EVOLUTIONARY LEARNING IN IDENTIFICATION OF FUZZY MODELS: APPLICATION TO DAMADICS BENCHMARK

Arunas LIPNICKAS*, Józef KORBICZ

Institute of Control and Computation Engineering, University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland, e-mail: {A.Lipnickas;J.Korbicz@issi.uz.zgora.pl}
***Permanent employment: Department of Control Technology, Kaunas University of Technology, Studentu 48, 3031 Kaunas, Lithuania, e-mail: lipnick@soften.ktu.lt**

Abstract: Evolutionary learning and especially genetic optimisation algorithms have recently received a lot of research attention as tools for identifying fuzzy models of the systems. Most often fuzzy modelling employ the fuzzy IF–THEN rules. In this paper, besides AND–operator the OR–operator is also considered in constructing the premise rule base. A genetic algorithm is utilised to find the premise structure of the rules, also to optimise fuzzy set membership functions and the consequent model structure of the rules at the same time. The performance of the approach is demonstrated on the DAMADICS benchmark problem.

Keywords: Evolutionary learning, fuzzy model, fault diagnosis

1. INTRODUCTION

Fuzzy modelling utilising fuzzy IF–THEN rules, provides a tool for designing qualitative models without employing precise quantitative analysis. However, there are many situations where expert domain knowledge, which is usually the basis for designing fuzzy models, is not sufficient, due to incompleteness of the existing knowledge, problems caused by different biases of human experts, difficulties in forming rules, etc. That is why, methods for data–driven identification of fuzzy models are of great interest. Most approaches proposed in literature emphasise the function approximation capabilities of fuzzy systems, and little attention is paid to minimisation of the rule base [3]. If the number of rules is too large, the fuzzy model is hardly interpreted by the man. Therefore in this work we require the model to be accurate and also to be as simple as possible.

We consider the problem of data–driven fuzzy rule–based modelling by the Takagi-Sugeno (TS) type [1, 4, 7].

Rule premises play a critical role in the TS fuzzy system since they determine the structure of a rule base. The fuzzy rules of the fuzzy model are extracted from training examples by means of genetic–based premise learning. In order to construct a simple fuzzy model with a high generalisation capability, a general premise structure allowing incomplete compositions of input variables as well as OR operation of input terms is considered.

In this paper a genetic algorithm (GA) is utilised to optimise the premise structure of the rules, fuzzy set membership functions, and the consequent part at the same time. Determination of rule conclusions is nested in the premise learning, where consequences of individual rules are determined under fixed preconditions. During the running of GA the actual rule number is adjusted automatically within a specified limit. The modelling procedure utilize GA to search in the combinatorial space for the optimal structure of premises also to optimise parameters of fuzzy set membership functions as well as to find the optimal parameters of consequent part simultaneously. GA searches a wide space of possible solution, so there is a high probability that the found solution is global or near global [5].

Recently, Xiong [8] introduced a general premise structure allowing not only incomplete composition of input variables but also OR connectives of input terms, so that a high generalisation ability can be achieved by an individual rule. The upper limit of the rule number is predetermined by the technician in advance. It can be considered as an estimate of the sufficient amount of rules to achieve a satisfactory accuracy.

When the rule–base premise is constructed, then the polynomial models in consequent part is found by the local weighted least square method.

The performance of the methods is demonstrated on the DAMADICS benchmark problem. The multi–disciplinary and complementary EU Research Training Network project DAMADICS is focused on development and application of methods for actuator diagnosis in industrial control systems [2].

The paper is organised as follows. In the second section, the background of fuzzy Takagi–Sugeno model is given. The evolutionary learning is presented in the third section. In the fourth section, the DAMADICS benchmark is described. The experimental results are presented in section five. Finally, section six presents conclusions.

2. TAKAGI–SUGENO MODEL

The R_i TS rule takes the following form:

$$R_i : \textbf{IF} \quad \mathbf{u} \text{ is } A_i \textbf{ THEN } y_i = f_i(\mathbf{u}), \quad i = 1,\ 2, \ldots, K, \tag{1}$$

where K is the number of rules.

