

“APPLYING SUBTRACTIVE CLUSTERING FOR NEURO-FUZZY MODELLING OF A BLEACHING PLANT”

R. P. Paiva*, A. Dourado*, B. Duarte§

**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. +351-39-790000, Fax: +351-39-701266, e-mail: ruipedro, dourado@dei.uc.pt

§*Companhia de Celulose do Caima, S A,*
P 2250 Constância, Portugal

Keywords: pulp bleaching, clustering, subtractive clustering, neuro-fuzzy modelling

Abstract

Presently, the demands for good paper quality are growing higher and higher. Since one important variable to assess paper quality is paper brightness, pulp bleaching is a most important concern. Therefore, it is extremely important to have a thorough understanding of the bleaching plant, in order to achieve those high standards. In this paper a neuro-fuzzy approach is proposed for modelling of the pulp bleaching plant at Companhia de Celulose do Caima, S.A. (Portugal). This strategy is conducted in two phase: in the first one, subtractive clustering is applied in order to extract a set of fuzzy rules; then, in the second stage, the centres and widths of the membership functions are tuned by means of a fuzzy neural network trained with backpropagation. This technique seems promising since it permits good results with large nonlinear plants. Furthermore, it describes the plant using a set of linguistic rules, which have the advantage of being closer to natural human language, so, more intuitive for operators. The results obtained so far can be acceptable, since the model root mean square error is about 0.2% of the real value.

1 Introduction

Pulp bleaching is a strongly nonlinear process for which many physical phenomena are only superficially understood. Furthermore, the bleaching sequence is influenced by a large set of variables for which relative importance is not so well comprehended. Therefore, finding a good model of the plant is a key point, yet not a trivial task.

In the absence of an accurate first-principles model, which is also very difficult to come up with, it is necessary to resort to Artificial Intelligent techniques, which permit to describe the plant based on experts' intuition, as well as on operating data. From the set of techniques available, fuzzy modelling appears to be the most adequate approach, since it

contributes to interpretability by allowing a system to be represented by linguistic rules. However, if process measurements and some insight of the operators are the only available knowledge, the building up of the fuzzy system is not a simple task. Clustering - in the input-output space - allows one to come up with a finite set of linguistic rules. Then, a fuzzy neural network, trained with backpropagation, adjusts the centres and widths of the membership functions.

One important characteristic of the process is its time-varying transport delay. The effects of this phenomenon will be presented and discussed.

This paper is organised in five sections. Section 2 describes the two-phase algorithm for fuzzy modelling. Section 3 gives a short description of the pulp bleaching plant. In section 4 the application of the referred techniques to a nonlinear system is described. Simulation results are presented. Section 5 is devoted to the description of a possible strategy for dealing with the variable time delays. Finally, Section 6 concludes the paper pointing out the advantages and limitations of the strategy used and the main problems encountered, as well as some directions for future work, and a possible strategy to include the variable time delays in the model.

2 Fuzzy modelling

The goal of fuzzy modelling is to obtain a set of fuzzy rules that describe the dynamics of a plant, given a set of operating data and, perhaps, an initial set of linguistic rules dictated by experts. In this work, a Fuzzy Auto-Regressive with Exogenous (FARX) variable structure is followed to model the bleaching plant. In that structure, the system is described, according to Mamdani inference, by a set of rules of type (1):

$$R_i : \text{If } y_1(k) \text{ is } A_{1i} \text{ and } \dots \text{ and } y_q(k - n_q) \text{ is } A_{qm_i} \\ \text{and } \dots u_1(k - d_1) \text{ is } B_{1i} \text{ and } \dots \text{ and } u_p(k - m_p - d_p + 1) \text{ is } B_{pm_i} \\ \text{then } y_1(k + 1) \text{ is } C_{1i} \text{ and } \dots \text{ and } y_q(k + 1) \text{ is } C_{qi} \quad (1)$$

where q stands for number of system outputs, p represents the number of inputs and the parameters $n_1, \dots, n_q, m_1, \dots, m_p, d_1,$

..., d_p are related to the system order and discrete pure time delay. A_{jki} , B_{jki} and C_{ji} are the linguistic values for each output and input variables, defined by their membership functions: $\mu_{A_{jki}}, \mu_{B_{jki}}, \mu_{C_{ji}}, i = 1, 2, \dots, R$.

