"COMPARISON OF NEURO-FUZZY STRUCTURES FOR SYSTEM IDENTIFICATION"

R. P. Paiva*, A. Dourado*

*CISUC – Centro de Informática e Sistemas da Universidade de Coimbra

Department of Informatics Engineering, University of Coimbra, Portugal

ABSTRACT

In this paper an experimental comparison of neuro-fuzzy structures, namely linguistic and zero and first order Takagi-Sugeno, is developed. 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. Furthermore, different fuzzy operators are compared, as well as regular and two-sided Gaussian functions.

KEYWORDS: system identification, neural networks, fuzzy systems, neuro-fuzzy networks, clustering.

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, there is presently a growing interest on neuro-fuzzy techniques for system identification. 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 comes from the difficulty to quantify the fuzzy linguistic terms. Therefore, neuro-fuzzy nets appear as an attempt to make possible learning capabilities in fuzzy systems.

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.

Based on the generic methodology referred, this paper makes an analysis regarding some important issues in fuzzy modeling, e.g., type of rules, e.g., Takagi-Sugeno [10] or linguistic, type of operators and membership functions. In terms of functions, this study is restricted to simple and two-sided Gaussian functions.

So, in Section 0 the main issues of fuzzy identification are introduced. In Section 0 subtractive clustering, used for structure learning, is presented, after what the parameter learning strategies are described in Section 0. The methodologies are applied to the Mackey-Glass time series, in Section 0. Finally, some conclusions are drawn in Section 0.

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):

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

Using data collected form the system, the goal is to come up with a fuzzy model, represented by a set of rules of type R_i (2):

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

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 $m_{A_{ji}}, m_{B_{ji}}, m_{C_{ji}}$. 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.

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 is overtaken. In fact, there the candidates are defined in a grid, leading to the curse of dimensionality.

So, let Z^N (1) be a set of N data samples, $z_1, z_2, \ldots, z_N$, 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):

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

where $r_a>0$ is *radii*, a constant which defines the neighborhood radius of each point. Thus points z_j located out 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 the 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):

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

where the constant $r_b>0$ defines the neighborhood radius with sensible reductions in its potential.

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 remaining points is again reduced. Generically, after determining the r^{th} group, the potential is reduced as (5):

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

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:

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

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

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

Here, x_j denotes a numeric value regarding to 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. Clearly, expression (6) is a consequence of using linguistic models, i.e., models in which the consequents are fuzzy sets. Such consequents result naturally from the application of subtractive clustering and are obtained as follows (8):

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

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^*.

Obtaining an initial structure for Takagi-Sugeno models, in which the terms in the consequents are typically zero and first order linear functions, is performed similarly. However, since the consequents are not fuzzy sets, the initialization procedure just described applies only to the antecedents. In fact, based on the linear characteristics of the consequents, their values can be easily obtained by means of linear optimization techniques.

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:

$$s_{rj} = \frac{r_a}{\sqrt{8}} \tag{9}$$

Finally, after the parameterization of the Gaussian membership functions, the data used, normalized, are restored to their initial values. In the same way, the function parameters are adjusted to the domains defined for each dimension.

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. Therefore, it is necessary to select the type of model to use. In linguistic models, the conditional rules are of type (6). Regarding zero order Takagi-Sugeno models, with constant consequents, the rules are of type (10). As for first order Takagi-Sugeno models, the rules are of type (11), where $f_{or}(x)$ is defined as in (12).

Rule r:

IF (X_1 *is* $LX1^{(r)}$) *AND* (X_2 *is* $LX2^{(r)}$) *AND* … *AND* (X_m *is* $LXm^{(r)}$)
THEN ($y_1 = b_{1r}$) *AND* ($y_2 = b_{2r}$) *AND* … *AND* ($y_n = b_{nr}$) (10)

Rule r:

IF (X_1 *is* $LX1^{(r)}$) *AND* (X_2 *is* $LX2^{(r)}$) *AND* … *AND* (X_m *is* $LXm^{(r)}$)
THEN [$y_1 = f_{1r}(x)$] *AND* [$y_2 = f_{2r}(x)$] *AND* … *AND* [$y_n = f_{nr}(x)$] (11)

$$f_{or}(x) = b_{or0} + b_{or1}x_1 + b_{or2}x_2 + \cdots + b_{orm}x_m$$
$$b_{orj} \in \Re, \quad j = 1,2,\cdots,m; \quad o = 1,2,\cdots,n; \quad r = 1,2,\cdots,g \tag{12}$$

As can be seen, first order Takagi-Sugeno models are characterized by having more flexible consequents. Thus, these structures can be seen as smooth shifters between local linear models, which is an advantage comparing to the interpolative properties of both linguistic and zero order models.