Usually fuzzy rules consist of canonical AND connection of fuzzy terms in a rule premise. In complex processes with multiple input variables such a system is not suitable, since the number of rules increases exponentially by increasing input dimension. Therefore rules with incomplete structure containing OR connections of fuzzy terms is preferable. In this way, the number of rules in the premise is decreased by finding similar consequence parts for different fuzzy terms.

Generally, $f_i(\cdot)$ is a polynomial function from the input variables $\mathbf{u}$, but in principle it can be any kind of function as long as it can appropriately describe the output of system within the fuzzy regions specified by the premise of the rule. A simple and practically useful parameterisation is the affine (linear in parameters) form, yielding the rules:

$$R_i : \textbf{IF} \quad \mathbf{u} \text{ is } A_i \textbf{ THEN } y_i = \mathbf{a}_i^T \mathbf{U} + a_{0,i}, \quad i = 1,\ 2, \ldots, K, \tag{2}$$

where $\mathbf{U}$ is the polynomial input vector constructed from $\mathbf{u}$, the $\mathbf{a}_i$ is a parameter vector and $a_{0,i}$ is a scalar offset. When we have the Z-dimensional polynomial of degree 2, then the parameter vector $\mathbf{a}_i$ consist of L elements:

$$L = \frac{(2+Z)!}{2!Z!} - 1. \tag{3}$$

Parameter optimisation can be performed very fast by a least squares algorithm.

However, the number of parameters grows rapidly with increasing input dimensionality. One way to decrease the number of parameters is to perform *structure optimisations*. The structure optimisation can be performed by a linear subset selection technique such as the orthogonal least squares algorithm [6]. However, this method suffers from the curse of dimensionality.

2.1. Proposed structure optimisation

For this problem, the authors proposed to use GA for selecting the parameters of 2 order polynomial. Willing to determine the efficient model structure for each rule. The chromosome of the consequent part in GA is coded by {0,1} meaning the usefulness of each term in the the parameter vector $\mathbf{U}_i$. Then the output function takes the following form:

$$y_i = f(\mathbf{u}, \mathbf{I}_i), \tag{4}$$

where $\mathbf{I}_i$ is the binary vector expressing the importance of parameters. The Z-dimensional polynomial of degree 2 is computed:

$$y_i = a_0 I_0 + \sum_{j=1}^{Z} a_j u_j I_j + \sum_{j_1=1}^{Z} \sum_{j_2=1}^{Z} a_{j_1,j_2} u_{j_1} u_{j_2} I_{j_1,j_2} \tag{5}$$

where $\mathbf{I}_i = [I_0, I_1, ..., I_Z, I_{1,1}, ..., I_{Z,Z}]$, I_j and $I_{j_1,j_2} \in \{0, 1\}$. When $I_j = 0$, then this term of polynomial function is absent in the model.

3. EVOLUTIONARY LEARNING OF RULE PREMISES

GA is the global search algorithm that emulates the mechanism of natural genetics and selection [5]. Based on probabilistic decisions it exploits historic information to guide the search for new points in the problem space with expected improvement in performances. In the genetic search a constant population size is always maintained. An individual in the population encodes a possible solution to the problem into a string, which is analogous to a chromosome in nature. At each iteration step, new strings are created by applying genetic operators on selected parents for recombination. Coding scheme, genetic operators (reproduction, crossover and mutation) and fitness function are the key points for the GA to optimise the structure of rule premises and input membership functions at the same time.

The GA algorithm consist of three basic operations: reproduction, crossover, and mutation. Reproduction is the process where members of the population are reproduced according to the relative fitness of the individuals, where the chromosomes with higher fitness have higher probabilities of having more copies in the new coming generation. There are a few selection schemes available for reproduction, i.e. the *roulette wheel*, the *tournament scheme* and etc. [5]. Crossover in the GA occurs when the selected chromosomes partially exchanges the information in the genes. Mutation is a random alternation of a bit in a string so as to increase the variability of population.

