Proceedings of V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 499-504, April 2-5, 2001 - Rio de Janeiro, RJ - Brazil

Neuro-Fuzzy Modeling with Structure and Parameter Optimization

Alexandre Evsukoff[1,2], Antonio C. P. D. Alves[1] and Roberto Schirru[1]
[1]*PEN/COPPE-UFRJ, Laboratório de Monitoração de Processos, Cx. Postal 68509, 21945-970 Rio de Janeiro RJ, BRASIL*
[2]*ILTC, Instituto de Lógica, Filosofia e Teoria da Ciência, Rua Almirante Teffé 637, 24030-080, Niterói-RJ, BRASIL*
E-mails: evsukoff@cepel.br, adpalves@bol.com.br, schirru@lmp.ufrj.br

Abstract

This work presents a method for non-linear fuzzy model identification. The main characteristic of the method is the automatic determination of the number of fuzzy sets in the domain of each input variable and the further optimization of their location by a gradient descent algorithm. The output rule parameters are also optimized by a Least Square Error minimization. The fuzzy sets are determined in such a way that the resultant fuzzy rule base is readable by domain experts. The methodology is applied to numerical examples found in the literature.

1 Introduction

Several methods for identification of fuzzy models have been reported in the literature ([1] [2] [3] [4], among others). Whatever the fuzzy model type, the model parameterization requires to determine the fuzzy sets that will be used to describe the considered variables. In the fuzzy literature, the structure identification problem is often restricted to this specific task. It means that the variable selection is previously done using conventional techniques.

This paper presents a fuzzy model identification method where the fuzzy model structure is defined automatically and further optimized by a gradient descent algorithm. The model output parameters are computed by *Least Square Error* (LSE) optimization.

The global strategy is based on an incremental building of the rule base that results in a good compromise between numerical precision and readability. Furthermore, no assumption is made concerning the model granularity, i.e. the number of input fuzzy sets.

The remainder of this paper is organized as follows: the next section introduces fuzzy systems and presents a vector-matrix formulation of fuzzy reasoning. In section three, the structure identification method is presented for SISO systems. The optimization of fuzzy sets location and the optimization of rules output parameters are also described. Finally, the method efficiency is illustrated through four numerical examples found in the literature.

2 Fuzzy Models

2.1 Preliminaries and notation

For many applications, the mathematical representation of natural language concepts through fuzzy sets is done by the definition of a *base variable* x, whose domain $X \subset R$ is the numerical support where concepts can be expressed

Fuzzy sets allow a meaningful representation of concepts (often vague, uncertain or imprecise) expressed in natural language. A fuzzy set is defined as:

$$\tilde{A} = \{(x, \mu_A(x)), x \in X\} \quad (1)$$

where X is the *universe* (often the numerical support) and $\mu_A : X \to [0,1]$ is the function that denotes the *membership* of an element $x \in X$ to the fuzzy set $\tilde{A}$.

The values $x \in X$ are described using ordered linguistic terms that belong to a *descriptor set* $\mathbf{A} = \{A_1, \dots, A_m\}$. The *meaning* of each term $A_i \in \mathbf{A}$ is given by the fuzzy set $\tilde{A}_i$.

The collection of fuzzy sets used to describe the base variable forms a *fuzzy partition* $\tilde{\mathbf{A}} = \{\tilde{A}_1, \dots, \tilde{A}_m\}$ of the base variable domain. It allows a fuzzy discretization [5] of the base variable domain. In this work, it is assumed that *strong* fuzzy partitions are used:

$$\textstyle\sum_i \mu_{A_i}(x) = 1, \forall x \in X \ . \quad (2)$$

Furthermore, as the aim of this work is the identification of fuzzy models that can be interpreted by domain experts, it is considered that the membership functions are triangular-shaped[1] and normalized in such a way that:

$$\forall i, \exists x_0 \in X, \ / \ \mu_{A_i}(x_0) = 1 \ . \quad (3)$$

Strong normalized triangular fuzzy partitions are completely determined by the location of the triangle vertices. A fuzzy partition $\tilde{\mathbf{A}}$ can thus be parameterized by the set $p_{\mathbf{A}} = \{a_1, \dots, a_m\}$, whose elements can be viewed as *prototypes* (best representatives) of the fuzzy sets forming partition $\tilde{\mathbf{A}}$ (see Figure. 1). Trapezoidal

[1] Notice that triangular-shaped membership functions are inherently convex.

membership functions are used for the two fuzzy sets at each end of the domain, as shown in Figure. 1, to deal with off-limit points.

