

A Method for Heuristic Fuzzy Modeling in Noisy Environment

Andri Riid

Laboratory of Proactive Technologies
Tallinn University of Technology
Ehitajate tee 5, Tallinn 19086, Estonia
Telephone: +372 6202 108
Email: andri@dcc.ttu.ee

Ennu Rüstern *Member, IEEE*

Department of Computer Control
Tallinn University of Technology
Ehitajate tee 5, Tallinn 19086, Estonia
Telephone: +372 6202 104
Email: ennu.rystem@dcc.ttu.ee

Abstract—This paper presents a fully automatic algorithm for fuzzy model identification that pays attention to the interpretability and reliability of the model and is particularly suitable for working in difficult conditions where data may be both noisy and corrupted. The working principles and essential characteristics of the algorithm are explained on the basis of simple examples, its approximation properties are tested on Box-Jenkins data set and its application to fed-batch fermentation process demonstrates that in conditions resembling real life it can take full responsibility for the modeling task in modeling-for-control methodology.

I. INTRODUCTION

Though many applications of fuzzy logic use it as the underlying logic system for expert systems (exploiting the ability of fuzzy logic to capture the inexact and approximate nature of human knowledge) and as a man-machine interface to make accumulated human expertise available for the applications in different fields of automated decision-making (e.g. control), the applications based on the reverse ability of fuzzy systems - to explain system behavior in linguistic terms and reveal important system characteristics in a convenient manner - are much less frequent for a number of reasons.

First, data-driven fuzzy identification, as it is applied today, is primarily a black-box technique - fuzzy system configuration employed most frequently belongs to the class of 1-st order Takagi-Sugeno systems [1], with admittedly poor interpretation values - and the identification techniques available generally ignore the semantic aspect of fuzzy systems. The resulting fuzzy systems are non-transparent to interpretation or the interpretation of extracted fuzzy rules would lead to erroneous conclusions.

Even if this is not the case (interpretability preservation related research is quite a hot topic now [2]), interpretability is only a measure of system's internal consistency [3] and does not give any guarantee that the model is consistent with available data (its external consistency). The primary measure of external consistency is the approximation error but it is just one side of the coin, the other being generalization accuracy (here simply termed as reliability) and difficulties arise when a fuzzy system is identified from scarce data and incorporates generalizations based on existing samples. This situation is quite common in practice because it is usually

difficult to get good coverage of the input space by training data as the number of inputs increases. The generalization accuracy of the model depends on the distribution of training data but also on how the identification algorithm utilizes the parameters of the system. Fuzzy identification methods (neural network inspired learning algorithms [4] in particular), however, generally rely on global learning techniques driven by numerical approximation error and tend to obtain the missing rules by drawing conclusions by the extrapolation of existing data samples often resulting in fuzzy rules that are unrealistic or simply untrue for the given application, interpretation of which would lead to invalid conclusions. For example, least squares estimation, commonly used for obtaining consequent parameters of Takagi-Sugeno (TS) systems, has such properties.

It has been suggested [5] that in situations where we are primarily interested in the validity of interpretation and analysis of fuzzy rules, simple heuristics-based or heuristic-fuzzy methods may be more fruitful than numerically superior and much more popular neuro-fuzzy algorithms. In current paper, the method proposed in [5], essentially made up of previously published methods [6], [7] is given a further treatment so as to make it insensitive to noise and it is shown that the application of it extracts meaningful and valid rules from identification data, even if the data set contains noise and erroneous samples, resulting in interpretable, reliable and sufficiently accurate models.

II. PRELIMINARIES

Consider a multi-input multi-output fuzzy system, consisting of R rules:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{1r} \text{ AND } x_2 \text{ is } A_{2r} \text{ AND } \dots \\ &\quad \dots \text{ AND } x_N \text{ is } A_{Nr} \text{ THEN} \\ &y_1 = b_{1r} \text{ AND } y_2 = b_{2r} \dots \text{ AND } y_M = b_{Mr} \\ &\quad \text{OR } \dots, \end{aligned} \quad (1)$$

where A_{ir} denote the linguistic labels of the i -th input variable ($i = 1, \dots, N$) and b_{jr} are the output singletons of the j -th output variable ($j = 1, \dots, M$), associated with the r -th rule.