Dynamic crossover and mutation probability rates, are used in the GA operation, as they provide faster convergence when compared to constant probability rates. There are many approaches which can be applied to settle the problem. In this work the crossover and mutation probability rates were calculated by the following formula:

$$\begin{aligned} P_{\text{cross},m+1} &= \exp(-m/M), \\ P_{\text{mut},m+1} &= \exp(0.02\ \ m/M) - 1, \end{aligned} \qquad (6)$$

where m and M are the current generation and the number of maximum generations respectively.

3.1. Premise rules and membership functions coding

Triangular formed fuzzy sets are used in this study. To achieve an optimal interface there is a requirement that the sum of membership values for every input variable should be always equal to one, so then only certain end points (also peaks) of membership functions need to be optimised by GA [8]. For instance, the five triangular membership function are defined with only three parameters. In this work we used 8 bits *Grey–Code* coding for each parameter of membership functions.

In logical part of rule premise each bit of the segment corresponds to a linguistic term with bit '1' presence and '0' for absence of the fuzzy sets forming the condition. For example, the input u has three linguistic terms (membership functions) i.e. $\{$*low*, *middle*, *high*$\}$, the segment '100' and '101' correspond to the conditions u_k='*low*' and the second one to u_i='*low*' OR '*high*'. Situation '111' represents a stay 'do not care about the input'. The all zeros in the segment correspond to harmful and meaningless premise, therefore this group is not considered for calculation.

However, there is also a negative effect of using OR operator in fuzzy rules. The consistency of the rule base is not guaranteed compared to the usual rules. The overlapping between two rules leads to a conflict between them, i.e. when the rules suggest different outputs in their consequence parts. To give the numerical assessment of knowledge coherence, several definitions were introduced in [9]. The amount of conflicts (AC) in the model is commuted as the sum of all intersections between term pairs in rules:

$$AC = \sum_{i=1}^{P} \sum_{j=i+1}^{P} \|R_i \cap R_j\|, \qquad (7)$$

where P is the number of fuzzy partitions in the premise.

The parameters of each consequent part in the rules are obtained as a least-squares estimate. Let for eq. 2 denote the $\Theta_i^T = [\mathbf{a}_i^T; a_{i,0}]$, the input matrix $[\mathbf{U}; \mathbf{1}]$ as $\mathbf{U}_{in}$, and let $\mathbf{\Phi}_i$ denote a diagonal matrix in $\mathbf{R}^{N_i \times N_i}$ having the degree of membership functions activation for the given training data. The weighted least squares solution of $y_i = \mathbf{U}_{in}\Theta_i + \varepsilon$ becomes:

$$\Theta_i = [\mathbf{U}_{in}^T \mathbf{\Phi}_i \mathbf{U}_{in}]^{-1} \mathbf{U}_{in}^T \mathbf{\Phi}_i y_i. \qquad (8)$$

When the available data is sparse and noisy or the dimension of input matrix is high, then data may be not informative enough to estimate the model that is more complex than linear. One solution is the use of quadratic programming technique with inequality constrains or to apply model *structure optimisation*.

3.2. Fitness function

The task of defining a fitness function is usually application specific, such that it is formulated to achieve the optimal model of the system. In this work we require the model to be accurate and also to have few premise rules as possible. According to those criteria, the fitness function F_l for l^{th} model is constructed as follows:

$$\text{MSE}_l = \frac{1}{N} \sum_{n=1}^{N} (y_{l,n} - y_n)^2, \qquad (9)$$

$$F_l = \frac{1}{1 + \text{MSE}_l(1 + \beta AC_l)}, \qquad (10)$$

where MSE_l is the mean squared error, y_l denotes the model's estimate of the real output y, the AC_l indicates the amount of conflicts in l^{th} model, N is the number of data samples, and the coefficient β express the importance of conflicts in the model. When $\beta = 0$ it means, that we do not care about conflicts in the rules.

The object of GA optimisation is to maximise the fitness (eq. 10).

