15th Triennial World Congress, Barcelona, Spain

AUTOMATIC FUZZY RULE GENERATION IN A BIOTECHNOLOGICAL PROCESS

M. J. Fuente [*] **G. I. Sainz Palmero** [**] **D. Pindado** [**]

[*] *Department of Systems Engineering and Control. Faculty of Science, University of Valladolid. Prado de la Magdalena s/n, 47011 Valladolid, Spain. E-mail: maria@autom.uva.es*
[**] *ETSII. University of Valladolid. Paseo del Cauce s/n, 47011 Valladolid, Spain. E-mail:gresai@eis.uva.es.*

Abstract This paper deals with the modeling of a biotechnological process from input-output data. A simple fuzzy-neural network is utilized to obtain a non-linear model of a wastewater plant. The fuzzy curves are utilized in order to know the significant input variables, the number of rules to model the system and the set of initial weights in the fuzzy neural network model. From the point of view of neural networks the model has weights to be trained from input output data and from the view point of fuzzy logic the model is a set of rules to explain the behaviour of the system.

Keywords: Fuzzy rules, neurofuzzy systems, modeling, wastewater plant.

1. INTRODUCTION

The modeling of nonlinear systems is useful for model-based control applications, when linear modeling results in unsatisfactory control performance, while nonlinear modeling can significantly improve system control.

The non-linear dynamical processes can be modeled in two basic ways: physical models based on nonlinear differential or difference equations, and black-box models which make use of general functions approximators and data-driven construction techniques. The first approach is only suitable for well understood processes, while the latter one can be used to approximate processes that are difficult to describe in the traditional framework based on prior knowledge. Examples of such systems can be found in various fields including food industries, biotechnology, combustion engines, etc.

The choice of a suitable black-box structure is not simple, as many different possibilities exist. Among them are the multilayered neural networks (Brouwn *et al.*, 1994), (Parlos *et al.*, 1994), the fuzzy logic based model (Takagi and Sugeno, 1985), (Czogala and Pedrycz, 1981) and fuzzy-neural networks models (Jang, 1993), (Lin and Cunningham III, 1995), (Sainz Palmero *et al.*, 2001). This paper concerns with this last type of modeling. A simple neural network is used to implement a fuzzy-ruled-based model with appropriate membership functions of a real system from input-output data.

The use of fuzzy curves concept permits to identify the number of significant input variables, the estimation of the number of rules needed in the fuzzy model and the determination of the initial weights for the neural network. The net is trained with different optimization techniques: an evolutive algorithm, random search techniques (Solis and Wets, 1981) and a modification of the delta rule (Pindado, 2001). It can prove that the best result has been obtained with the delta rule modification. When the network is trained, it is possible to see the system as a fuzzy base knowledge where to gain insight into the plant. The system modeled in this paper is a wastewater treatment plant.

The rest of the paper is organized as follows: section 2 describes the network architecture, section 3 shows the system under study the wastewater treatment plant.

Section 4 uses the fuzzy curves to determine the network structure, section 5 discusses the training algorithm and finally in section 6 are the modeling results.

2. THE FUZZY NEURAL NETWORK

The architecture of the four layer(input, fuzzification, inference and defuzzification) fuzzy-neural network is shown in Figure 1, (Lin and Cunningham III, 1995). This neural network maps each one of the components of a fuzzy system into a layer of the net. The first layer is the input layer that contains N neurons, one for each input variable into the system. The second is the fuzzification layer and it has NxR neurons, where R is the number of rules wished for the system. Each neuron of this layer represents a fuzzy membership function for one of the input variables. The output of this layer is:

$$\mu_{ij}(x_i) = exp(-|w_{ij1}x_i + w_{ij0}|^{l_{ij}}) \qquad (1)$$

where μ_{ij} is the fuzzy membership function of the i^th input variable corresponding to the j^th rule, and l_{ij} is typically in the range $0.5 \le l_{ij} \le 5$ and initially equals to one or two. This parameter permits to modify the membership function of each neuron from nearly a triangular form until a trapezoidal one, as it is shown in Figure 2. The weights between the input and the fuzzification layers are labeled as $W = \{\{w_{ij0}, w_{ij1}\} : i = 1,...,N; j = 1,...,R\}$.