Each A_{i_r} has its representation in the numerical domain - the membership function μ_{i_r} . The inference function that computes the crisp output $y_j(k)$ of the system (1) corresponding to a given input vector $\mathbf{x}_k = [x_1(k), x_2(k), \dots, x_N(k)]$ appears as

$$y_j(k) = \frac{\sum_{r=1}^R \tau_r(k) \cdot b_{jr}}{\sum_{r=1}^R \tau_r(k)}, \quad (2)$$

where $\tau_r(k)$ is the activation degree of the r -th rule.

$$\tau_r(k) = \prod_{i=1}^N \mu_{i_r}(x_i(k)). \quad (3)$$

Despite having all properties of a linguistic (Mamdani) fuzzy system, (2) is typically called a fuzzy singleton or a 0-th order Takagi-Sugeno system.

In the following we assume that the membership functions of (2) are given by

$$\mu_i^s(x_i) = \begin{cases} \frac{x_i - a_i^{s-1}}{a_i^s - a_i^{s-1}}, & a_i^{s-1} < x_i < a_i^s \\ \frac{a_i^{s+1} - x_i}{a_i^{s+1} - a_i^s}, & a_i^s < x_i < a_i^{s+1} \\ 0, & a_i^{s+1} \leq x_i \leq a_i^{s+1} \end{cases}, \quad (4)$$

that satisfies input transparency condition assumed for correct interpretation of fuzzy rules (see [8] for further details).

III. THE ALGORITHM

Arguably, good learning schemes should be able to place optimal lone rules so that they cover the extremes or bumps of the approximand and then fill in between with extra rule patches if the rule budget allows [9]. The method by Nakoula et al. [6] that serves as the basic building block of the proposed approach, is principally an implementation of this strategy and places the rules iteratively at the locations in the input space responsible for maximum local error. The algorithm consists of following steps:

- Initialization. For each input variable x_i , two MFs are placed at the extremes of its domain (x_i^{min}, x_i^{max}). This is followed by rule generation phase where 2^N rules containing all possible unique input MF combinations are created and b_{jr} in these rules (1) are given the values of output readings $y_j(k)$ that correspond to the sample $\mathbf{z}_k = [x_1(k), \dots, x_i(k), \dots, x_N(k), y_1(k), \dots, y_j(k), \dots, y_M(k)]$ that provides the maximum value of (3) for the given rule.
- At l -th iteration the absolute value of approximation error $\epsilon(l)$ is computed over the training data set and the rule node, the sample $\mathbf{z}_k(l)$ responsible for $\max(\epsilon(l))$ is identified. The input coordinates of the rule node $[x_1(k), \dots, x_i(k), \dots, x_N(k)]$ are used as the centers (a_i^s in (4)) of MFs added in this step (one per each input variable) and the MFs in the immediate neighborhood of added MFs are updated to preserve (4). The existing rulebase is then revised - all consistent rules that can be formulated on the basis of updated partition are added to the rule base (unless $\sum_{k=1}^K \tau_r(k) < \tau_{min}$ for

the given (r -th) rule, where τ_{min} is a prespecified rule filtering threshold, which excludes the rules that have little evidence in training data. This is followed by another iteration until we feel like calling it a day for one reason or another (approximation error is low enough, there are enough rules and MFs already or there is no further improvement).

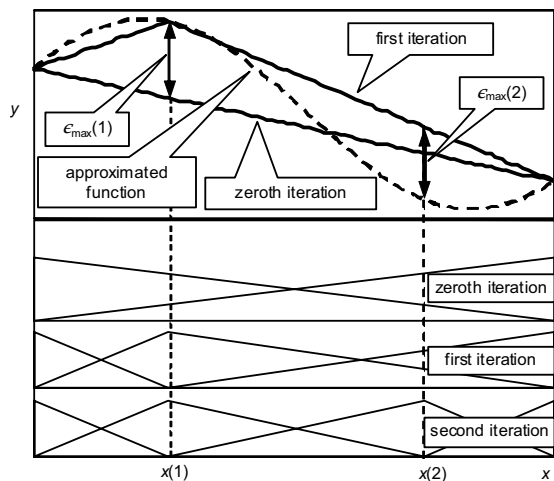


Fig. 1. First few iterations of Nakoula's algorithm.