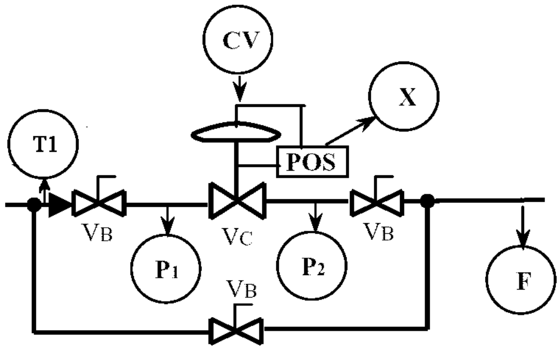

Fig. 1. The scheme of the actuator block.

4. CASE STUDY: DAMADICS BENCHMARK

Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems (DAMADICS) is a Research Training Network funded by the European Commission under Framework V.

This multidisciplinary and complementary RTN DAMADICS is focusing on drawing together wide-ranging techniques and fault diagnosis within the framework of a real application to online diagnosis of a 5-stage evaporization plant of the sugar factory at Lublin, Poland.

The actuator block can be considered as a four-input and two-output system as shown in Fig. 1. The input u_k and output y_k are defined as follows:

$$u_k = (CV, P1, P2, T1),$$
$$y_k = (F, X), \qquad (11)$$

where process variables are: *CV* - control valve, $P1$ - pressure at the inlet of the valve, $P2$ - pressure at the outlet of the valve, $T1$ - juice temperature at the inlet of the valve, F - juice flow at the outlet of the valve, and X - servomotor rod displacement.

Within the DAMADICS project the actuator simulator was developed under MATLAB Simulink. This tool makes it possible to generate normal operating mode data as well as data for 19 faults. The comprehensive description of DAMADICS benchmark is available via web site [2].

It is well known that every MIMO system can be decomposed into several MISO systems and analysed separately. In our experiments the outputs F and X were analysed as separate systems.

5. EXPERIMENTAL RESULTS

The parameters for the GA used in the experiments are as follows: population size 50; maximum number of iterations 50; the rates of mutation and crossover are changed according to (eq. 6). Training and testing data of the actuator block was generated through MATLAB Simulink: 750 and 1000 of data points for training and testing respectively. In all the tables given we calculate *MSE* according to (eq. 9).

At the very beginning we had no idea about the starting number of rules and fuzzy partitions. Therefore, only the upper limit of rules was determined as 20. The number of fuzzy partitioning, plays a role for the fuzzy system interpretability. According to Nelles [6], if number of fuzzy partitioning is more than four, then such a system is hard to be interpreted. Therefore, in our experiments we used only four fuzzy partitioning per input in the rules premise. Meaning, that for the TS model these numbers will be sufficient to achieve desirable model accuracy.

After several experiments, the coefficient β in the fitness function was found to be 0.005 and kept unchanged. We had to play also with input parameters selection for the premise as well as for the consequent part of the fuzzy model. The TS model with selected parameters are:

a) for the juice flow

$$R_i : \textbf{IF } \{u_{1,k}, y_{1,k-1}\} \text{ is } A_i \textbf{ THEN } y_{1,k} = f_i(\mathbf{u}_k, \mathbf{u}_{k-1}, \mathbf{I}_i) \qquad (12)$$

b) for the rod displacement

$$R_i : \textbf{IF } \{u_{1,k}, u_{1,k-1}\} \text{ is } A_i \textbf{ THEN } y_{2,k} = f_i(\mathbf{u}_k, \mathbf{u}_{k-1}, \mathbf{I}_i) \qquad (13)$$

where $y_{1,k-1}$ is the previous output of the model, $\mathbf{I}_i$ is the binary parameters importance vector used only in the proposed approach and A_i is the composition of fuzzy sets.

For us the structure of the models looks reasonable, since the rod displacement (X) do not depend on the previous output, but on the previous control value (CV) as it is seen from Fig.1. However, the model for flow depends on previous flow value ($y_{1,k-1}$) and actual control value (CV).