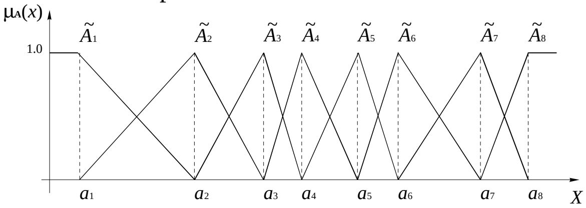

Figure. 1: Triangular shaped membership functions.

The structure identification of a fuzzy model aims to determine the number and location of prototypes that define the fuzzy partition over the domain of each variable. Actually, it consists in determining the partition granularity. The greater the number of fuzzy sets in a fuzzy partition, the smaller the granularity is. The representation is then more precise but less convenient for linguistic interpretation. The granularity is thus related to the generality or precision of the fuzzy description.

2.2 TSK fuzzy systems

A very popular fuzzy model is the one introduced by Takagi, Sugeno and Kang [6][7], often called the TSK fuzzy model in the fuzzy system literature [4]. In the simplified or zero order TSK fuzzy model, the rule conclusions are stored in a column vector $\boldsymbol{\theta}$ and the rules for SISO (Single Input Single Output) systems are written as:

$$if \;\; x \;\; is \;\; A_i \;\; then \;\; \hat{y} = \theta_i \,. \tag{4}$$

For an input $x(t) \in X$, the output of a zero order TSK fuzzy model is computed by the vector expression:

$$\hat{y}(t) = \mathbf{w}(t).\boldsymbol{\theta} \,. \tag{5}$$

where $\mathbf{w}(t) = \left(\mu_{A_1}(x(t)), \ldots, \mu_{A_m}(x(t))\right)$ are the vector whose components are the *fire strengths* of rule premises [4] and $\boldsymbol{\theta} = \left(\theta_1, \ldots, \theta_m\right)^t$ is the parameter vector.

The extension of the method to MISO (Multiple Inputs Single Output) models is achieved by considering the input variables as components of a vector. The rules are written as:

$$If \;\; x_1 \; is \; A_{i1} \; and \ldots and \; x_r \; is \; A_{ir} \;\; then \;\; \hat{y} = \theta_i \,. \tag{6}$$

where $\boldsymbol{\theta} = \left(\theta_1, \ldots, \theta_M\right)$ is the parameter vector whose components are the outputs of TSK rules

All the combinations of terms in the premises must be considered in such a way that the model is complete, *i.e.* it produces an output for whatever input values. The MISO fuzzy model described by rules like (6) is analogous to the SISO fuzzy model described by rules like (4). Its output is computed as in (5):

$$\hat{y}(t) = \mathbf{w}(t).\boldsymbol{\theta} \,. \tag{7}$$

were $\boldsymbol{\theta} = \left(\theta_1, \ldots, \theta_M\right)$ and each component of the fire strength vector is computed as:

$$w_i(t) = \frac{\prod_{r=1..n} \mu_{A_{ir}}(x_r(t))}{\sum_{i=1..M} \prod_{r=1..n} \mu_{A_{ir}}(x_r(t))} \tag{8}$$

The TSK fuzzy model is often called neuro-fuzzy model in the literature, since the zero order TSK model is equivalent to a radial basis neural network [4]. The ANFIS neural network is an implementation of the first order TSK model in a neural network like topology. In ANFIS, the number of rules are fixed by the expert, the location of fuzzy sets are optimized by a gradient descent algorithm and the output parameters are computed by the solution of the LSE problem.

In next section, a structure identification method is presented were where the fuzzy model structure is defined automatically and the further optimized by a gradient descent algorithm and the model output parameters are computed by LSE optimization

3 Fuzzy model identification

Generally, fuzzy model identification methods follow the three generic steps stated in the introduction: structure identification, parameter estimation and model validation.

The structure identification of a fuzzy model consists of the determination of the model type and the number and location of fuzzy sets in the domain of each variable. The model parameters are generally associated with the rule base. The model validation must check the model precision, but also must certify that the model is readable by domain experts.

For rule base parameter estimation, recent methods fall into a Least Square Error (LSE) minimization problem [8][2][4]. From this point of view, identification of fuzzy systems is close to classic non-linear model identification.