For the sake of satisfactory identification, the parameters $n_1, \dots, n_q, m_1, \dots, m_p, d_1, \dots, d_p$ must be properly chosen. This is accomplished on the basis of prior knowledge or by comparison of different values in terms of some criteria. Assuming this problem is solved, the issue is: 1) to obtain a set of rules of type (1); 2) to adjust the parameters of the membership functions using data collected from the system:

$$X = [\theta(1) \ \dots \ \theta(N-1)]^T, \Psi = [Y(1) \ \dots \ Y(N-1)]^T \quad (2)$$

where N is the number of data samples available for the identification purpose and \mathbf{q} is the regression vector.

2.1 Clustering

In order to obtain a set of R rules avoiding the problems inherent to grid partitioning, e. g., rule base explosion, subtractive clustering is applied [Chiu94]. This technique, as well as some other fuzzy clustering methods, is employed since it allows a scatter input-output space partitioning.

Subtractive clustering is, essentially, a modified form of the Mountain Method. Thus, let Z be the data set obtained by concatenation of the sets X and Y (2). Assuming that all the data points are normalised in each dimension, the data set Z is bounded by a hypercube. In the algorithm, each point is seen as a potential cluster centre, for which some measure of potential is assigned (3):

$$P_i = \sum_{j=1}^N e^{-\alpha \|z_i - z_j\|^2} \quad (3)$$

where $\alpha = 4/r_a^2$ and $r_a > 0$ defines the neighbourhood radius for each cluster centre. Therefore, the potential associated to each cluster depends on its distance to all of the points, leading to clusters with high potential where the neighbourhoods are dense.

After computing the potential for each point, the one with higher potential is selected as the first cluster centre. Let z_1^* be the centre of the first group and P_1^* its potential. Then, the potential for each point z_i^* is reduced, especially for the points closer to the centre of the cluster:

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

where $b = 4/r_b^2$ and $r_b > 0$ represents the radius of the neighbourhood for which significant potential reduction will occur. The radius for reduction of potential should be to some extent higher than the neighbourhood radius to avoid closely spaced clusters. Typically, $r_b = 1.5r_a$. Since the points closer to the cluster centre will have their potential strongly reduced, the probability for those points to be chosen as the next

cluster is lower. This procedure (selecting centres and reducing potential) is carried out iteratively, until the stopping criteria is reached:

```

If  $P_k^* > e^{up} P_1^*$ 
  Accept  $z_k^*$  as the next cluster centre and continue
Otherwise,
  If  $P_k^* < e^{down} P_1^*$ 
    Reject  $z_k^*$  and finish the algorithm.
  Otherwise
    Let  $d_{min}$  be the shortest distance between  $z_k^*$  and all
    the centres already found
    If  $d_{min}/r_a + P_k^*/P_1^* \geq 1$ 
      Accept  $z_k^*$  as the next cluster centre and continue
    Otherwise
      Reject  $z_k^*$  and assign it the potential 0.0.
      Select the point with higher potential as new  $z_k^*$ .
      Repeat the test.
  End If
End If
End If

```

There, e^{up} specifies a threshold above which the point is selected as a centre, without any doubts and e^{down} specifies the threshold below which the point is definitely rejected. The third case is where the point is characterised by a good trade-off between having a sufficiently high potential and being distant enough from the clusters determined before. Typically, $e^{up} = 0.5$ and $e^{down} = 0.15$.