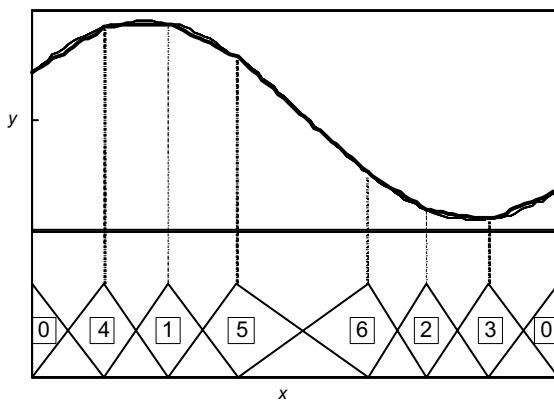


Fig. 2. Final approximation result with 8 fuzzy rules. Numbers on MFs indicate the order in what they were generated.

For illustration, an example of approximation of a single-input-single-output function is depicted in Fig. 1. As the final result in Fig. 2 demonstrates, what we have here is a very simple yet clever and also computationally cheap method that can produce a reasonable approximation just in a few iterations.

Approximation in higher order input spaces can be similarly be very effective. Consider a two-input-single output fuzzy system depicted in Fig. 3. Applying Nakoula's algorithm, the original input partition is reconstructed in just four iterations (4 rules are added in zeroth iteration, 5 in first iteration, and 3 in each of remaining iterations, see Fig. 4). If the duplicate consequent singletons present at this stage

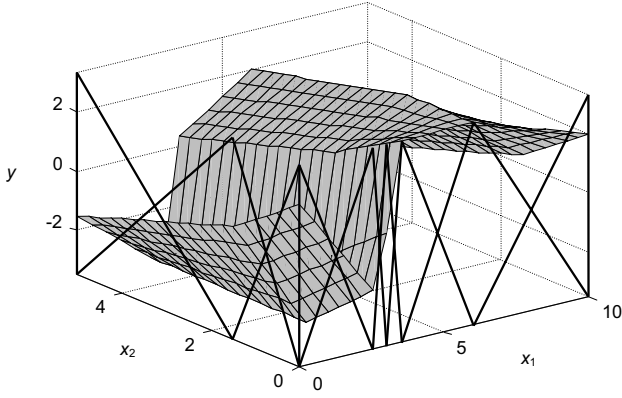


Fig. 3. Two-input fuzzy system for approximation

are removed, the number of consequent singletons reduces to 5, that are the singletons of the original system, which means that the application of Nakoula's method has been a total success.

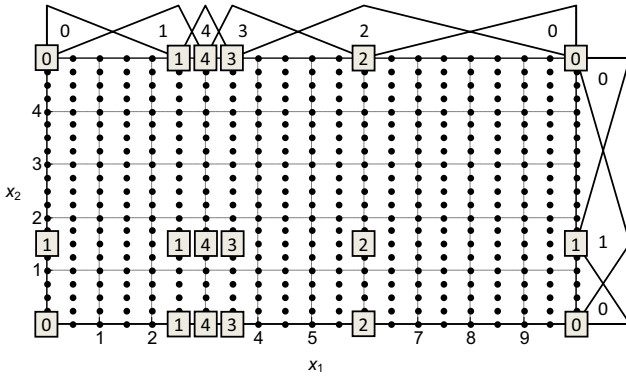


Fig. 4. The order in which the MFs of the model and rules are created in input space

However, the fundamental shortcoming of the method is that it does not cope well with noisy data, as an example in Fig. 5 bluntly demonstrates. The main issue with the algorithm is that it tends to learn noise rather than the signal (i.e. it favors the samples with highest noise ratio as rule nodes). There are two kinds of problems - first, outliers (erroneous samples 2, 6 and 8) are the worst offenders as those are concentrated on in the first place. This can be evidenced from the resulting input partition, i.e. the high concentration of input MFs at certain locations. Even if there are no obvious outliers in the data set, the samples with higher noise ratio are still among the first to be picked (e.g. samples no 1, 4, 12) and the resulting approximation is therefore grossly non-smooth. Besides, more iterations are required to obtain an approximation of any quality than in the noise-free case.