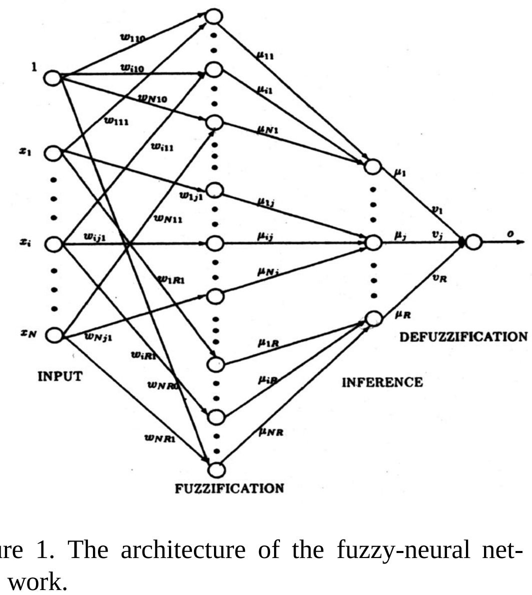

Figure 1. The architecture of the fuzzy-neural network.

The third layer or inference one implements a set of R fuzzy rules $r_j (j = 1,2,...,R)$ of the form: "if x_1 is μ_{1j} and x_2 is μ_{2j} and ... and x_N is μ_{Nj} then y is μ_j". The operator "and" represents a T-norm, and it is implemented as the multiplicative inference, i.e., to use the algebraic product to evaluate the premises of the different rules. The activation function used is $f(net_j) = net_j$, and its output is:

$$\mu_j(x_1, x_2, ...x_N) = \prod_i^N \mu_{ij}(x_i) \qquad (2)$$

The connecting weights between the third layer and the fourth one, called defuzzification layer, are the central values, v_j, of the fuzzy membership functions of the output variable. The weighted sum defuzzification method is used, for this reason the output of this layer is:

$$o(x_1, x_2, ...x_N) = \sum_j^R \mu_j(x_1, x_2, ..., x_N) v_j \qquad (3)$$

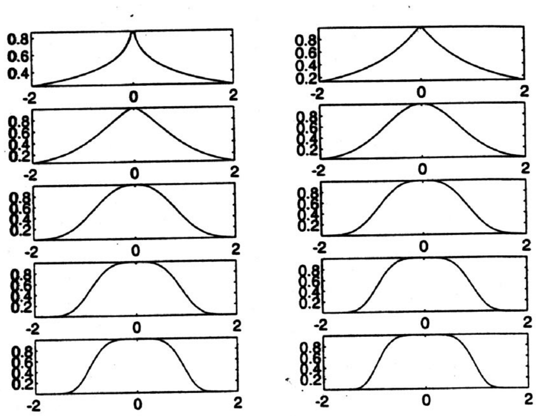

Figure 2. Input fuzzy membership functions generated by $\mu(x) = e^{-|x|^l}, x \in [-2,2], l = 0.5, ..., 5$.

Finally the total output of the net is in the equation (4), and it is possible to show that this equation can represent any continuous function $f : R_N \longrightarrow R$ over a compact set as closely as it is desired (Lin and Cunningham III, 1995).

$$o(x_1, ...x_N) = v_j \sum_j^R \prod_i^N exp(-|w_{ij1}x_i + w_{ij0}|^{l_{ij}}) \qquad (4)$$

3. WASTEWATER TREATMENT PLANT

Wastewater treatment plants are designed to process effluents and return clean water to the river. In most industrialised countries, legislation imposes limits on the quality of water discharged from the plant. Fig. 3 shows a schematic diagram of the plant and its description is given by (Moreno, 1994). Treatment comprises the following steps:

- A physical-chemical procedure that removes sand, oil and suspended solids.
- Aerobic treatment activated sludge process. The effluent and the recycled activated sludge (RAS) are introduced into six tanks where the aerobic action of biomass reduces the organic waste material in the water. The dissolved oxygen required is provided by a set of aeration turbines.

