"Development of Interpretable Models through Neuro-Fuzzy Networks"

R. P. Paiva*, A. Dourado*

*CISUC – Centro de Informática e Sistemas da Universidade de Coimbra
Departamento de Engenharia Informática, PÓLO II da Universidade de Coimbra,
Pinhal de Marrocos, P 3030, Coimbra, Portugal
Tel. +351-239-790000, Fax: +351-239-701266, e-mail: {ruipedro, dourado}@dei.uc.pt

ABSTRACT

In this paper a methodology for development of linguistically interpretable fuzzy models is presented. The implementation of the model is conducted through the training of a neuro-fuzzy network, i.e., a neural net architecture capable of representing a fuzzy system. In the first phase the structure of the model is obtained by subtractive clustering, which allows the extraction of a set of relevant rules based on a set of representative input-output data samples. In the second phase, the model parameters are tuned via the training of a neural network through backpropagation. In order to attain interpretability goals, the method described imposes some restrictions on the tuning of parameters and performs membership function merging. In this way, it will be easy to assign linguistic labels to each of the membership functions used. Therefore, the model obtained for the system under analysis will be described by a set of linguistic rules, easily interpretable.

Keywords: system identification, fuzzy systems, neuro-fuzzy networks, clustering, interpretability

1 INTRODUCTION

Nowadays, information is playing a more and more relevant role in society. This circumstance is notorious not only in complex industrial production systems but also in simple leisure activities. Several studies have already been conducted in terms of development of modeling and control algorithms for industrial systems based on so-called intelligent techniques, as a means of integrating "intelligence" in production systems. Fundamentally, such developments aim to overtake some of the limitations and difficulties associated with classical methodologies. In this context, more precisely in system modeling, sometimes it is necessary that the resulting models have some transparency, i.e., that their information be interpretable, so as to permit a deeper understanding of the system under study. It is in this point that fuzzy modeling founds its maximum potential. In fact, fuzzy models have some properties that make them particularly interesting, namely universal approximation [1] and the possibility of linguistic interpretation [11], being the former hardly attained via MLP (Multi-Layer Perceptrons). However, they have associated an important limitation, which results from the difficulty to quantify the fuzzy linguistic terms. Therefore, neuro-fuzzy nets appear as an attempt to conjugate the advantages of fuzzy systems in terms of transparency with the advantages of neural networks regarding learning capabilities.

The methodology presented is carried out in two main phases: in the first one, structure learning is performed, i.e., a set of fuzzy rules is obtained; in the second one, the model parameters are tuned, i.e., the parameters of the membership functions of the fuzzy system. The generality of strategies developed by several authors, based on the scheme referred, aim fundamentally to obtain models with high prediction accuracy. Clearly, if this is the main goal, it is questionable whether fuzzy modeling is the most adequate strategy, as it is pointed out in [6]. However, in case model transparency is a fundamental goal, the strategy referred does not guarantee anything regarding that objective. In fact, since parameter tuning is carried out under no restrictions, highly complex fuzzy data bases may come up.

Therefore, this paper tries to explore the potential of neuro-fuzzy networks to help getting real transparent models. Thus, linguistic models, i.e., models whose consequents are fuzzy sets, are used instead of Takagi-Sugeno [10] models, where the consequents implement, typically, first order linear functions, thus, difficult to interpret linguistically. Additionally, parameter learning is restricted and similar membership functions are merged, in order to ease the assignment of linguistic labels to the final functions.

So, in Section 2 the main issues of fuzzy identification are introduced. In Section 3 subtractive clustering, used for structure learning is presented, after what the unrestricted parameter learning strategy is described in Section 4. In Section 5, the strategies for

implementation of interpretable models are presented. The methodologies are applied to the Mackey-Glass chaotic time series, in Section 6. Finally, some conclusions are drawn in Section 7.