A proposed solution to this problem [5] replaces the consequent parameter identification routine in Nakoula's original

approach with the method of Nozaki et al. [7]

$$b_{jr} = \frac{\sum_{k=1}^K (\tau_r(k))^\alpha \cdot y_j(k)}{\sum_{k=1}^K (\tau_r(k))^\alpha}, \quad (5)$$

where α is a parameter that influences model accuracy in terms of root-mean-squared error (RMSE) - it is reported in [7] that $\alpha = 10$ provides best results in ideal environment and that it should be smaller if data is bad. Note also that if $\alpha = 1$, (5) is the local least squares method [10] and if it is very large, (5) performs the original approach of Nakoula, again. The basic important characteristic of Nozaki's method is that consequent parameters for a given rule are computed as the weighted average of relevant (relevancy is expressed by rule activation degree τ_r in (5)) output samples that gives the algorithm interpolating rather than extrapolating character.

Also, a subsequent input partition refinement procedure is suggested in [5]. While this effectively cures at least one part of the problem, the outliers in the data set still do their damage and leave a distinct footprint on the input partition, even after its refinement. Moreover, the solution may not converge at all.

Therefore, in current study we propose an alternative method for noisy data approximation. The basic idea is quite simple. First, we define a resolution vector $\text{res} = [\text{res}_1, \text{res}_2, \dots, \text{res}_N]$ that specifies the resolution for each input variable, measured as a percentage of its domain.

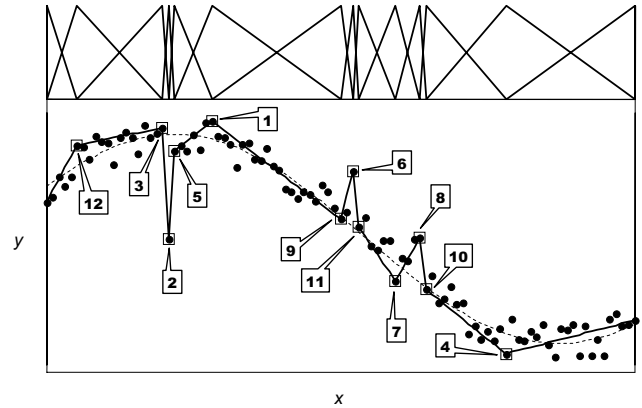


Fig. 5. Nakoula's algorithm fails when data is noisy. Numbers indicate the order in which the rule nodes (highlighted samples) are picked

Each iteration progresses normally as with Nakoula's approach with consequent parameters computed with (5). At the end of each iteration, however, the data samples that fall within the hypercube around the last rule node with dimensions $\text{res}_1 \cdot (x_1^{\max} - x_1^{\min}) \times \text{res}_2 \cdot (x_2^{\max} - x_2^{\min}) \times \dots \times \text{res}_N \cdot (x_N^{\max} - x_N^{\min})$ are removed from the training data set. The specified resolution measures apply to input axes too, i.e. if the distance between the i -th coordinate of the rule node $x_i(k)$ and the center a_i^s of an already existing MF is smaller than t_i , the center of already existing one is updated so that $a_i^s = (a_i^s + x_i(k))/2$ is the arithmetic mean

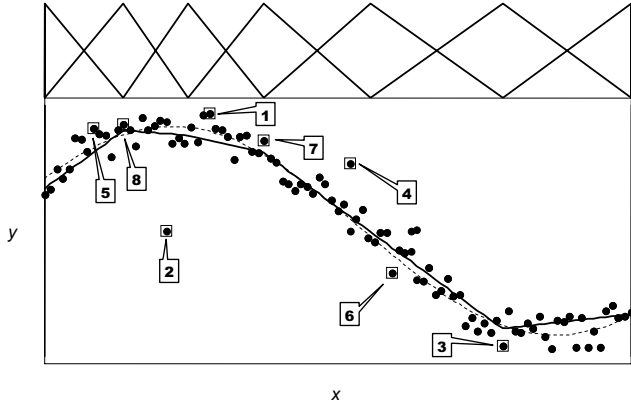


Fig. 6. Approximation of noisy data with the proposed algorithm ($\text{res} = 0.1$)

of those two (a new MF is not added to the partition of i -th input variable).