- Clarification. The effluent is fed into clarification tanks, where the activated sludge and clean water are separated. The water is discharged to the river. Part of the settled activated sludge is recycled to the reactors.

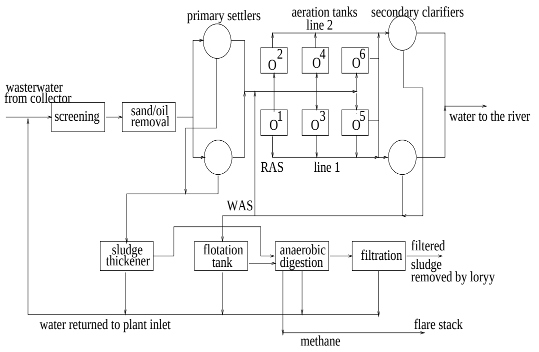

Figure 3. Schematic diagram of the wastewater treatment plant

Of interest are the activated sludge and the clarification processes (Fig. 4), and this part will be modeled with the neuro-fuzzy system. The variables in this process are the organic substrate dissolved in the water, s, and the concentration of microorganisms in the plant, called biomass, x, which will be the outputs of the model. The input variables are the substrate and flow of water at the input of the plant, called s_i and q_i respectively that are measurable disturbances and q_2 the recycled flow considered as the manipulated variable.

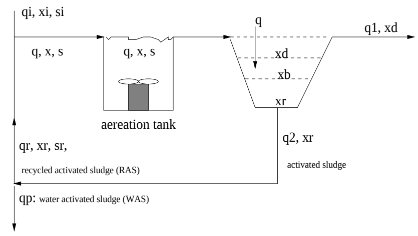

Figure 4. Aeration tank and secondary clarifier scheme.

4. NETWORK STRUCTURE

The network structure, i.e., the number of inputs N, the number of rules R, and the initial values for the weights w_{ij} and v_j, will be determined with a set of input-output data from the plant. For the model of the biomass, x, the system output is denoted $y(t)$, and the variables considered as candidates to inputs to the model are: $x(t-1), x(t-2), \ldots, x(t-6), s(t-1), \ldots, s(t-6), s_i(t), s_i(t-1), \ldots, s_i(t-6), q_i(t), q_i(t-1), \ldots, q_i(t-6), q_2(t), q_2(t-1), \ldots, q_2(t-6)$, i.e., the past inputs and outputs of the process. There are thirty three candidates, denoted in this work as $x_i (i = 1, \ldots, n)$. Although assume that there are m training data points available and $x_{ik} (k = 1, \ldots, m)$ is the i^{th} coordinate of each one of the m training points. In this case as the training set must be a representative selection of all the ways of system operation, training data was generated as a step train on q_i, s_i and q_2 with different amplitudes and frequencies, generating $m = 5000$ data pairs.

In order to simplify the model, the fuzzy curves are used, these curves give an idea of the most significant system variables:

$$c_i(x_i) = \frac{\sum_{k=1}^{m} \phi_{ik}(x_i).y_k}{\sum_{k=1}^{m} \phi_{ik}(x_i)} \tag{5}$$

where ϕ_{ik} is a fuzzy membership function of the variable x_i with the training pattern x_{ik}, such that, if x_i is x_{ik} then ϕ_{ik} is 1, and it is defined as:

$$\phi_{ik}(x_i) = exp(-(\frac{x_{ik} - x_i}{b})^2), k = 1, 2, \ldots, m \tag{6}$$