By the end of clustering, a set of fuzzy rules will have been obtained. Each cluster will represent a rule. However, since the clustering procedure is conducted in a multidimensional space, fuzzy sets must be obtained. As each axis of the multidimensional space refers to a variable, the centres of the membership functions for that variable are obtained by projecting the centre of each cluster in the corresponding axis. As for the widths, they are obtained on the basis of the neighbourhood radius, r_a , defined while performing subtractive clustering. Since gaussian membership functions are used, their standard deviations are computed by:

$$\sigma_{ij} = r_a \cdot \frac{\max(z_{kj}) - \min(z_{kj})}{\sqrt{8}}, k = 1, \dots, N \quad (5)$$

2.2 Fuzzy neural network

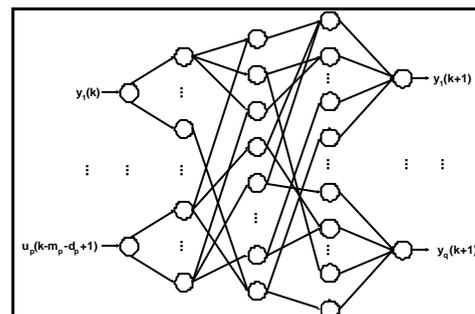


Figure 1: Structure of the fuzzy neural net

After deriving an initial fuzzy inference system based on fuzzy clustering, its parameters, i.e., the centres and widths of membership functions must be optimised. In this paper, this is accomplished by means of training a fuzzy neural network (FNN) using standard backpropagation.

The structure of the FNN is presented in Figure 1. This structure can be found in [Lin95]. There, gaussian membership functions are used. In the present work, two-sided membership functions are used, in order to allow more flexibility. The fuzzy neural network consists of five layers, which are described as follows.

Layer 1 contains the input nodes, which represent input linguistic variables. This layer simply passes the inputs to layer 2.

The nodes in layer 2 are the linguistic terms of each input variable, represented by gaussian membership functions. This layer is responsible for the fuzzification of the crisp input values (6)

$$a_j^{(2)} = \begin{cases} e^{-\frac{(x_i - c_{ijL})^2}{s_{ijL}^2}} & , x_i < c_{ijL} \\ 1 & , c_{ijL} \leq x_i \leq c_{ijR} \\ e^{-\frac{(x_i - c_{ijR})^2}{s_{ijR}^2}} & , x_i > c_{ijR} \end{cases} \quad (6)$$

where $a_j^{(2)}$ denotes the activation for each node on the second layer, c_{ijL} and c_{ijR} stand for the left and right centres of a two-sided gaussian, s_{ijL} and s_{ijR} refer to the left and right standard deviations and x_i represents the i th input. In the following, the superscript will always stand for the layer number.

In layer 3, each node is assigned to a rule of the fuzzy inference system. The antecedents of each rule are defined by setting proper links from nodes at layer 2 to nodes at layer 3. This layer fires each rule based on some fuzzy AND operation. In this work, the truncation operator *min* was used. Normally, an algebraic operator, like *product*, should be used in order to apply the gradient for training the neural network. The two approaches were tested and, since better results were obtained with the operator *min*, despite not being continuous, the operator referred was selected. The output of the third layer is as follows (7).

$$a_k^{(3)} = \min(a_{j_1}^{(2)}, a_{j_2}^{(2)}, \dots, a_{j_n}^{(2)}) \quad (7)$$

Since there are some rules that share the same consequent, layer 4 integrates those rules using some fuzzy OR operation. The nodes at layer 4 define the linguistic terms for each output, represented by gaussian membership functions, as in layer 2. For the same reason as in layer 3, a truncation operator, namely *max*, was used:

$$a_l^{(4)} = \max(a_{k_1}^{(3)}, a_{k_2}^{(3)}, \dots, a_{k_n}^{(3)}) \quad (8)$$

Layer 5 is the output layer. The role of this layer is to perform defuzzification, i.e., convert fuzzy numbers into crisp numbers. In this work, an adaptation of the centre of area defuzzification method is used, in order to cope with two-sided gaussian functions and to incorporate the effect of the widths into the defuzzification strategy (eq. 9).