If we compare Figs. 5 and 7 with Figs. 6 and 8, respectively, we can see that the proposed method has several advantages: identified model has less MFs and consequently a lower number of rules as well as a lower RMSE. For the two-input system it also extracts the input partition that is very similar to the one that was used to generate the data set (compare Figs. 3 and 8).

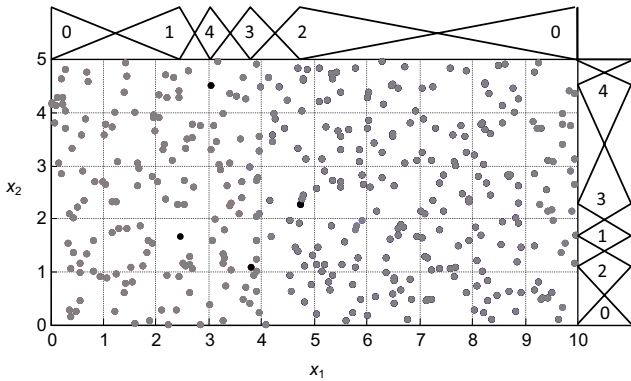


Fig. 7. Approximation of noisy multidimensional data with the Nakoula's method and MFs created at each iteration (numbered). Data points acting as rule nodes responsible for MFs are highlighted. $\text{RMSE} = 0.609$

IV. MODELING THE GAS FURNACE SYSTEM

The gas furnace data set [11] has been used extensively as a benchmark example for process identification. The data set consists of 296 input-output measurements sampled at a fixed interval of 9 seconds. The measured input $u(k)$ represents the flow rate of the methane gas in a gas furnace and the output measurement $y(k)$ represents the concentration of carbon dioxide in the gas mixture flowing out of the furnace under a steady air supply. Most studies (e.g. [12], [13], [14], [15], [16]) have used the inputs $y(k-1)$ and $u(k-4)$ which have the highest correlation with the output $y(k)$. Some studies (see Table I) use different and even more inputs.

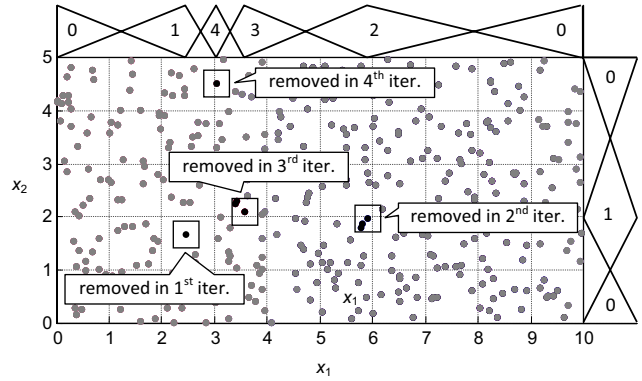


Fig. 8. Approximation of noisy multidimensional data with the proposed method: MFs and data sets removed at each iteration. $\text{res} = [0.05 \ 0.10]$, $\text{RMSE} = 0.429$.

We apply the proposed algorithm to obtain the models with 2, 4 and 5 inputs (for which there exists comparison material in literature). There is no apparent noise in the data set, so for output parameter identification we can use Nakoula's original procedure. Selection of input variables, along with obtained errors (measured by mean squared error (MSE) in this case) and the number of rules (R) are given in Table I. It must be noted, though, that several studies use 1-st order TS models and finding a common ground for comparison of the number of rules in different types of systems is not straightforward at all (each rule of a 1-st order TS system contains $N + 1$ independent parameters vs. a single one in a 0-th order TS rule, and is also dependent on input variables).