Fig. 5 shows the fuzzy curves $c_1, c_2, \ldots, c_{33}$, for the candidates to the system input variables. If the fuzzy curve for a given input is flat, then this input has little influence in the output data and it is not a significant input. If the range of a fuzzy curve c_i is about the range of the output data, then the input candidate x_i is important for the output variable. The fuzzy curve tells that the output is changing when x_i is changing, then the importance of the input variable x_i is according to the range covered by their fuzzy curves c_i. As there are so many fuzzy curves, it is not possible to see in the graphic the importance of the different input variables. So, in Table 1 are the fuzzy curves ranges for the input candidates in the biomass model.

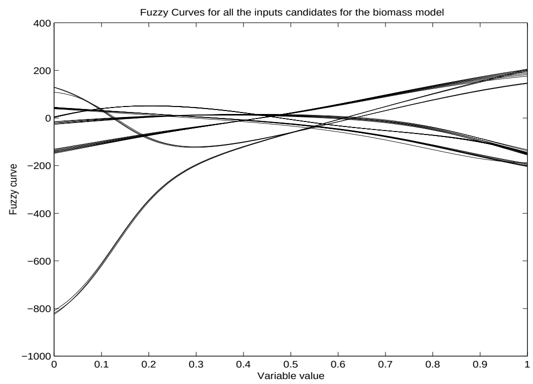

Figure 5. Fuzzy curves for all the input candidates to the biomass model.

As it is possible to see in that table the most significant variables, used as input to the model are $x(t-4), x(t-2), x(t-6)$ because they cover all the output range. But, in order to get the dynamic of the system it is necessary to take into account as inputs, other variables that influence the system output, as the substrate (s_i) and flow (q_i) at the input of the plant. By this reason the most significant of those variables, as it is shown in Table 1, are also chosen as inputs to the system: $s_i(t-6), s_i(t-5), s_i(t-4), q_i(t-2), q_i(t-1)$ and $q_i(t-3)$. The fuzzy curves permit pass from $n = 33$ candidates to only $N = 9$ input variables.

Table 1. Range of fuzzy curves for all inputs candidates for the biomass model.

Variable	%	Variable	%
$x(t-4)$	100.00	$q_i(t-5)$	22.97
$x(t-2)$	99.56	$q_i(t-6)$	22.61
$x(t-6)$	97.96	$q_i(t)$	22.06
$s_i(t-6)$	34.44	$q_2(t)$	19.73
$s_i(t-5)$	33.83	$q_2(t-1)$	19.67
$s_i(t-4)$	33.19	$q_2(t-2)$	19.60
$s_i(t-3)$	32.41	$q_2(t-3)$	19.50
$s_i(t-2)$	31.59	$q_2(t-4)$	19.34
$s_i(t-1)$	30.66	$q_2(t-5)$	19.20
$s_i(t)$	29.73	$q_2(t-6)$	19.15
$x(t-1)$	26.65	$s(t-1)$	18.19
$x(t-3)$	26.16	$s(t-2)$	17.71
$x(t-5)$	26.16	$s(t-3)$	16.92
$q_i(t-2)$	24.29	$s(t-4)$	16.18
$q_i(t-1)$	24.09	$s(t-5)$	15.48
$q_i(t-3)$	23.94	$s(t-6)$	14.81
$q_i(t-4)$	23.51		

The fuzzy curves also are used to estimate the number of rules, R, needed to model the system. Each fuzzy curve c_i of each input will be approximated for a number of rules, R_i, by the maximum and minimum points on the curve, the fuzzy model will interpolate between these points. If the maximum and minimum points are far apart, or the curve is not smooth, it is possible to add more rules. As the system has N inputs, it has N fuzzy curves and will have N number of rules $R_1,...,R_N$, to determine the total number of rules, $R = max(R_1,...,R_N)$. For this case, the fuzzy curves corresponding to the inputs chosen are in Fig.6, and it is possible to see that the number of rules needed to model the system are $R = 2$, because the curves are so smooth and only the maximum and minimum points have been considered.