2 FUZZY IDENTIFICATION

Dynamical system identification deals with the implementation of models using experimental data. Thus, when a model is developed based on the theory of system identification, its parameters are tuned according to some criteria, aiming to obtain a final representation adequate for the modeling purposes. In this sense, fuzzy identification is presented as a particular case of system identification, in which the model is included in the class of fuzzy systems.

Thus, without loss of generality, let us assume a single-input single-output (SISO) model, with one input, u, and one output, y, from where N data samples are collected (1):

$$Z^N = [[u(1), y(1)], [u(2), y(2)], ..., [u(N), y(N)]] \quad (1)$$

Using the data collected, the goal is to derive a fuzzy model, represented by a set of rules of type R_i (2):

$$R_i: If \quad y(t-1) \text{ is } A_{1i} \text{ and } u(t-d) \text{ is } B_{1i} \text{ then } \hat{y}(t) \text{ is } C_{1i} \quad (2)$$

where d represents the system delay time and A_{ji}, B_{ji} and C_{ji} denote linguistic terms associated to each input and output. Those terms are defined by their respective membership functions $\mu_{A_{ji}}, \mu_{B_{ji}}, \mu_{C_{ji}}$. In this way, the previous structure is called a *FARX* structure (Fuzzy Auto Regressive with eXogenous inputs) as a generalization of the well-known ARX structure. Thus, the selection of a set of rules of type (2), as well as the definition of the fuzzy sets A_{ji}, B_{ji} and C_{ji}, constitute some project issues specific to fuzzy systems.

3 STRUCTURE LEARNING

In order to obtain a set of g fuzzy conditional rules, capable of representing the system under study, clustering algorithms are particularly suited, since the permit a scatter partitioning of the input-output data space, which results in finding only the relevant rules. Comparing to grid-based partitioning methods, clustering algorithms have the advantage of avoiding the rule base explosion, i.e., the curse of dimensionality.

In this paper, Chiu's subtractive clustering is applied [2]. This scheme possesses some interesting advantages, especially in a neuro-fuzzy identification context. In fact, the algorithm is characterized by its efficiency and for being suited for the initialization of iterative optimization procedures, as is the case.

Chiu's algorithm belongs to the class of potential function methods, being, more precisely, a variation of the mountain method (see [3]). In this class, a set of points is defined as possible group centers, each of them being interpreted as an energy source. In subtractive clustering, the center candidates are the data samples themselves. In this way, the main limitation of the mountain method, i.e., curse of dimensionality resulting from defining candidates in a grid, is overtaken.

So, let Z^N (1) be a set of N data samples, z_1, z_2, …, z_N, defined in a $m+n$ space, where m denotes the number of inputs and n the number of outputs. In order to make the range of values in each dimension identical, the data samples are normalized, so that they are limited by a hypercube.

As was referred, it is admitted that each of the samples defines a possible cluster center. Therefore, the potential associated to z_i is (3):

$$P_i(z_i, Z^N) = \sum_{j=1}^{N} e^{-a\|z_i - z_j\|^2}, \quad a = \frac{4}{r_a^2}, \quad i = 1, 2, \cdots, N \quad (3)$$

where $r_a > 0$ is *radii*, a constant which defines the neighborhood radius of each point. Thus, points z_j located outside of the radius of z_i will have a reduced influence in its potential. On the other hand, the effect of points close to z_i will grow with their proximity. In this way, points with a dense neighborhood will have higher potentials associated.

After computing the potential for each point, the one with the highest potential is selected as the first cluster center. Next, the potential of all the remaining points is reduced. Defining z_1^* as the first group center and denoting its potential as P_1^*, the potential of the remaining points is reduced as in (4), where the constant $r_b > 0$ defines the neighborhood radius with sensible reductions in its potential.

$$P_i \leftarrow P_i - P_1^* e^{-b\|z_i - z_1^*\|^2}, \quad b = \frac{4}{r_b^2} \quad (4)$$

In this way, points close to the chosen center will have their potential reduced in a more significant manner, and so the probability of being selected as centers decreases. This procedure has the advantage of avoiding the concentration of identical clusters in dense zones. Therefore, r_b is selected to be a bit higher than r_a, so as to avoid closely spaced clusters. Typically, $r_b = 1.5\, r_a$.