The problem of structure identification of a fuzzy system is much more difficult. When available, the expert knowledge may be used to fix the *number* of fuzzy sets associated to each variable. In such a case, the optimization of the parameters that determine the *location* of fuzzy sets is computed by non-linear optimization [8]. Some methods use the *back propagation* algorithm in a neural network-like formulation of the fuzzy system [8] [4]. Such methods have some drawbacks related to non-linear optimization, such as the dependency on initial conditions. Besides the resulting fuzzy sets may not be meaningful for domain experts.

When no *a priori* knowledge is available, most methods use training data clustering for the determination of the model structure [1][9][3]. Several clustering techniques have been reported in the literature (such as Fuzzy c-Means and Fuzzy ISODATA), but results are very dependent on the choice of some parameters and, again, on the *number* of clusters (fuzzy sets) to represent each variable. Yager and Filev [4] have developed a clustering method where the number and location of clusters are computed by the value of a distribution function defined over pre-defined cluster center candidates.

An other methodology for fuzzy model structure identification, which is not based on data clustering has recently been developed [10] [11]. The method begins with a model with only two fuzzy sets to represent each variable. In the refining phase, new fuzzy sets are created at each iteration over the variable domain, and rule base parameters are computed. The iterative process continues until a specified tolerance is achieved.

The next section describes the basic ideas of the structure determination for SISO model identification. The fuzzy sets location optimization and the parameter optimization are then described.

3.1 Structure identification

For the sake of simplicity, the structure identification method is presented for the SISO case. Consider a training set T composed by N pairs $\langle \mathbf{x}, y \rangle(t)$, where $y(t) = f(x(t))$. The objective of the identification method is to build a fuzzy system to compute an estimation $\hat{y}(t) = \hat{f}(x(t))$, which minimizes the Mean Square Error (MSE) criterion, defined as:

$$J = \frac{1}{N} \sum_{t=1..N} \xi(t)^2 = \frac{1}{N} \sum_{t=..N} (\hat{y}(t) - y(t))^2 \qquad (9)$$

where N is the number of points in the training set and $\xi(t) = (\hat{y}(t) - y(t))$ is the *residual* at each point.

The structure identification aims to determine the prototype set $p_{\mathbf{A}} = \{a_1, \ldots, a_m\}$ whose elements define the fuzzy partition $\tilde{\mathbf{A}}$. The parameter identification aims to compute the parameter vector $\boldsymbol{\theta}$ whose components are the outputs of each rule.

The idea of the structure determination is to identify a set of *learned points* $P \subset T$ where for each element $(\alpha, \varphi)_i \in P$, $\varphi_i = f(\alpha_i)$ defines a TSK rule as (4), where $a_i = \alpha_i$ is the prototype of the triangular membership function of the fuzzy set $\tilde{A}_i$ and $\theta_i = \varphi_i$ is the corresponding rule output. Once $(\alpha, \varphi)_i \in T$, the fuzzy model output will have a null residual for all learned points. The convergence of the method is based on the fact that for $P \equiv T$, the fuzzy model will have a null residual for all training points.

The process starts with the definition of the set of input variable domain limits $\Omega = \{\omega_1, \omega_2\}$. In the SISO case $\omega_1 = x_{\min}$ and $\omega_2 = x_{\max}$. At initialization (iteration $\kappa = 0$), the domain limits are used as the prototypes of the two first fuzzy sets of the model:

$$p_{\mathbf{A}}(\kappa = 0) = \{a_1, a_2\} = \{x_{\min}, x_{\max}\} \qquad (10)$$

which defines the two first rules:

$$\begin{aligned} &If \;\; x \;\; is \;\; A_1 \;\; then \;\; \hat{y} = \theta_1 \\ &If \;\; x \;\; is \;\; A_2 \;\; then \;\; \hat{y} = \theta_2 \end{aligned} \qquad (11)$$

In the same way, the corresponding output values are used as the first two rules output:

$$\boldsymbol{\theta}(\kappa = 0) = (\varphi_1, \varphi_2) = (f(a_1), f(a_2)). \qquad (12)$$

The membership functions of the two limit fuzzy sets are trapezoidal to take into account off-limit points (Figure 2). All other membership functions generated by the algorithm will be triangular.