$$a_m^{(5)} = \frac{\sum_l \frac{1}{2} (c_{lmL} s_{lmL} + c_{lmR} s_{lmR}) a_l^{(4)}}{\sum_l \frac{1}{2} (s_{lmL} + s_{lmR}) a_l^{(4)}} \quad (9)$$

As in layer 2, c_{lmL} , s_{lmL} , c_{lmR} , and s_{lmR} represent the left and right parameters of the two-sided gaussian membership function.

As stated before, the objective of the presented FNN is to perform optimisation of the centres and widths of the gaussian membership functions. For that matter, supervised learning is carried out based on acquired data (2), using standard backpropagation. The goal is to minimise the error function:

$$E_m = \frac{1}{2} [y_m - a_m^{(5)}]^2 \quad (10)$$

where y_m stands for the desired network output and $a_m^{(5)}$ is the actual network output for the m -th output. Assuming that w_{ij} is the parameter to adjust, the general learning rule is as (11):

$$\Delta w_{ij} = lr \cdot \left(-\frac{\partial E_m}{\partial w_{ij}} \right) \quad (11)$$

where lr is the learning rate.

Based on equations (10) and (11), the expressions for adapting the centres and widths of the membership functions are presented below.

Layer 5

In this layer, the centres and widths of the output membership functions are updated. This is conducted by (12), (13) and (14).

$$d_m^{(5)} = y_m - a_m^{(5)} \quad (12)$$

$$\frac{\partial E_m}{\partial c_{lmL}} = -d_m^{(5)} \cdot \frac{s_{lmL} a_l^{(4)}}{\sum_r (s_{rmL} + s_{rmR}) a_r^{(4)}} \quad (13)$$

$$\frac{\partial E_m^{(5)}}{\partial s_{lmL}} = -d_m^{(5)} \cdot \frac{c_{lmL} a_l^{(4)} \sum_r (s_{rmL} + s_{rmR}) - a_l^{(4)} \sum_r (c_{rmL} s_{rmL} + c_{rmR} s_{rmR})}{\left[\sum_r (s_{rmL} + s_{rmR}) \right]^2} \quad (14)$$

Layer 4

In this layer, there are no parameters to update. Therefore, only the error signals (d) need to be computed for backpropagation:

$$d_l^{(4)} = \sum_m d_m^{(5)} \frac{(c_{lmL} s_{lmL} + c_{lmR} s_{lmR}) \sum_r (s_{rmL} + s_{rmR}) - (s_{lmL} + s_{lmR}) \alpha_l^{(4)} \sum_r (c_{rmL} s_{rmL} + c_{rmR} s_{rmR})}{\left[\sum_r (s_{rmL} + s_{rmR}) \right]^2} \quad (15)$$

Layer 3

As in layer 4 only the error signals need to be computed:

$$d_k^{(3)} = \sum_l d_l^{(4)} \frac{\partial a_l^{(4)}}{\partial a_k^{(3)}} \quad (16)$$

Layer 2

In layer 2, the centres and widths of the input membership functions are updated according to eqs. (17), (18), (19), (20) and (21).

$$\frac{\partial E_m^{(1)}}{\partial c_{ij}} = \left(\sum_k d_k^{(3)} \frac{\partial a_k^{(3)}}{\partial a_j^{(2)}} \right) \frac{\partial a_j^{(2)}}{\partial c_{ij}} \quad (17)$$

$$\frac{\partial E_m^{(1)}}{\partial s_{ij}} = \left(\sum_k d_k^{(3)} \frac{\partial a_k^{(3)}}{\partial a_j^{(2)}} \right) \frac{\partial a_j^{(2)}}{\partial s_{ij}} \quad (18)$$