TABLE I
COMPARISON OF GAS FURNACE MODELS

study	inputs	MSE	R
Tong [12]	$y(k-1), u(k-4)$	0.469	19
Pedrycz [13]		0.320	81
Xu [14]		0.328	25
Sugeno [15]		0.359	2
Wang [16]		0.158	5
proposed		0.168	14
Sugeno [17]	$y(k-1), u(k-3)$	0.190	6
Kim [18]		0.129	2
proposed		0.185	14
Sugeno [17]	$y(k-1), u(k-3), u(k-4), u(k-5)$	0.190	6
proposed		0.231	42
Lin [19]	$y(k-1), y(k-3), u(k-3), u(k-6)$	0.261	6
proposed		0.248	43
Nie [20]	$y(k-1), y(k-2), u(k-3), u(k-4), u(k-5)$	0.169	45
proposed		0.234	87

In what concerns modeling parameters, for 2, 4 and 5-input models we used uniform resolution specs for all input variables - 0.2, 0.3 and 0.4, respectively (we try to keep them as small as possible to avoid information loss but smaller values would result in a non-convergent model, thus, the proposed scheme is good for convergence) and the models were obtained in 15, 10 and 5 steps respectively. Rule filtering threshold was 0.1.

From Table I we can see that the proposed method

compares favorably with the majority of linguistic modeling attempts (that can be distinguished by a larger number of rules ($R > 10$)) and surprisingly well with TS modeling experiments.

Fig. 9 depicts the responses of the worst and best identified models. However, this is not the end to it. For further refinement of those models we can apply a recently developed simplification algorithm [21] that reduces the number of unique output singletons by subtractive clustering [22] and then updates their values by the least squares estimate (which in this case does not compromise their reliability because singletons are associated with several rules and thus constrained). Doing so, we are able to improve accuracy and reduce complexity of all identified models (see Table II, where S denotes the number of singletons of the model after simplification that must be compared against R of respective models in Table I). Interestingly, though, the least complex models can be considered the best (in [23] it is brought out that the gas furnace data set is known to represent an approximately linear input-output behavior and this trend is also apparent in the relationships between the commonly used predictors $y(k-1)$ and $u(k-4)$ and the output variable $y(k)$.)

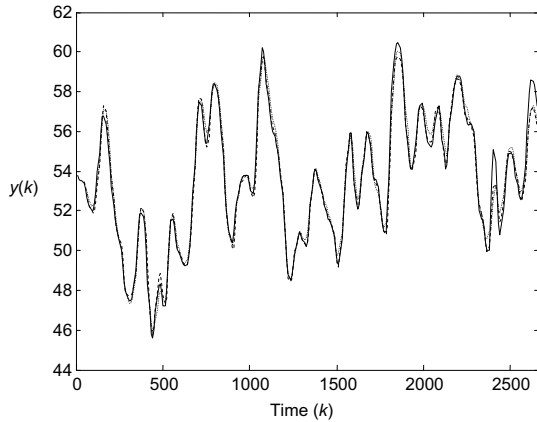


Fig. 9. Approximation of gas furnace data. Dashed line: response of the best model; dotted line: response of the worst model; normal line: original output

TABLE II
FURTHER IMPROVEMENT OF EXTRACTED MODELS BY THE ALGORITHM FROM [21]

N	initial MSE	final MSE	S
2	0.168	0.151	9
2	0.185	0.152	9
4	0.231	0.150	12
4	0.248	0.150	14
5	0.234	0.147	21

V. DESIGNING THE CONTROLLER TO FED-BATCH FERMENTATION

The fed-batch fermentation process that produces a secondary metabolite as the product is a minor benchmark

for modeling and control [24]. 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

$$\begin{cases} f_1 = 10 + \frac{25}{(1+e^{5-0.1t})} \\ f_2 = 3.5 + \frac{3.5}{(1+e^{10-0.15t})} \end{cases}, \quad (6)$$

is not good enough for the production (which is measured as $J = pV/T$, where T is the duration of fermentation). A notable problem with process is that it is impossible to obtain the same output series for each batch because initial state varies randomly within a subspace for each batch, the parameters of the model vary within specified limits and there are non-measurable disturbances.

The first step in controller design procedure is the identification of the model of the fermentation process with the proposed method, having the following structure

$$\text{IF } x \text{ is } A_{1r} \text{ AND } s_1 \text{ is } A_{2r} \text{ AND } s_2 \text{ is } A_{3r} \\ \text{THEN } \Delta x = b_{1r} \text{ AND } \Delta p = b_{2r}, \quad (7)$$

where the output variables are growth of biomass (Δx) and growth of the product (Δp). The training data is collected from 10 process runs with nominal feed profiles and is appropriately processed to match the model structure. The application of the method with $\text{res} = [0.1, 0.02, 0.02]$ (x is treated as the scheduling variable in present application, thus the larger value for res_1) and $\tau_{\min} = 0.1$ results in a 33-rule model with the input partition depicted in Fig. 10. Root-mean-squared-errors for Δx and Δp are 0.168 and 9.547, respectively (which are lower figures compared to those obtained in [25], which is quite remarkable considering the smaller number of rules in present case).