The parameters of each of the fuzzy structures referred are adjusted by means of a fuzzy neural network, i.e., a neural network able to represent the functions of a fuzzy system, namely, fuzzification, fuzzy implication and defuzzification.

Basically, the nets presented in the following paragraphs are composed by an input layer, after which comes a fuzzification layer and then a rule layer. After the initial layers, the next layer is, in case of Takagi-Sugeno models, the final linear output layer. As for fuzzy consequents, the fourth layer is the union layer, used to integrate rules with the same consequents, and the last one is the output layer, responsible for defuzzification.

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*");

As for Takagi-Sugeno models, the output takes place in the fourth layer, resulting:

- $a_o^{(p4)} = y_o^{(p)}$: 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.

Takagi-Sugeno models

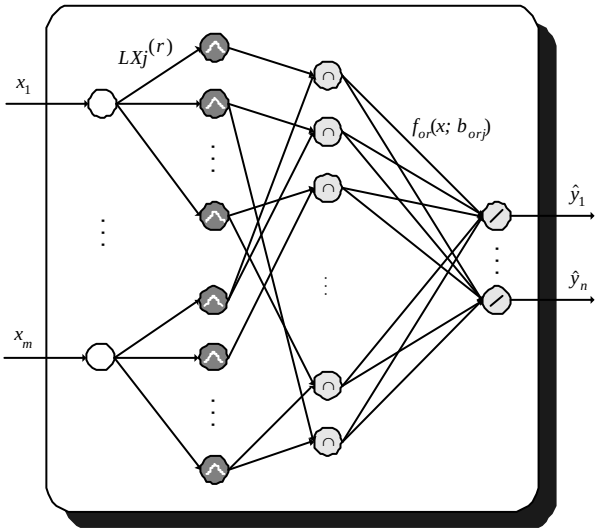

Figure 1. Neuro-fuzzy network: Takagi-Sugeno type

consequents.

From the described previously, Takagi-Sugeno structures are represented by the architecture in Figure 1. Naturally, the network presented serves both first and zero order models, in which the consequents will be either first order functions or constants, respectively.

In this structure, 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. Defining conventional Gaussian functions, the output of each neuron in this layer is given by (13):

$$a_i^{(p2)} = e^{-\frac{\left(x_j^{(p)} - c_{ij}\right)^2}{2s_{ij}^2}} \tag{13}$$

where c_{ij} and s_{ij} represent, respectively, the center and standard deviation of the i^{th} membership function related to the j^{th} input. Such parameters constitute the weights of 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*.

Alternatively, it is possible to define two-sided Gaussian functions, which are characterized by their possibility of being asymmetric and containing a plateau, as a generalization of conventional functions (Figure 2). Therefore, the possibility of coming up with better results can be formulated, due to the increased flexibility of the generalized functions.

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, e.g., product (14) or minimum (15). The first one is classified as an algebraic operator and the second one is a truncation operator.

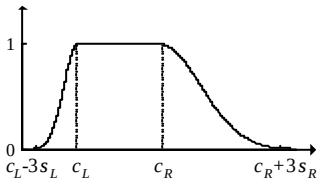

Figure 2. Two-sided Gaussian function.

$$a_r^{(p3)} = T - norm\left(a_i^{(p2)}\right) = \prod_{i=1}^{na_r} a_i^{(p2)} \tag{14}$$

$$a_r^{(p3)} = T - norma\left(a_i^{(p2)}\right) = \min_{i=1}^{na_r}\left(a_i^{(p2)}\right) \tag{15}$$

In the previous expressions, na_r stands for the number of inputs in the antecedent of rule *r*.

Regarding the *output layer*, its task consists of computing numeric outputs based on the level of activation of each rule. As referred previously, in zero order models the weights in this layer denote the rule consequents, defined by constants. Therefore, each output neuron is activated as in (16).

In the implementation of first order Takagi-Sugeno models, the net defines a fuzzy system with rules of type (11). In this way, the task of the output neurons is very similar to the previous, being defined as in (17).

$$\hat{y}_o^{(p)} = a_o^{(p4)} = \frac{\sum_{r=1}^{g} a_r^{(p3)} \cdot b_{0r}}{\sum_{r=1}^{g} a_r^{(p3)}} \tag{16}$$

$$\hat{y}_o^{(p)} = a_o^{(p4)} = \frac{\sum_{r=1}^{g} a_r^{(p3)} \cdot \left(\sum_{j=1}^{m} b_{orj}\, x_j^{(p)} + b_{or0}\right)}{\sum_{r=1}^{g} a_r^{(p3)}} \tag{17}$$

Fuzzy consequents

As a basis for dealing with fuzzy consequents, Lin defines in his architecture NFCN [5] a fuzzy neural net composed by five layers, as in Figure 3. However, the original structure is adapted in the present work, so as to allow different operators and membership functions.

Comparatively to the net depicted in Figure 1, the first three layers perform exactly the same tasks. Obviously, the difference lays in the following layers.