At this point there are R rules, and N inputs variables, i.e., N is the number of neurons in the input layer, R is the number of neurons in the inference layer, and (NxR) is the number of nodes in the fuzzyfication layer, then the structure of the network has been determined.

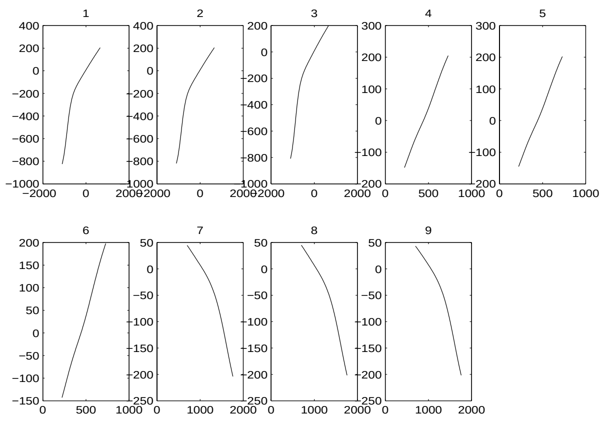

Figure 6. Fuzzy curves for the variables chosen as inputs in the biomass model.

Before to train the network, it is necessary to know the initial value of the weights W and V. The weights V are the centers of output variable fuzzy membership functions, in order to obtained them, the range of the training output data are divided into R intervals, and the initial $v_j (j = 1,2,...,R)$ are the central value of these R intervals in ascending order. In this case the plant output, biomass, varies from 1200 to 2600, and using 2 rules, this range is divided into two intervals with centers at $v_1 = 1550$ and $v_2 = 2250$.

The initial values of the weights (w_{ij1}, w_{ij0}) are calculated in the same way that v_j, i.e., the domain for each input variable is divided into R intervals corresponding to the R intervals in the output space, and the weights are such that the centers of the fuzzy membership functions are the centers of those intervals. $x_{ij} (j = 1,...,R)$ denotes the centers of those intervals, ordered such that x_{iR} corresponds to the interval containing the largest value of the fuzzy curve, x_{iR-1} is the center of the interval which contains the next largest central point on the same fuzzy curve c_i and so on for $j = R-2, R-3,...,1$.

The length of the interval over which a rule applies in the domain of c_i is denoted Δx_i. And the initial fuzzy membership function of x_i over the rule j is defined as $exp((-\mid \frac{(x_{ij}-x_i)}{a\Delta x_i} \mid)^{l_{ij}})$, where a is in the range $[0.5, 2]$. Hence, referring to eq. (1) where is represented the desired fuzzy membership function for this network, the initial weights are $w_{ij1} = -\frac{x_{ij}}{a\Delta x_i}$ and $w_{ij0} = \frac{1}{a\Delta x_i}$.

5. TRAINING THE MODEL

At this point, the neural network has to be trained, three techniques have been probed, the first one has been an evolution strategy, but this algorithm depends too much of the initial value of the weights in the fuzzy neural net. This method converges to a global minimum in a compact set, and it is necessary to define in advance that set, but there is not knowledge about the final values of the weights, i.e., if the compact set it is very restrictive, it is not possible to know if will be another compact set where the weights reach a better solution. Also, the compact set has to be defined in advanced and is different from each system to be modeled, and it is a big difficult in order to automate the training step, and by other hand the time calculus to obtain the solution is very big.

The second method used to train the network has been a random search techniques, the MOAM algorithm (Solis and Wets, 1981), which assures the global convergence in a compact set. The problems with this algorithm are the same as with the evolution one, it is necessary to know in advance the range of the weights in the neural network for each application.