After performing the potential reduction for all the candidates, the one with the highest potential is selected as the second cluster, after what the potential of the remaining points is again reduced. Generically, after determining the r^{th} group, the potential is reduced as (5):

$$P_i \leftarrow P_i - P_r^* e^{-b\left\|z_i - z_r^*\right\|^2} \quad (5)$$

The procedure of center selection and potential reduction is repeated until a stopping criterion is reached [3].

As can be understood from the description of the algorithm, the number of clusters to obtain is not pre-specified. However, it is important to note that the parameter *radii* is directly related to the number of clusters found. Thus, a small radius will lead to a high number of rules, which, if excessive, may result in overfitting. On the other hand, a bigger radius will lead to a smaller number of clusters, which may originate underfitting and so, models with reduced representation accuracy. Therefore, in practice it is necessary to test several values for radii and select the most adequate according to the results obtained.

After applying subtractive clustering, each of the clusters obtained will constitute a prototype for a particular behavior of the system under analysis. So, each cluster can be used to define a fuzzy rule able to describe the behavior of the system in some region of the input-output space. Typically, g fuzzy conditional rules of type (6) are obtained:

Rule r:

$$\begin{aligned} &\text{IF } (X_1 \textit{ is } LX1^{(r)}) \textit{ AND } (X_2 \textit{ is } LX2^{(r)}) \textit{ AND } \ldots \\ &\quad \textit{AND } (X_m \textit{ is } LXm^{(r)}) \\ &\text{THEN } (Y_1 \textit{ is } LY1^{(r)}) \textit{ AND } (Y_2 \textit{ is } LY2^{(r)}) \\ &\quad \textit{AND } \ldots \textit{AND } (Y_n \textit{ is } LYn^{(r)}) \end{aligned} \quad (6)$$

where each of the linguistic terms in the antecedent, $LXj^{(r)}$, has associated a membership function defined as follows (7):

$$\mu_{LXj^{(r)}}(x_j) = e^{-a\left\|x_j - x_{rj}^*\right\|^2}, \quad r = 1,2,\cdots,g;\ j = 1,2,\cdots,m \quad (7)$$

Here, x_j denotes a numeric value regarding the j^{th} input dimension and x_{rj}^* is the j^{th} coordinate in the m-dimensional vector x_r^*. Equation (7) results from the computation of the potential associated to each point in the data space.

As for the consequents, it will be possible to associate them either a fuzzy set or a constant directly. In the present case, linguistic models are derived, so fuzzy sets are utilized (8):

$$\mu_{LYo^{(r)}}(y_o) = e^{-a\left\|y_o - y_{ro}^*\right\|^2}, \quad o = 1,2,\cdots,n \quad (8)$$

where y_o denotes a numeric value regarding the o^{th} output dimension and y_{ro}^* is the j^{th} coordinate in the n-dimensional vector y_r^*.

Comparing (7), (8) and the general equation for Gaussian functions, it becomes clear that the membership functions considered belong to the type referred. Thus, regarding the standard deviation of each function, expression (9) is obtained trivially:

$$s_{rj} = \frac{r_a}{\sqrt{8}} \quad (9)$$

4 PARAMETER LEARNING

After determining a fuzzy model structure, the model parameters, i.e., the centers and standard deviations of the Gaussian membership functions, should be tuned. In the present work, such task is performed by training a fuzzy neural network based on [4] (Figure 1).