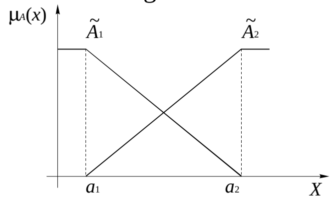

Figure 2: Initial fuzzy sets.

The limit points are included into the set of learned points:

$$P(\kappa = 1) = \{(\alpha, \varphi)_1; (\alpha, \varphi)_2\} \qquad (13)$$

where $\alpha_1 = a_1$, $\alpha_2 = a_2$, $\varphi_1 = f(a_1)$ and $\varphi_2 = f(a_2)$.

In the refining phase, the process becomes iterative; while the mean square error (9) does not reach a specified tolerance, the method will search for a new point to be included in the set of learned points. In a generic iteration κ with p learned points, the point $(\alpha, \varphi)_{p+1}$ to be included in the set of learned points is the farthest point in the training set with respect to the approximation error function [10] computed as:

$$(\alpha, \varphi)_{p+1} = \arg\max_{t=1..N} (\xi(t)^2). \qquad (14)$$

The point $(\alpha, \varphi)_{p+1}$ computed as (14) defines a new rule as (4). In the new iteration, the prototype set is updated, including the new learned point:

$$p_{\mathbf{A}}(\kappa) = p_{\mathbf{A}}(\kappa - 1) \cup \{\alpha_{p+1}\}. \qquad (15)$$

The prototype set must be reordered so that the prototype vector is in a strict ascending order. For instance, Figure 3 shows the introduction of the first learned fuzzy set into the fuzzy partition $\tilde{\mathbf{A}}$.

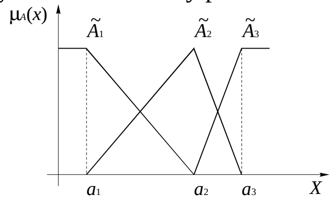

Figure 3: New fuzzy set.

The set of components of the output parameter vector is updated to include the rule output corresponding to the new learned point and it is reordered in such a way that each component corresponds to the rule output associated to each learned prototype.

The global identification error (9) is computed for the new model. The iterative process continues until the global error is smaller than the specified tolerance. Next section presents the optimization of the fuzzy sets location in the input variable domain.

3.2 Structure optimization

The structure identification method presented gives, at each iteration, a rough idea of the number of fuzzy sets required to build the fuzzy model. Further structure optimization can be achieved by moving the fuzzy sets location in order to reduce the MSE criterion (9).

The ANFIS system [4] is a neuro-fuzzy systems that implements TSK fuzzy models in neural networks. The ANFIS adaptation algorithm considers fixed number of rules and optimizes output parameters and fuzzy sets location in a hybrid algorithm that combines LSE minimization and back-propagation (gradient descent) non-linear optimization.

In this work this approach is used within the structure identification algorithm. The *prototype vector* $\boldsymbol{\alpha}(k) = (a_1, \dots, a_m)$, whose components are the elements of the prototype set, represents the fuzzy sets location parameters. In the gradient descent algorithm, the update rule is given by:

$$\boldsymbol{\alpha}(k+1) = \boldsymbol{\alpha}(k) - \eta \nabla J(k) \quad (16)$$

where k is a generic iteration of the gradient descent algorithm; η is the step length[2] and $\nabla J(k)$ is the gradient of the MSE criterion with respect to the prototype vector, which is computed as:

$$\nabla J(k) = \frac{1}{N} \sum_{t=..N} \left(\hat{y}(t) - y(t)\right) \frac{d\hat{y}(t)}{d\boldsymbol{\alpha}(k)} \quad (17)$$

In SISO models, the derivative of the output vector is computed as (see (5)):

$$\frac{d\hat{y}(t)}{d\boldsymbol{\alpha}(k)} = \frac{d\mathbf{w}(t)}{d\boldsymbol{\alpha}(k)}.\boldsymbol{\theta} \quad (18)$$

where Jacobian matrix of the fire strength vector is computed as:

$$\frac{d\mathbf{w}(t)}{d\boldsymbol{\alpha}(k)} = \begin{bmatrix} \frac{\partial w_1(t)}{\partial a_1(k)} & \cdots & \frac{\partial w_m(t)}{\partial a_1(k)} \\ \vdots & \ddots & \vdots \\ \frac{\partial w_1(t)}{\partial a_m(k)} & \cdots & \frac{\partial w_m(t)}{\partial a_m(k)} \end{bmatrix}. \quad (19)$$