The third method used is a modification of the delta rule. This rule has been defined for feedforward neural networks, where the input to a neuron is a linear combination of the output of the neurons in the layer before , and the neurons activation functions are differentiable. The fuzzy network, used in this paper, does not satisfy these conditions. In order to use the delta rule it is necessary to modify the network, only in the training step, (Pindado, 2001). In the fuzzification layer, the output of the neurons is in eq.(1), where it is possible to see that the activation function depends of a parameter to be calculated during the training step, l_{ij}. Then, this layer is divided into two layers, i.e., a fictitious layer is defined with output function y_1 as:

$$y_1 = log(|w_{ij1}x_i + w_{ij0}|) \tag{7}$$

And the fuzzification layer output will be:

$$\begin{aligned} \mu_{ij}(x_i) &= exp(-|w_{ij1}x_i + w_{ij0}|^{l_{ij}}) = \\ & exp(-exp(log(|w_{ij1}x_i + w_{ij0}|^{l_{ij}}))) = \\ & exp(-exp(l_{ij}y_1)). \end{aligned} \tag{8}$$

In the inference layer, the output of the neurons is in eq. (2), in order to use the delta rule, a modification is made:

$$\begin{aligned} \mu_j &= \prod_i^N exp(-exp(l_{ij}y_1)) = \\ & exp(\sum_{i=1}^N -exp(l_{ij}y_1)) = exp(\sum_{i=1}^N y_2) \end{aligned} \tag{9}$$

With these modifications, now it is possible to use the delta rule for this new network with five layers (Pindado, 2001), which inputs and outputs, y, are expressed in the following equation:

$$\begin{aligned} net1_j &= 1, x_1, ..., x_N \\ net2_{ij} &= w_{ij1} * net1_j + w_{ij0} \\ y2_{ij} &= log(|net2_{ij}|) \\ net3_{ij} &= l_{ij} * y2_{ij} \\ y3_{ij} &= -exp(net3_{ij}) \\ net4_j &= \sum_{i=1}^N y3_{ij} \quad y4_j = exp(net4_j) \\ net5 &= \sum_{j=1}^R v_j * y4_j = y5 \end{aligned} \tag{10}$$

The performance index is defined as:

$$PI = \frac{\sqrt{\sum_{k=1}^M (y_k - {y_k}^d)^2}}{\sum_{k=1}^M |\, {y_k}^d \,|} \tag{11}$$

where ${y_k}^d$, $(k = 1, 2, ..., m)$, are the desired output values and y_k, $(k = 1, 2, ..., m)$, are the outputs from the net. The training is continuing until, for some i, $\sum_{j=1}^{i+100} PI_j - \sum_{j=i+100}^{i+200} PI_j \leq \varepsilon$, or the number of iterations reaches its maximum value, I_{max}, and $\varepsilon > 0$ is a chosen small number. If after training the model performance is not adequate, the number of rules are increased, or the initial weights are changed and the network is retrain. Once the network is trained, the model can be also reduced, (Lin and Cunningham III, 1995), and (Pindado, 2001).

6. WASTEWATER PLANT APPLICATIONS

The results to apply this fuzzy neural network to the wastewater treatment plant can be summarized in this section. For the first plant output, the biomass, the fuzzy curves are in Fig. 5, and the number of inputs and rules considered are $N = 9$ and $R = 2$ respectively as has been explained in Section 4. The network has been trained with the delta rule and with the following parameters: $I_{max} = 5000$, and $\varepsilon = 0.0001$. The training procedure concludes in 2134 iterations with a $PI = 0.006915$. As the system is a dynamical model, in order to know if the neural net has learned the plant performance, the network needs to be validate with different data of the training procedure ones. Fig. 7 represents the model behaviour with an input different to the training data, with an adequate result and a performance index $PI = 0.007343$. The fuzzy membership functions of the two-rules model are plotted in Fig.8.