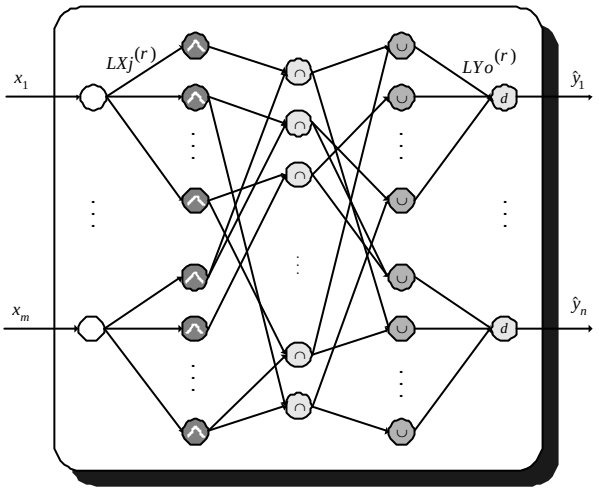

Figure 1. Neuro-fuzzy network.

Basically, the previous network is composed by five layers: an input layer, a fuzzification layer, a rule layer, an union layer and an output or defuzzification layer, sequentially.

In order to make the next expressions more readable, the notation used is presented beforehand:

- $a_i^{(p2)}$: activation of the neuron i in layer 2, regarding the training pattern p (i denotes an input term: "*input*");
- $a_r^{(p3)}$: activation of the neuron r in layer 3, regarding the pattern p (r denotes "*rule*");
- $a_s^{(p4)}$: activation of the neuron s in layer 4, regarding the pattern p (s denotes "*S-norm*");
- $a_o^{(p5)} = y_o^{(p)}$: activation of the neuron o in layer 5, i.e., output, regarding the pattern p (o denotes "*output*");
- $y_o^{(p)}$: desired activation for neuron o in layer 5, i.e., for the network output, regarding pattern p.

In the structure presented, the *input layer* simply receives data from the external environment and passes

them to the next layer.

In the second layer, the *fuzzification layer*, each of the cells corresponds to a membership function associated to each of the inputs. Since this work assumes the goal of obtaining interpretable models, two-sided Gaussian functions are used (Figure 2). Thus, the activation of each of the neurons in this layer is given by (10).

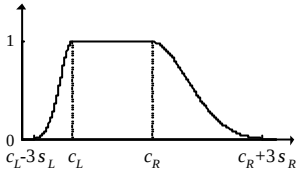

Figure 2. Two-sided Gaussian function.

$$a_i^{(p2)} = \begin{cases} e^{-\frac{(x_j^{(p)} - c_{ijL})^2}{2s_{ijL}^2}} & , x_j^{(p)} < c_{ijL} \\ 1 & , c_{ijL} \le x_j^{(p)} \le c_{ijR} \\ e^{-\frac{(x_j^{(p)} - c_{ijR})^2}{2s_{ijR}^2}} & , x_j^{(p)} > c_{ijR} \end{cases} \quad (10)$$

Here, c_{ijL} and s_{ijL} represent, respectively, the center and standard deviation of the left component of the i^{th} membership function related to the j^{th} input. For the right component, the index R is used. Such parameters constitute the weights of the layer one to layer two links ($LXj^{(r)}$ in Figure 1). In the same expression, $x_j^{(p)}$ denotes the p^{th} pattern associated do input j.

As for the neurons in the *rule layer*, their function consists of performing the antecedent conjunction of each rule, by means of some T-norm. By experimental testing, it was concluded that truncation operators (e.g., minimum) lead to better results than algebraic operators (e.g., product), when interpretability is desired. So, operator minimum is selected for fuzzy conjunction (11):

$$a_r^{(p3)} = T - \underset{i=1}{\overset{na_r}{norm}}\left(a_i^{(p2)}\right) = \min_{i=1}^{na_r}\left(a_i^{(p2)}\right) \quad (11)$$

where na_r stands for the number of inputs in the antecedent of rule r.