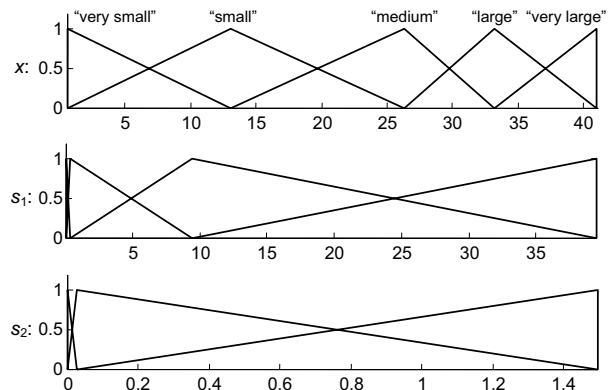


Fig. 10. Input MFs of the identified model

The extracted rules are grouped into five subsets (this is the number of MFs defined for biomass as can be seen from

Fig. 10) so that each subgroup contains the rules that have the same label of x of which one rule is selected and inverted

$$\text{IF } x \text{ is } A_{1r} \text{ THEN } s_1 = p_{1r} \text{ AND } s_2 = p_{2r}, \quad (8)$$

where $p_{1r} = \text{core}(A_{2r})$, $p_{2r} = \text{core}(A_{3r})$.

The selection and inversion task is performed automatically, thanks to the little tool developed in [26] and the resulting rules are as follows

IF x is "very small" THEN $s_1 = 9.432$ AND $s_2 = 0.027$
 IF x is "small" THEN $s_1 = 9.432$ AND $s_2 = 0.027$
 IF x is "medium" THEN $s_1 = 9.432$ AND $s_2 = 0.027$
 IF x is "large" THEN $s_1 = 0.395$ AND $s_2 = 0$
 IF x is "very large" THEN $s_1 = 9.432$ AND $s_2 = 0$

As we see, the inversion of the identified model is a five-rule fuzzy system that can be used as the control manager in the control system depicted in Fig. 11, where PI controllers take care of the substrate flow regulation. Achieved productivity with this manager averaged over 10 process runs is $1.20 \cdot 10^5$, which is on about the same level with the controllers in [25].

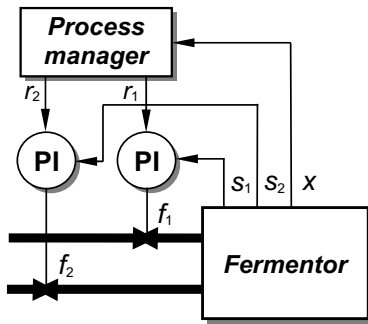


Fig. 11. The control system

As we have witnessed, the proposed method produces a valid model of the process even though the data is bad and is instrumental in making the controller design process a fully automated sequence.

VI. CONCLUSIONS

Identification of interpretable, reliable and numerically accurate linguistic fuzzy systems from bad data is a challenge because, generally, interpretability and accuracy are contradictory requirements. In current paper we proposed a modification to the combined algorithm that specifically targets this situation. It was demonstrated that removal of training data samples in the neighborhood of the rule node after it has been used for updating the model not only makes the method applicable in noisy environment but also improves convergence of the model. The algorithm can be coupled with another one [21] to further reduce the gap between the internal and external consistency of the model.

ACKNOWLEDGMENT

This work was partially supported by Estonian Science Foundation under grant no. 6837.