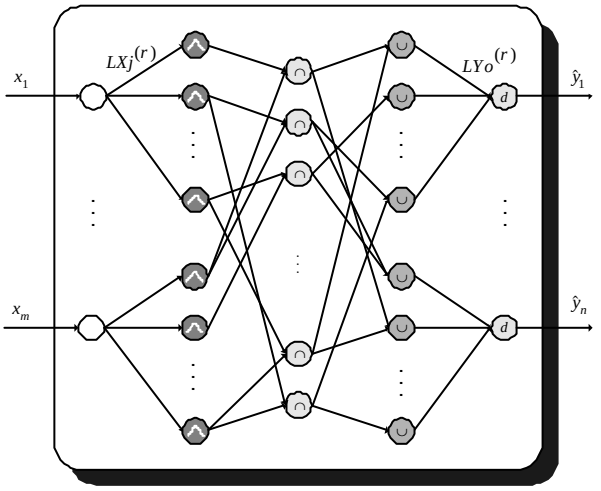

Figure 3. Neuro-fuzzy network: fuzzy consequents.

Thus, the fourth layer, called the *union layer*, is responsible for integrating the rules with the same consequents, via some S-norm, e.g., bounded sum (18) or maximum (19). The first one is classified as an algebraic operator and the second one is a truncation operator. There, nr_s stands for the number of rules which have the neuron *s* as consequent.

$$a_s^{(p4)} = S - norm\left(a_r^{(p3)}\right) = \min\left(1, \sum_{r=1}^{nr_s} a_r^{(p3)}\right) \quad (18)$$

$$a_s^{(p4)} = S - norm\left(a_r^{(p3)}\right) = \max_{r=1}^{nr_s}\left(a_r^{(p3)}\right) \quad (19)$$

As for the *output layer*, or *defuzzification layer* (*d*, in Figure 3), 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 fuzzy consequents, as the one presented in [5] (20):

$$\hat{y}_o^{(p)} = a_o^{(p5)} = \frac{\sum_{s=1}^{|T(Y_o)|} c_{os}\, s_{os}\, a_s^{(p4)}}{\sum_{s=1}^{|T(Y_o)|} s_{os}\, a_s^{(p4)}} \quad (20)$$

$$\hat{y}_o^{(p)} = a_o^{(p5)} = \frac{\sum_{s=1}^{|T(Y_o)|} \frac{1}{2}\left(c_{osL}\, s_{osL} + c_{osR}\, s_{osR}\right) a_s^{(p4)}}{\sum_{s=1}^{|T(Y_o)|} \frac{1}{2}\left(s_{osL} + s_{osR}\right) a_s^{(p4)}} \quad (21)$$

In (20), c_{os} and s_{os} represent the center and standard deviation of the s^{th} membership function related to output *o*. In case two-sided Gaussians are used, equation (21) results, as is defined in [9].

In the previous expressions, $|T(Y_o)|$ 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. Clearly, expression (21) is equivalent to equation (20) in case one deals with regular Gaussian functions.

Based on the function performed by each neuron, the linguistic networks are trained in batch mode, via the well-known backpropagation algorithm. Regarding Takagi-Sugeno models, several alternatives are applicable. In fact, the training can be also conducted through backpropagation. However, as a consequence of the linearity in the output layer, it is possible to apply the least square estimator in matrix form. This strategy has the advantage of leading to a significant reduction of the number of epochs required. However, the time necessary for each epoch will be greater. The implementation of the training methodologies referred is described with some detail in [8].

SIMULATION RESULTS

One of the most commonly used case studies in system identification consists of the prediction of the Mackey-Glass chaotic time series [6], described by equation (22):

$$\dot{x}(t) = \frac{0.2\, x(t-t)}{1 + x^{10}(t-t)} - 0.1 x(t) \quad (22)$$

It 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 techniques 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 interval $t \in [0; 2000]$, after what 1000 input-output pairs were selected from interval $t \in [118; 1117]$. 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.

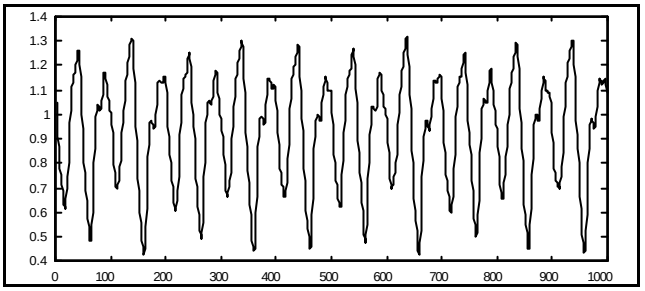

Figure 4. Chaotic time series: identification data.

After the application of the set of methodologies described, the results presented in Table 1 were obtained, where FC, CC and FOC denote respectively fuzzy consequents, constant and first order consequents.