The fourth layer, called the *union layer*, is responsible for integrating the rules with the same consequents, via some S-norm. Once again, truncation operators are preferred, namely operator maximum (12).

$$a_s^{(p4)} = S - \underset{r=1}{\overset{nr_s}{norm}}\left(a_r^{(p3)}\right) = \max_{r=1}^{nr_s}\left(a_r^{(p3)}\right) \quad (12)$$

There, nr_s stands for the number of rules which have the neuron s as consequent.

As for the *output layer*, or *defuzzification layer* (*d*, in Figure 1), the layer four to layer five links ($LYo^{(r)}$ in the same figure) define the parameters of the membership functions associated to the output linguistic terms. Thus, based on these membership functions and on the activation of each rule, its neurons should implement a defuzzification method suited for two-sided Gaussian functions, as the one presented in [8] (13):

$$\hat{y}_o^{(p)} = a_o^{(p5)} = \frac{\sum_{s=1}^{|T(Y_o)|} \frac{1}{2}(c_{osL}\, s_{osL} + c_{osR}\, s_{osR})\, a_s^{(p4)}}{\sum_{s=1}^{|T(Y_o)|} \frac{1}{2}(s_{osL} + s_{osR})\, a_s^{(p4)}} \quad (13)$$

where c_{osL} and s_{osL} represent the center and standard deviation of the left component of the s^{th} membership function related to output o. In the previous expression, $|T(Y_o)|$ stands for the number of membership functions associated to each linguistic output variable Y_o. The main idea of the defuzzification method proposed is to weight the activation of each rule, not only by the centers, right and left, but also by their standard deviations.

Based on the function performed by each neuron, the network is trained in batch mode, via the well-known backpropagation algorithm. The implementation of the training methodology referred is described with some detail in [8].

5 INTERPRETABILITY

The philosophy of fuzzy systems lays on the possibility of linguistic interpretation that characterizes them. Nevertheless, this issue is often ignored, being given prevalent relevance to the factors related to approximation capabilities. However, as Nauck and Kruse refer [6], in case interpretability is not a major concern, it is important to consider other methods, which might be more adequate.

Thus, as for the neuro-fuzzy identification strategy described in the previous sections, some questions are addressed regarding the model transparency after learning. In fact, the unrestricted parameter adjustment may lead to a highly complex set of membership functions, for which it will be difficult to assign linguistic labels. Therefore, it is important to impose adequate restrictions for parameter tuning, so that interpretability is attained. In the same way, two-sided

Gaussian functions are appealing due to their increased flexibility, which permits to control function overlapping and improves function distinguishability.

Therefore, three main criteria for model interpretability are defined. The first one, and most important, is related to the point just referred, i.e., function distinguishability. The others come from human cognitive issues, according to which the number of rules and the number of membership functions associated to each variable should not be excessive. In the present case, these issues are monitored by the modeler, in order to obtain a satisfactory trade-off between model accuracy and interpretability.

5.1. Similar Membership Function Merging

The first step in order to attain model interpretability consists of finding and merging similar membership functions.

Structure learning by means of clustering techniques leads to initial membership functions with a high similarity degree. That makes the model lack transparency and originates an excessive number of parameters to adjust, and the consequent computational cost. Thus, it seems useful to merge functions with a high enough similarity degree.

In order to perform function merging, directed to rule base simplification, it was concluded in [9] that S_1 (14) is a very adequate similarity measure. There, the similarity between two fuzzy sets A and B is given by the result of the division of the area of their intersection by the area of their union:

$$S_1(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} \tag{14}$$

where the fuzzy intersection and union are performed, respectively, by the operators minimum and maximum.

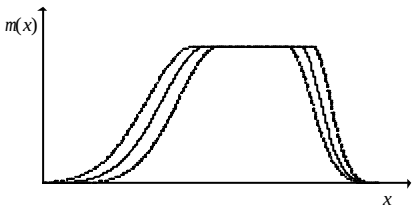

Figure 3. Membership function merging.