In Fig. 2 there is given an example of input space partitioning obtained by GA algorithm for *flow* modelling with 4 rule input premise. Each input is partitioned in 4 fuzzy sets, and fuzzy region are logically assigned to one of the rules. For this example, there is a possibility to obtain 16 rules without intersection. The process data is distributed only in the restricted fuzzy areas, therefore the output models for some regions is not constructed. For instance, bottom–left part of the Fig. 2 is absent in the model. The erroneous situation arises, when unexpected input data from those regions is given to the TS model. The solution in this situation is by adding special conditional rule *"if no rule selected"*, then produce some output. As the output it might by a constant signal or some prespecified function output.

The other conflicting situations in the Fig. 2 is the earlier mentioned conflicting situations in the rule–base. In this example, there are two conflicting points $\{R_1, R_3\}$ and $\{R_3, R_4\}$. By running some subroutine there is a possibility to eliminate those points. In our work we have been using the rule order procedure i.e. the first fitted rule produce the output for the given input.

In the experimentation the consequent part of the model is constructed using linear, polynomial and polynomial function with proposed selection. The experimental results is shown in Tables 1, 2 and Figs. 3, 4. The tables presents the calculated MSE error on the test set, maximum absolute difference between modelled and original output, number of rules in the premise and the total number of parameters in TS model.

From the results we can see, that the relationship between the input u_k and the juice flow $y_{1,k}$ cannot be accurately modelled by linear functions. The modelling error is approximately 15 %. The model with polynomial function reach the 10 % of accuracy. On the other hand, the relation between the input u_k and the rod displacement $y_{2,k}$ can be modelled by the TS with linear

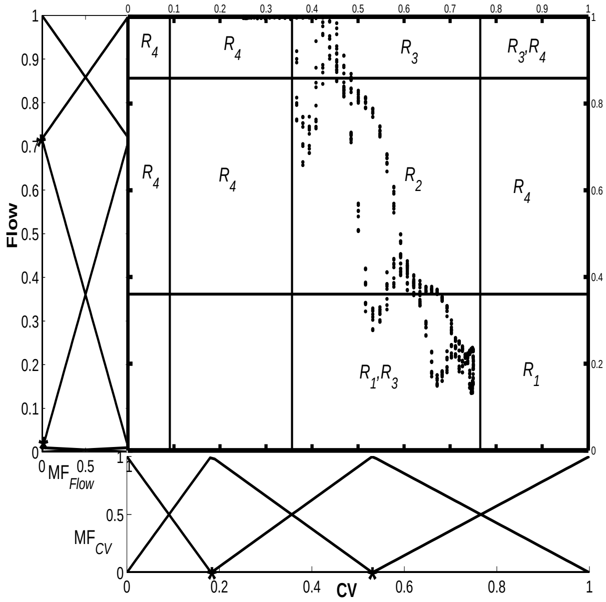

Fig. 2. Input premise partitioning and fuzzy sets for $y_{2,k}$ $(Flow)$ modelling.

Table 1. Model performance for *Flow*

	MSE *e-5	Max. Abs. error	#of rules	#of parameters
Linear	100	0.15	4	48
Polynomial	26	0.13	5	250
Proposed	25	0.10	4	117

Table 2. Model performance for *Rod displacement*

	MSE *e-5	Max. Abs. error	#of rules	#of parameters
Linear	21	0.05	3	41
Polynomial	6	0.032	3	151
Proposed	8	0.031	3	93

consequent functions. From Figs. 3 and 4 we can see the residuals of the designed models and original system.

The polynomial model and proposed approach reach approximately the same results, but the advantage of the proposed approach is that it has half as many parameters in the final model.

