design from the fuzzy system model," in *Proc. of the IFSA World Congr.* Sao-Paulo, Brazil, pp. 509-511, 1995

[11] P. Baranyi, I.M. Bavelaar, R. Babuska, L.T. Kóczy, A. Titli, H.B.Verbruggen, "A method to invert a linguistic fuzzy model," *Int. J Systems Science*, Vol. 29, No. 7, pp. 711-721, 1998

[12] A. Riid, E. Rüstern, "Transparent fuzzy systems in modeling and control" in *Interpretability Issues in Fuzzy Modeling*, J. Casillas, O. Cordon, F. Herrera, L. Magdalena, Eds. New York: Springer, 2003, pp. 452-476

[13] R. Yager, D. Filev, *Essentials of Fuzzy Modeling and Control*, New York: John Wiley and Sons, 1994

[14] M. Setnes, R. Babuska, H.B. Verbruggen, "Rule based modeling: precision and transparency," *IEEE Trans. Systems, Man Cybern.*, Part C, Vol. 28, No. 1, pp. 165-169, 1998