Method		Gaus.	Nr. Par.	Fuz. Op.	Nr. Ép.	Time p/ Ép.	RMSE	
							Train	Test
1	FC	2-sid.	180	Alg.	2000	0.27s	0.0070	0.0076
2	"	"	"	Trunc.	"	0.24s	0.0111	0.0121
3	"	Reg.	90	Alg.	"	0.26s	0.0066	0.0071
4	CC	2-sid.	153	Alg.	1500	0.54s	0.0047	0.0050
5	"	"	"	Trunc.	"	0.52s	0.0097	0.0108
6	"	Reg.	81	Alg.	"	0.53s	0.0050	0.0052
7	FOC	2-sid.	189	Alg.	300	4.1s	0.0025	0.0030

8	“	“	“	Trunc.	“	4.3s	0.0038	0.0043
9	“	Reg.	147	Alg.	“	4.0s	0.0030	0.0033

Table 1. Chaotic series: training results.

The results presented allow some conclusions to be drawn. Thus, using regular Gaussian functions presents some advantages in case fuzzy consequents are utilized. However, the higher complexity that results from the use of two-sided Gaussian functions does not originate a significant gain in terms of model accuracy, and so regular Gaussians are preferable. As for fuzzy operators, algebraic operators lead to much better results than truncation operators do. Models with constant consequents allow better results than linguistic models but first order models are the most accurate and need a considerable lower number of training epochs. However, linear optimization leads to high processing time, which may be problematic for real-time learning. As a consequence, zero order models appear to have a satisfactory trade-off between accuracy and efficiency.
Figure 5 depicts the output for real and test data, regarding method 1. That figure shows the high prediction accuracy of the model, which is not the best achieved.

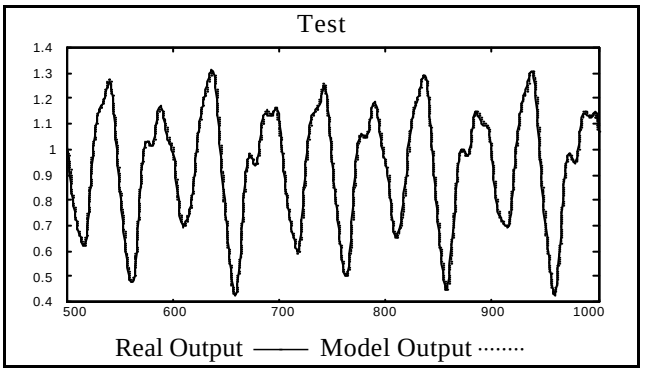

Figure 5. Chaotic series: output prediction in a linguistic model with algebraic operators and two-sided Gaussians.

CONCLUSIONS

In this paper a comparative analysis of different fuzzy structures and parameterizations is performed. By the application of subtractive clustering an initial structure for the fuzzy model is obtained, which is used for the initialization of a fuzzy neural network. Next, the parameters are adjusted through the training of the network, according to the structure defined, i.e., linguistic, zero order or first order Takagi-Sugeno. Algebraic and truncation operators are used, as well as regular and two-sided Gaussian functions. The techniques described are applied to the Mackey-Glass time series, having been concluded that first order models are the most accurate. However, zero order models present the best trade-off between model accuracy and efficiency. In terms of fuzzy operators, algebraic operators lead to more precise models. As for membership functions, it was concluded that the additional complexity of two-sided Gaussians did not originate a significant gain. So, regular Gaussian functions are preferable.

REFERENCES

1. Castro J. L. 1995. “Fuzzy logic controllers are universal approximators”, IEEE Transactions on Systems, Man and Cyberetics, 25 (4), 629-635.

2. Chiu S. L. 1994. “Fuzzy model identification based on cluster estimation”, Journal of Intelligent and Fuzzy Systems, 2 (3), 267-278.

3. Davé R. N. and Krishnapuram R. 1997. “Robust clustering methods: a unified view”, IEEE Transactions on Fuzzy Systems, 5, (2), 270-293.

4. Jang J.-S. R. 1993. “ANFIS: Adaptive Network-based Fuzzy Inference System”, IEEE Transactions on Systems, Man and Cybernetics, 23 (3),665-685.

5. Lin C.- T. 1995. “A neural fuzzy control scheme with structure and parameter learning”, Fuzzy Sets and Systems, 70, 183-212.

6. Mackey M. C. and Glass L. 1977. “Oscillation and chaos in physiological control systems”, Science, 197, 287-289.

7. Nauck D. and Kruse R. 1999. “Neuro-fuzzy systems for function approximation”, Fuzzy Sets and Systems, 101, 261-271.

8. 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).

9. 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.

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, 15 (1), 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, 3 (1), 28-44.

ACKNOWLEDGEMENTS

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