After detecting the most similar pair of membership functions, if their degree of similarity is above some threshold, those functions are merged. The new function results by averaging the parameters of the original functions, i.e., centers and standard deviations, as is depicted in Figure 3. There, the original functions are represented in dashed lines and the resulting function in solid line. The procedure of membership function merging, one pair in each iteration, continues until no more pairs satisfy the merging threshold.

As a result of function merging, the rule base is updated. In fact, the rules regarding the fuzzy sets merged will then contain the new function obtained. Thus, in the original rules, the premises and conclusions are updated so as to include the new terms. Therefore, the rule base may be simplified in case redundant rules are obtained. Besides that, situations of inconsistency may result, if rules with the same antecedents have different consequents. This may be a consequence of deficient structure learning or may indicate that the merging threshold should be adjusted.

5.2. Restricted Parameter Learning

After rule base simplification through function merging, it is essential to guarantee the interpretability is maintained during parameter optimization. Thus, it was decided to monitor the optimization procedure, so that function distinguishability is attained.

Thus, regarding the issue just referred, it is assumed that the overlapping degree between two functions is excessive in case the supreme of the support of the function to the left, i.e., its right "zero", goes beyond the right zero of the function to the right. The same reasoning applies to the left component of the functions. Formally, it turns out (15):

$$\begin{aligned} c_{kR} + 3s_{kR} &\le c_{iR} + 3s_{iR} \\ c_{kL} - 3s_{kL} &\le c_{jL} - 3s_{jL} \end{aligned} \tag{15}$$

where k refers to some membership function and i and j are, respectively, its right and left neighbor functions. In case function overlapping does not satisfy the constraints presented in (15), the standard deviations of function k are altered in order to keep those conditions. Therefore, the right and left components are changed as in (16) and (17), respectively:

$$s_{kR} = \frac{c_{iR} + 3s_{iR} - c_{kR}}{3} \tag{16}$$

$$s_{kL} = \frac{c_{jL} - 3s_{jL} - c_{kL}}{-3} \tag{17}$$

Besides overlapping monitoring, it was concluded that function distance should also be checked. This procedure aims to avoid inclusions, i.e., functions total or almost totally "inside" other functions. Furthermore, the fact that the functions are not too close also improves model interpretability. Thus, the constraint (18) for the minimal distance between functions was defined:

$$c_{iL} - c_{kR} \le a(X_{\max} - X_{\min})$$
$$c_{kL} - c_{jR} \le a(X_{\max} - X_{\min}) \quad (18)$$

where $a \in [0;1]$ denotes the percentage of the domain $[X_{\min}; X_{\max}]$ used for calculating the minimal distance allowed. In case this condition does not apply, the function centers are changed as follows (19):

$$c_{kR}^{new} = \frac{c_{kR} + c_{iL}}{2} - \frac{a(U_{\max} - U_{\min})}{2}$$
$$c_{iL}^{new} = \frac{c_{kR} + c_{iL}}{2} + \frac{a(U_{\max} - U_{\min})}{2} \quad (19)$$

In this situation, the new centers will be based on the average of the right and left original centers of the two functions compared, from which their values are altered in order to guarantee the distance required.

Despite the restrictions imposed, it may turn out that the final model is not sufficiently interpretable, as a result of the trade-off between interpretability and accuracy. Therefore, it is useful to perform function merging every x training epochs.

6 SIMULATION RESULTS

One of the most commonly used case studies in system identification consists of the prediction of the Mackey-Glass chaotic time series [5], described by equation (20).

$$\dot{x}(t) = \frac{0.2\,x(t-t)}{1+x^{10}(t-t)} - 0.1x(t) \quad (20)$$

The time series does not show a clear periodic behavior and it is also very sensible to initial conditions.

The problem consists of predicting future values of the series.