By using a formulation of the membership function is used in such a way that strong fuzzy partitions (3) are always obtained as shown in Figure 4, the membership function $\mu_{A_i}(x)$ of the fuzzy set $\tilde{A}_i$ is computed as:

$$\mu_{A_i}(x) = \begin{cases} f_i(x) = \dfrac{x - a_{i-1}}{a_i - a_{i-1}}, \ if \ a_{i-1} < x \le a_i \\ g_i(x) = \dfrac{a_{i+1} - x}{a_{i+1} - a_i}, \ if \ a_i < x \le a_{i+1} \end{cases} \quad (20)$$

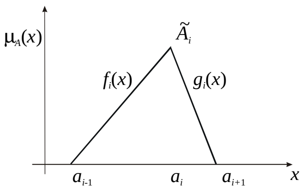

Figure 4: Fuzzy set formulation.

In the structure identification method the trapezoidal fuzzy sets in the domain limits are not optimized. In this case, only a few elements of the Jacobian matrix (19) will be different from zero and:

$$\frac{\partial w_i(t)}{\partial a_{i-1}(k)} = \frac{\partial f_i(x(t))}{\partial a_{i-1}(k)} \quad (21)$$

$$\frac{\partial w_i(t)}{\partial a_i(k)} = \frac{\partial f_i(x(t))}{\partial a_i(k)} + \frac{\partial g_i(x(t))}{\partial a_i(k)} \quad (22)$$

$$\frac{\partial w_i(t)}{\partial a_{i+1}(k)} = \frac{\partial g_i(x(t))}{\partial a_{i+1}(k)} \quad (23)$$

The derivatives of the membership functions with respect to parameter that represents the center of the triangle $\frac{\partial f_i(x(t))}{\partial a_i(k)}$ and $\frac{\partial g_i(x(t))}{\partial a_i(k)}$ will never be different from zero simultaneously. This avoids the problems that could arise by the fact that the triangle membership function is not continuously differentiable.

The extension of the structure optimization procedure for MISO models is achieved by considering the combination of membership functions in (8).

3.3 Parameter optimization

The fuzzy model identification algorithm aims to determine the prototype set that defines the fuzzy partition for each variable and the rule output parameter vector $\boldsymbol{\theta}$.

Consider the matrix $\mathbf{W} \in I^{N \times m}$ that stores the fire strenth of all points in the training set, such that:

$$\mathbf{W} = \begin{bmatrix} w_1(1) & \dots & w_m(1) \\ \vdots & \ddots & \vdots \\ w_1(N) & \dots & w_m(N) \end{bmatrix} \quad (24)$$

where M is the number of all premise term combinations of the SISO model rules. The matrix $\mathbf{W}$ is often called *regressors' matrix* in the system identification literature [12].

The parameters $\boldsymbol{\theta}$ in (7) are computed by the minimization of the residual norm. When the quadratic normis used thi optimization problem is known as the *Least Square Error* (LSE) problem [13]

$$\min_{\boldsymbol{\theta}} \|\mathbf{W}\boldsymbol{\theta} - \mathbf{y}\|_2, \quad (25)$$

Robust algorithms for the solution of the least square problem can be obtained from the *singular value decomposition* (SVD) of the regressors' matrix [13].

The parameter optimization is combined with the gradient descent learning as in ANFIS adaptation in a hybrid-learning rule. In each iteration k of the gradient descent algorithm used to update fuzzy sets location parameters (the prototype vector) is followed by LSE minimization to identify the linear output parameters.

The extension of the above presented structure and parameter optimization method for MISO systems is straightforward.

[2] The step length of the gradient descent algorithm is often called learning rate in the neural networks literature.

4 Applications

Two numerical examples found in the literature are used in this section to illustrate the performance of the presented approach. The data for the two examples are derived from real processes.

Example 1: This example deals with the modeling of human operator control in a chemical plant. The data provided in [14] contain 70 data points with 5 input variables and one output. To avoid the "curse of dimensionality" due to the number of input variables, the most important variables were selected to build the model. A general discussion on the selection of input variables in fuzzy modeling, (and in this particular problem), can be found in [14]. According to the regularity criterion presented by Sugeno and Yasukawa [14], the input variables u_2 and u_3 were selected to build the fuzzy model in this example. The training data are thus composed by samples in the form $(\mathbf{x}(t), y(t))$, where $\mathbf{x}(t) = [u_2(t) \quad u_3(t)]$ and $y(t)$ is the output variable.