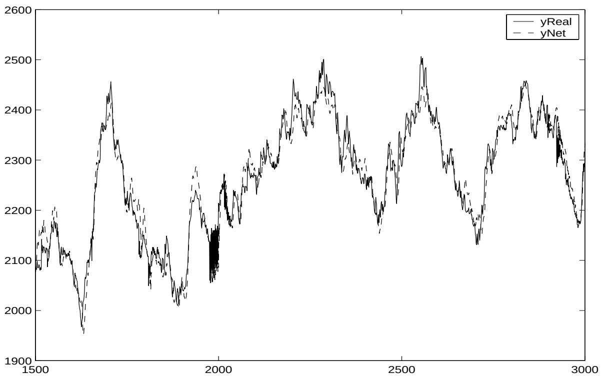

Figure 7. Network and real output for validation data.

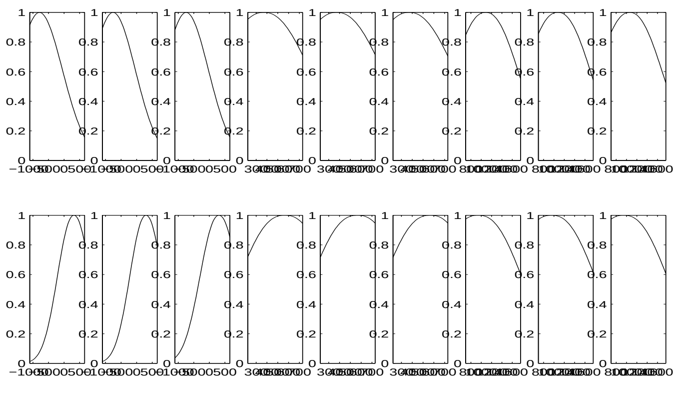

Figure 8. The two-rules model for the biomass.

The second output of the wastewater treatment plant, is the substrate, s. The procedure to calculate this model is the same that the biomass. First, the inputs candidates are, $x(t-1), ..., x(t-6), s(t-1), ..., s(t-6), s_i(t), s_i(t-1), ..., s_i(t-6), q_i(t), q_i(t-1), ..., q_i(t-6), q_2(t), q_2(t-1), ..., q_2(t-6)$, Fig 9 and Table 2, shown the fuzzy curves for the input candidates, and the range covered for each one. As before, the inputs chosen either cover all the output range or influence

Table 2. Range of fuzzy curves for all inputs candidates for the substrate model.

Variable	%	Variable	%
$s(t-1)$	100.00	$s_i(t-4)$	21.65
$s(t-2)$	96.83	$s_i(t-5)$	20.85
$s(t-3)$	93.26	$s_i(t-6)$	20.22
$s(t-4)$	89.89	$q_2(t)$	5.44
$s(t-5)$	86.64	$q_2(t-1)$	5.36
$s(t-6)$	83.49	$q_2(t-2)$	5.32
$q_i(t-1)$	36.47	$q_2(t-3)$	5.29
$q_i(t-2)$	36.29	$q_2(t-4)$	5.26
$q_i(t-3)$	35.40	$q_2(t-5)$	5.23
$q_i(t)$	34.90	$q_2(t-6)$	5.22
$q_i(t-4)$	34.35	$x(t-6)$	3.66
$q_i(t-5)$	33.37	$x(t-4)$	3.35
$q_i(t-6)$	32.72	$x(t-2)$	2.98
$s_i(t-1)$	24.00	$x(t-5)$	2.47
$s_i(t)$	23.63	$x(t-1)$	2.31
$s_i(t-2)$	23.43	$x(t-3)$	2.11
$s_i(t-3)$	22.53		

the behaviour of the plant output, and as it is shown in Table 2 are $s(t-1), s(t-2), s(t-3), s(t-4), q_i(t-1), q_i(t-2), q_i(t-3), s_i(t), s_i(t-1)$ and $s_i(t-2)$. The rule number considered in this case are $R = 3$, and the training procedure concludes in 293 iterations with $PI = 0.008769$. The comparison between the real substrate and the output of the model with data different from the training data is in Fig. 10, obtained with a $PI = 0.009124$.