The application of the technique described previously is carried out based on identification data from the "*IEEE Neural Network Council, Standards Committee, Working Group on Data Modelling Benchmarks*", which are also used in the analysis of several other methodologies. So, in order to obtain a numeric solution the fourth order Runge-Kutta method was applied. For integration, it was assumed $x(t)=0$, $t<0$, and a time interval of 0.1. The initial condition $x(0)=1.2$ and the parameter $\tau=17$ were also defined. In this case, $[x(t\text{-}18), x(t\text{-}12), x(t\text{-}6), x(t)]$ are used to predict $x(t+6)$. Based on the parameterization described, data was obtained in the interval $t \in [0; 2000]$, after what 1000 input-output pairs were selected from $t \in [118; 1117]$. The data collected are depicted in Figure 4.

Using the samples obtained, the chaotic time series was modeled, according to the procedures described in the previous sections. Thus, the parameter r_a was assigned the value 0.5, resulting 9 fuzzy rules. Next, the network, with four inputs and one output, was trained, defining 0.65 for the merging threshold and $x = 200$.

So, after 800 epochs the RMS (Root Mean Square) error was 0.0228 for training data and 0.0239 for test data. As for the number of membership functions for the variables $x(t\text{-}18)$, $x(t\text{-}12)$, $x(t\text{-}6)$, $x(t)$ and $x(t+6)$, it resulted, respectively, 5, 4, 5, 4 and 5, leading to 92 adjustable parameters.

In Figure 5 the results obtained for test data are depicted. It can be seen that they are satisfactory.

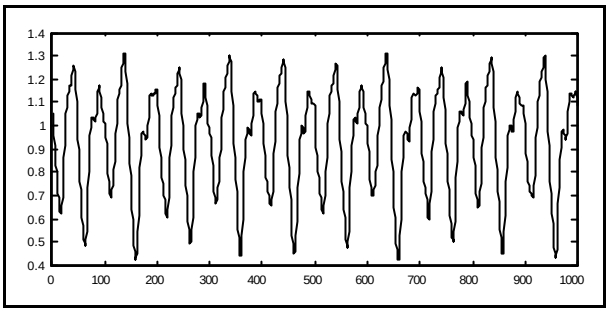

Figure 4. Chaotic time series: identification data.

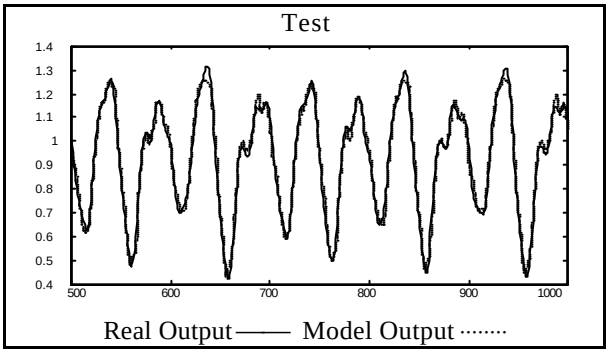

Figure 5. Chaotic series: output prediction.

As for membership functions, the results obtained are presented in Figure 6. As can be seen, it is not too difficult to assign linguistic terms to each of the membership functions. In the same figure, the labels *VS*, *S*, *M*, *B* and *VB* denote, respectively, the linguistic terms "very small", "small", "medium", "big" and "very big". Thus, the fundamental dynamics of the chaotic time series are interpreted according to Table 1.

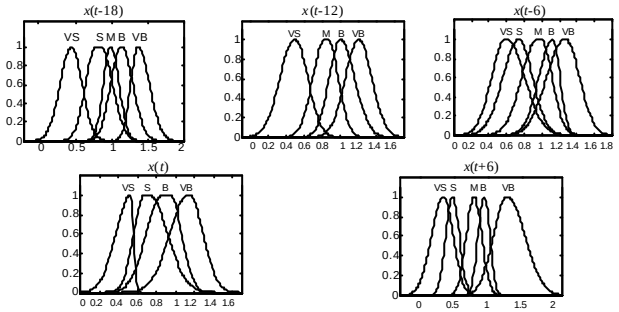

Figure 6. Membership functions obtained.