6. CONCLUSIONS

In this paper we showed the synthesis of evolutionary learning and the fuzzy model for system identification. The size of the rule base were

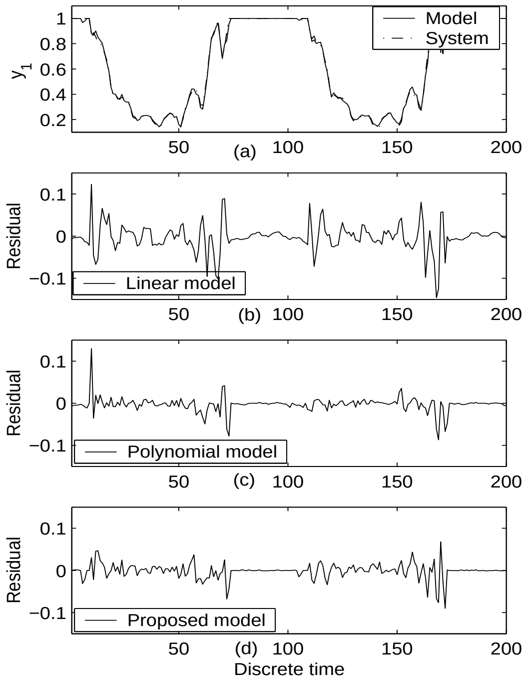

Fig. 3. The outputs and residuals for the flow modelling: a) system (dotted) and model (solid) outputs; b) residual of linear model; c) residual of polynomial model; d) residual of proposed approach.

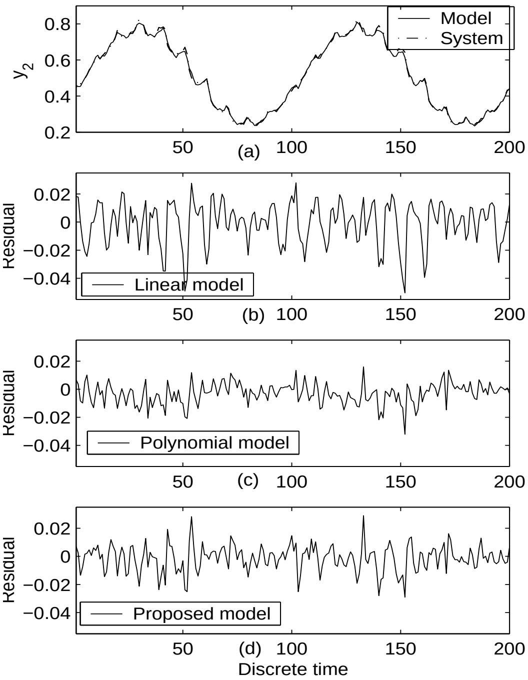

Fig. 4. The outputs and residuals for the rod displacement modelling: a) system (dotted) and model (solid) outputs; b) residual of linear model; c) residual of polynomial model; d) residual of proposed approach.

generated and consequent part model parameters were determined by the GA. The rule premise consisted with AND and OR connectives, this led to the reduction of the number of rules by finding similar consequent parts of the model. The proposed approach, comparing with second order polynomial models, for consequent part optimisation produced not only accurate models, but the models with half the number of parameters.

7. ACKNOWLEDGEMENTS

This work was supported by EU FP 5 Research Training Network project DAMADICS: Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems.

REFERENCES

[1] Babuska R., *Fuzzy Modeling for Control*, Boston, USA:Kluwer Academic Publishers, 1998.

[2] EC FP5 Research Training Network DAMADICS:Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems, (http://www.eng.hull.ac.uk/research/control /damadics1.htm).

[3] Homaifar A. and McCormick E., Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, *IEEE Transaction on Fuzzy Systems*, Vol. 3, 1995, pp. 129-139.

[4] Mamdani E.H., Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems* 26,1977, pp. 1182-1191.

[5] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin:Springer, 1996.

[6] Nelles O., *Non–linear Systems Identification. From Classical Approaches to Neural Networks and Fuzzy Models*, Berlin:Springer, 2001.

[7] Takagi T. and Sugeno M., Fuzzy identification of systems and its application to modeling and control. *IEEE Transaction on Systems, Man and Cybernetics 15* (1), 1985, pp.116-132.

[8] Xiong N., Evoliutionary learning of rule premises for fuzzy modeling, *International Journal of Systems Science*, 2001, Vol.32, No.9, pp. 1109-1118.

[9] Xiong N., Litz Z., Reduction of fuzzy control rules by means of premise learning – method and case study, *Fuzzy Sets and Systems* 2002, Vol.132, pp.217-231.