A very simple fuzzy model obtained at the initialization phase, with only 4 rules and no learned points, is shown in Figure 5.

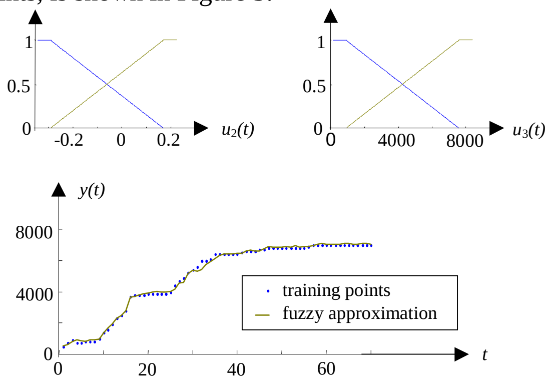

Figure 5: Initial model.

The meanings of the terms "small" and "big" for each variable are given by the fuzzy partitions of variables u_2 and u_3, as shown respectively in the left and right top graphs in Figure 5. The output variable values in the training set and the fuzzy model approximation are plotted as a function of time in the lower part of Figure 5.

This model already gives acceptable results. A more precise result, obtained with a fuzzy model described by $M = 20$ rules shown in Figure 6. It can be noted from Figure 6 that the learned points are located in the region where the approximation has the greatest error. This is the main characteristic of the structure identification method, which allows an automatic discretization of the input variable domain.

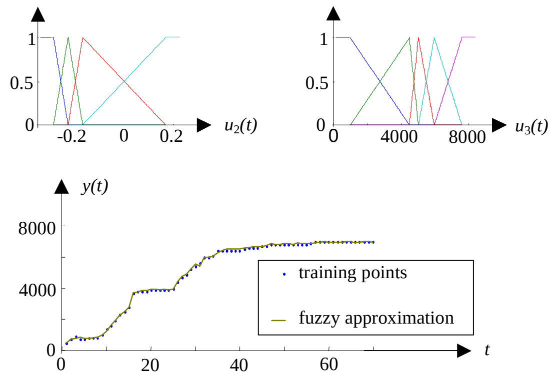

Figure 6: Final model.

Example 2: This example is the well-known Box & Jenkins data set, where the system to be modeled is a gas furnace [15]. The original data set is composed of 296 pairs, where the input variable is the methane concentration (the airflow is kept constant) and the output variable is the CO_2 concentration.

This data set has been used to evaluate several system identification methods, but the data have not always been used in the same way. For a comparative study, the discrete formulation used by [11] has been used here. A prediction model was built using some past outputs of the system as model inputs. In this example, the original, one input data set, has been converted into a two input data set. Each training point is thus expressed in the form $(\mathbf{x}(t), y(t))$, where $\mathbf{x}(t) = (u(t-3), y(t-1))$.

A first model containing $M = 36$ rules has been generated with the modified data set (Figure 7). As in the previous example, the fuzzy partitions obtained for both input variables (respectively $x_1(t) = u(t-3)$ and $x_2(t) = y(t-1)$) are shown respectively in the left and right top graphs in Figure 7. The output variable values in the training set and the fuzzy model approximation are plotted as function of time in the lower part of Figure 7.

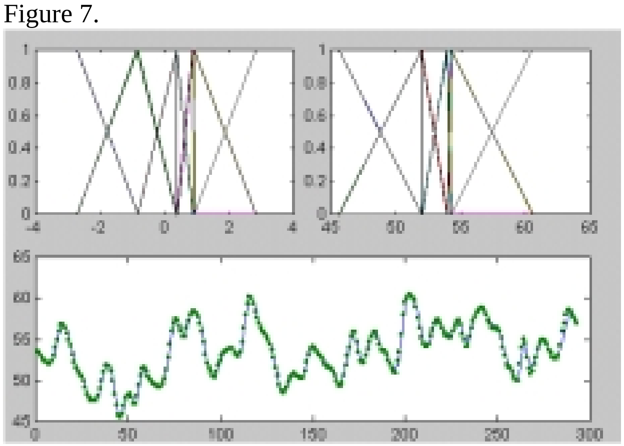

Figure 7: Results with $M = 36$.