Rule	$x(t-18)$	$x(t-12)$	$x(t-6)$	$x(t)$	Þ	$x(t+6)$
1	M	VB	B	VB		B
2	B	VB	M	S		S
3	S	M	M	VB		VB
4	M	M	VS	VB		B
5	S	B	S	VS		M
6	S	VB	VB	B		M
7	S	VS	S	B		B
8	VS	VS	M	B		B
9	VB	VB	VB	B		VS

Table 1. Linguistic description of the series.

Comparing to NEXPROX [6] (Table 2), the results obtained are clearly better.

Method	Nr. Rules	Nr. Param.	RMSE
Paiva and Dourado	9	92	0.0239
NEFPROX (A)	129	105	0.0332
NEFPROX (G)	26	38	0.0671

Table 2. Chaotic series: comparison with other techniques.

7 CONCLUSIONS

In this paper a neuro-fuzzy methodology for the implementation of real interpretable fuzzy models is described. By the application of subtractive clustering, an initial structure for the fuzzy model was obtained, which is used for the initialization of a fuzzy neural network. However, adjusting membership function parameters without any constraints leads usually to complex overlapping between functions, which limits interpretability. Therefore, a learning scheme to allow the development of interpretable fuzzy models is proposed. The methodology presented is based on similar membership function merging and on constrains regarding parameter tuning, in order to improve function distinguishability in terms of distance and overlapping. The approach described is applied to the prediction of the Mackey-Glass chaotic time series, resulting a satisfactory trade-off between model accuracy and interpretability. However, it is important to point out that the results are not always acceptable. In fact, as complexity grows, the constraints imposed may lead to inaccurate models, which, consequently, are of no use. Clearly, it can be said that interpretability bounds accuracy and vice-versa.

REFERENCES

[1] Castro J. L. (1995). "Fuzzy logic controllers are universal approximators", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 4, pp. 629-635.

[2] Chiu S. L. (1994). "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems*, Vol. 2, No. 3, pp. 267-278.

[3] Davé R. N. and Krishnapuram R. (1997). "Robust clustering methods: a unified view", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 2, pp. 270-293.

[4] Lin C.- T. (1995). "A neural fuzzy control scheme with structure and parameter learning", *Fuzzy Sets and Systems*, Vol. 70, pp. 183-212.

[5] Mackey M. C. and Glass L. (1977). "Oscillation and chaos in physiological control systems", *Science*, vol. 197, pp. 287-289.

[6] Nauck D. and Kruse R. (1999). "Neuro-fuzzy systems for function approximation", *Fuzzy Sets and Systems*, Vol. 101, pp. 261-271.

[7] Paiva R. P. (1999). *Identificação Neuro-Difusa: Aspectos de Interpretabilidade* (Neuro-Fuzzy Identification: Interpretability Issues), MSc Thesis, Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Portugal (in Portuguese).

[8] Paiva R. P., Dourado A. and Duarte B. (1999). "Applying subtractive clustering for neuro-fuzzy modelling of a bleaching plant", *Proceedings of the European Control Conference - ECC'99*, CD-ROM.

[9] Setnes M. (1995). *Fuzzy Rule-Base Simplification Using Similarity Measures*, MSc Thesis, Department of Electrical Engineering, Delft University of Technology, The Netherlands.

[10] Takagi T. and Sugeno M. (1985). "Fuzzy identification of systems and its applications to

modelling and control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 1, pp. 116-132.

[11] Zadeh L A. (1973). "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, No.1, pp. 28-44.

ACKNOWLEDGEMENTS

This work was supported partially by the Portuguese Ministry of Science and Technology (MCT), under the program PRAXIS XXI.