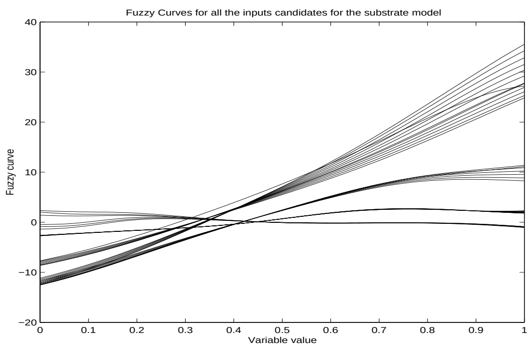

Figure 9. Fuzzy curves for the substrate.

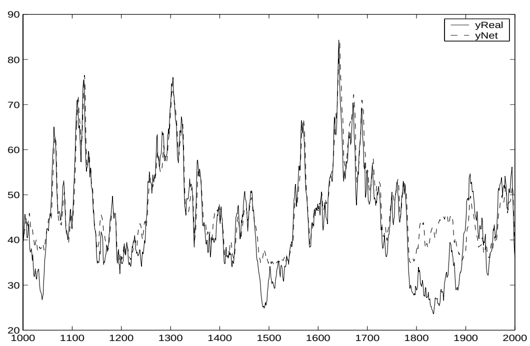

Figure 10. Network and real output for validation data for the substrate.

7. CONCLUSIONS

A simple and effective fuzzy-neural model of complex systems from input-output data has been used to model a biotechnological process. The fuzzy curves are used to identify the input variables, estimated the number of rules needed, and set the initial weights for the fuzzy neural network. The network has been modified in order to use the backpropagation learning algorithm, and observing the learning rules and membership functions for the input variables it is possible to simplify the model. This network has been used to model a complex system, a wastewater treatment plant with adequate results. As possible future work a genetic algorithm will be used to train the neuro-fuzzy system and to use this model in order to design a based-model controller.

8. ACKNOWLEDGEMENTS

The authors are grateful to CICYT (Spain) for support through the project DPI2000-0691-C02-02.

9. REFERENCES

Brouwn, G.G., A.J. Krisjgsman, H.B. Verbruggen and P.M. Bruijin (1994). Single-layer networks for nonlinear system identification. *Engineering Applications of Artifical Intelligence* **7**, 227–243.

Czogala, E. and W. Pedrycz (1981). On identification in fuzzy systems and its application in control problems. *Fuzzy Sets and Systems* **6**, 73–83.

Jang, J.R. (1993). ANFIS: Adaptative-network-based fuzzy inference system. *IEEE Trans. on Systems, Man and Cybernetics* **23**, 665–685.

Lin, Y. and G.A. Cunningham III (1995). A new approach to fuzzy-neural system modeling. *IEEE Trans. on Fuzzy Systems* **3**, 190–197.

Moreno, R. (1994). Estimación de Estados Y Control Predictivodel Proceso de Fangos Activados. PhD thesis. Universidad Autonoma de Barcelona (Spain).

Parlos, A.G., K.T. Chong and A.F. Atiya (1994). Application of the recurrent multilayer perceptron in modeling complex process dynamics. *IEEE Trans. on Neural Networks* **5**, 255–266.

Pindado, D. (2001). Generación automática de una red neurodifusa para el modelado de una planta de depuración de aguas residuales. Technical report. Valladolid University.

Sainz Palmero, G. I., M.J. Fuente and P. Vega (2001). Recurrent neuro-fuzzy modeling of a biotechnological process. In: *European Control Conference 2001 (ECC'2001)*. Oporto, Portugal. pp. 3822–3827.

Solis, T.J. and R.J. Wets (1981). Minimization by random search techniques. *Mathematics of Operation Research* **6**, 19–30.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on System, Man and Cybernetics* **15**, 116–132.