To allow a direct comparison with the results presented by [11], a model described by $M = 90$ rules was also generated. The number of rules M and the Mean Squared Error (MSE) (9) of the models obtained in the present approach as well as the models presented by [11] and other authors are shown in Table 1.

It can be seen that the results obtained by the present approach are much better than those obtained in [11]. The present approach allows the generation of a model described by 36 rules that has a better performance than the 90-rule model provided by Nakoula and al. [11].

Table 1: Comparison of the results.

Method	*M*	MSE
Box and Jenkins [15]	-	0.710
Tong [16]	19	0.469
Xu and Lu [17]	25	0.328
Wang and Langari [9]	5	0.158
Nakoula and al. [11]	90	0.175
Present	36	0.153
Present	90	0.090

5 Conclusion

A method for structure and parameter optimization of fuzzy models has been proposed. The main characteristic of the method is the automatic determination of the number of the fuzzy sets and the optimization of the location of the vertices of their triangular membership functions in the domain of each input variable. The structure identification method, which is not based on clustering techniques, is original and quite simple. The resulting model can be written either as a TSK rule base allowing a model linguistic interpretation.

The main contribution of this work is the extension of the identification algorithm to allow the optimization of the fuzzy sets location by a gradient descent algorithm. The contribution of this work can also be seen as an extension of the ANFIS neuro-fuzzy system to allow automatic determination of the fuzzy system structure.

The applications show that the improvement suggested in this work allows a much better performance of the identification algorithm.

The structure determination and the optimization of rule output parameters are separated. If more precision is required, more complex fuzzy models such as first order TSK models can be used.

The perspectives of this work include the extension of the method to the identification of multi-dimensional fuzzy sets in the input variable domain.

Acknowledgements

Alexandre Evsukoff is supported by Rio de Janeiro State Research Council – FAPERJ, under the grant 151.233/98.

References

[1] Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems, Second Edition*. John Wiley & Sons.

[2] Wang, L. X. (1994). *Adaptative Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Clifs, NJ, Prentice-Hall.

[3] Kosko, B. (1997) *Fuzzy Engineering*. Prentice Hall.

[4] Jang, J.-R. S.; C.-T. Sun and E. Mizutani (1997). *Neuro-fuzzy and Soft Computing – A Computational Approach to Learning and Machine Intelligence*. Prentice Hall.

[5] Willaeys, D. and N. Malvache (1981). The use of fuzzy sets for the treatment of fuzzy information by computer. *Fuzzy Sets and Systems*, **5**, pp 323-327.

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

[7] Sugeno, M. and G. T. Kang (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, **42**, pp 15-33.

[8] Yager, R. R. and D. P. Filev (1994). *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons.

[9] Wang, L. and R. Langari (1996). Complex systems modeling via fuzzy logic. *IEEE Trans. on Systems Man an Cybernetics*, **26** (1), pp. 100-106.

[10] Nakoula, Y.; S. Galichet and L. Foulloy (1996). Simultaneous learning of rules and linguistic terms. *Proc. Int. Conf. on Fuzzy Systems – FUZZ IEEE'96*, New Orleans, pp 1743-1749.

[11] Nakoula, Y.; S. Galichet and L. Foulloy (1997). A learning method for structure and parameter identification of fuzzy linguistic models. In H. Hellendoorn and D. Driankov (eds.) *Selected Approaches to Fuzzy Model Identification*, Springer Verlag, pp.281-319.

[12] Yen, J. and L. Wang (1999). Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE Trans. on Systems Man an Cybernetics*, Part B, **29** (1), pp. 13-24.

[13] Golub, G. H. and C. F. Van Loan (1989). *Matrix Computations*, John Hopkins, Baltimore.

[14] Sugeno, M. and T. Yasukawa (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Trans. on Fuzzy Systems*, **1** (1), pp. 7-31.

[15] Box, G. E. P. and G. M. Jenkins (1970). *Time Series Analysis: Forecasting and Control*. San Francisco, Holden Day.

[16] Tong, R. M. (1980). The evaluation of fuzzy models derived from experimental data. *Fuzzy Sets and Systems*, **4**, pp 1-12.

[17] Xu, C. W. and Y. Z. Lu (1987). Fuzzy model identification and self learning for dynamic systems. *IEEE Trans. on Systems Man an Cybernetics*, **17** (4), pp. 683-689.