REFERENCES

- [1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Systems Man and Cybern.*, vol. 15, pp. 116-132, 1985.
- [2] J. Casillas, O. Cordon, F. Herrera and L. Magdalena (Eds.), *Interpretability Issues in Fuzzy Modeling*, Springer, Berlin, 2003.
- [3] A. Riid and E. Rüstern, "Interpretability of Fuzzy Systems and Its Application to Process Control," *Proc. IEEE Int. Conf. Fuzzy Systems*, London, 6 pp., 2007.
- [4] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, 1996.
- [5] A. Riid and E. Rüstern, "Heuro-Fuzzy Extraction of Interpretable Fuzzy Rules from Data," *Proc. IEEE Int. Conf. Syst., Man, Cybernetics*, The Hague, 2266-2271, 2004.
- [6] Y. Nakoula, S. Galichet, and L. Foulloy, "Simultaneous Learning of Rules and Linguistic Terms," *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, pp. 1743-1749, 1996.
- [7] K. Nozaki, H. Ishibuchi and H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets and Systems*, vol. 65, pp. 251-270, 1997.
- [8] A. Riid and E. Rüstern, "Transparent fuzzy systems and modeling with transparency protection," *Proc. IFAC Symp. on Artificial Intelligence in Real Time Control*, Budapest, pp. 229-234, 2000.
- [9] B. Kosko, "Optimal Fuzzy Rules Cover Extrema," *Int. J. of Intelligent Systems*, vol. 10, no. 2, pp. 249-255, 1995.
- [10] J. Abonyi, *Fuzzy Model Identification for Control*, Birkhauser, Boston, 2003
- [11] J. E. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, San Francisco, Holden Day, 1970.
- [12] R. M. Tong, "Synthesis of Fuzzy Models For Industrial Processes: Some Recent Results," *Int. J. General Syst.*, vol. 4, pp. 143-162, 1978.
- [13] W. Pedrycz, "An identification algorithm in fuzzy relational systems," *Fuzzy Sets and Systems*, vol. 13, pp. 153-167, 1984.
- [14] C. W. Xu and Y. Z. Lu, "Fuzzy model Identification and self-learning for dynamic systems," *IEEE Trans. Syst. Man and Cyber.*, vol. 17, pp. 683-689, 1987.
- [15] M. Sugeno and K. Tanaka, "Successive Identification of a Fuzzy Model and its Application to Prediction of a Complex System," *Fuzzy Sets and Systems*, vol. 42, pp. 315-334, 1991.
- [16] L. Wang and R. Langari, "Complex Systems Modeling via Fuzzy Logic," *IEEE Trans. on Syst., Man, and Cybernetics*, vol. 26, no. 1, pp. 100-106, 1996.
- [17] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7-31, 1993.
- [18] M. S. Kim, C. H. Kim and J. J. Lee, "Evolving compact and interpretable Takagi-Sugeno fuzzy models with a new encoding scheme," *IEEE Trans. Syst. Man Cybernet B: Cybernet.*, vol. 36, no. 5, pp. 1006-1023, 2006.
- [19] Y. Lin and G. A. Cunningham, "A new approach to fuzzy-neural system modeling," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 190-197, 1995.
- [20] J. Nie, "Constructing fuzzy model by self-organizing counterpropagation network," *IEEE Trans. Syst. Man Cybernet.* vol. 25, no. 6, pp. 963-970, 1995.
- [21] A. Riid and E. Rüstern, "Interpretability Improvement of Fuzzy Systems: Reducing the Number of Unique Singletons in Zeroth order Takagi-Sugeno Systems," submitted for IEEE Int. Conf Fuzzy Systems, 2010.
- [22] S. L. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, vol.2, pp. 267-278, 1994.
- [23] W. Faraq and A. Tawfik, "On Fuzzy Model Identification and the Gas Furnace Data," *Proc. IASTED Int. Conf. Intelligent Systems and Control*, Honolulu, pp. 210-214, 2000.
- [24] R. Leigh, "Fed-batch fermentation process modelling and control competition," *Trans. Inst. MC*, vol. 20, no. 1, pp. 3-4, 1998.
- [25] A. Riid and E. Rüstern, "Extracting and Exploiting Linguistic Information from a Fuzzy Process Model," in J. A. Tenreiro Machado, B. Patkai and I. J. Ruda (Eds.), *Intelligent Engineering Systems and Computational Cybernetics*, Springer-Verlag, pp. 95-105, 2009.
- [26] A. Riid and E. Rüstern, "Automatic Linguistic Inversion of a Fuzzy Model for Fed-Batch Fermentation Control," *Proc. Int. Conf. Intelligent Engineering Systems*, London, pp. 129-134, 2006.