$$\frac{\partial a_j^{(2)}}{\partial c_{ijL}} = \frac{2(x_i - c_{ijL})}{s_{ijL}^2} e^{-\frac{(x_i - c_{ijL})^2}{s_{ijL}^2}} \quad (19)$$

$$\frac{\partial a_j^{(2)}}{\partial s_{ijL}} = \frac{2(x_i - c_{ijL})^2}{s_{ijL}^3} e^{-\frac{(x_i - c_{ijL})^2}{s_{ijL}^2}} \quad (20)$$

$$\frac{\partial a_k^{(p3)}}{\partial a_j^{(p2)}} = \begin{cases} 1 & , a_k^{(p3)} = a_j^{(p2)} \\ 0 & , a_k^{(p3)} \neq a_j^{(p2)} \end{cases} \quad (21)$$

3 Pulp bleaching plant

The main goal of bleaching is to decolourise the lignin present in wood fibers. In order to achieve this goal, chemicals are added, which react with the unbleached chromophores producing the desired bleached chromophores so that pulp properties can satisfy the standards demanded by paper industry. A major concern is to obtain satisfactory outlet brightness.

Chlorine is known to be a universal bleaching agent. However, due to ambient restrictions, pulp bleaching must be carried out without any compound containing chlorine. This

has led to a Totally-Chlorine Free (TCF) bleaching sequence. Some TCF sequences have been used in the past years. In our case an EOP/P sequence is conducted, as presented in Figure 2 [Caima94].

3.1 Description of the bleaching sequence

After cooking the wood with acid for delignification, washing and screening the pulp, the bleaching stage is ready to begin. First of all, the pulp is washed in washers 1 and 2. Then, in the EOP (**E**xtraction with **NAOH**, **O**xygen and **H**ydrogen **P**eroxide) stage the pulp is mixed with chemicals, namely hydrogen peroxide, oxygen (bleaching agents), caustic soda (to adjust the pH of the reacting mixture), and sodium silicate (peroxide stabiliser). This mixture reacts within towers 1 and 2 for approximately 4 hours. Before the P (extraction with Hydrogen Peroxide) stage, the pulp is washed in washer 3 in order to recover chemicals and energy. In the P stage the same chemicals as before, except for oxygen are added. The reaction takes place in tower 3 for approximately 2 hours. After this residence time, the pulp is washed in washer 4 and is then conducted to the drying section where it stays for about 1 hour. The total bleaching time, from washers 1 and 2 until dried pulp is obtained takes about 8 hours.

The final bleaching quality is influenced by a great deal of variables. According to experts' knowledge the variables that have a stronger influence on the final pulp quality are inlet brightness, inlet pulp flow, inlet permanganate number (which is a lignin concentration measurement), hydrogen peroxide in both of the stages and inlet pulp flow.

3.2 Brightness analysis

There are a few high-level rules that give some insight on the final brightness achieved: it increases with peroxide flow; it increases with pH; it increases with the consistency; it increases with temperature, until some threshold; it increases with inlet pulp; it decreases with inlet permanganate number. This information can be compared with the set of linguistic rules obtained by the fuzzy inference system.

In [Duarte95], information on the delay times relating each input variable and the outlet brightness is presented. There, it is said that the delay time from inlet brightness to outlet brightness is 7-8 hours, which corresponds to the bleaching time referred above. For inlet pulp flow and inlet permanganate number the delay time should be the same. Concerning the peroxide flow in the P stage, the effect of a change on it affects outlet brightness from 3 to 5 hours later. For the peroxide flow in the EOP stage, the delay time should correspond to time elapsed since inlet pulp is washed in washers 1 and 2. So, a delay time of 6.5-7.5 hours is assumed.

4 Simulation results

The presented techniques are applied to modelling of the pulp bleaching plant described above.

Some of the measured variables are not sufficiently exciting. Thus, their contribution for the achieved bleaching quality is not easily assessed only with measurements.

Moreover, according to the experts' experience, the most important input variables are peroxide flow, inlet brightness

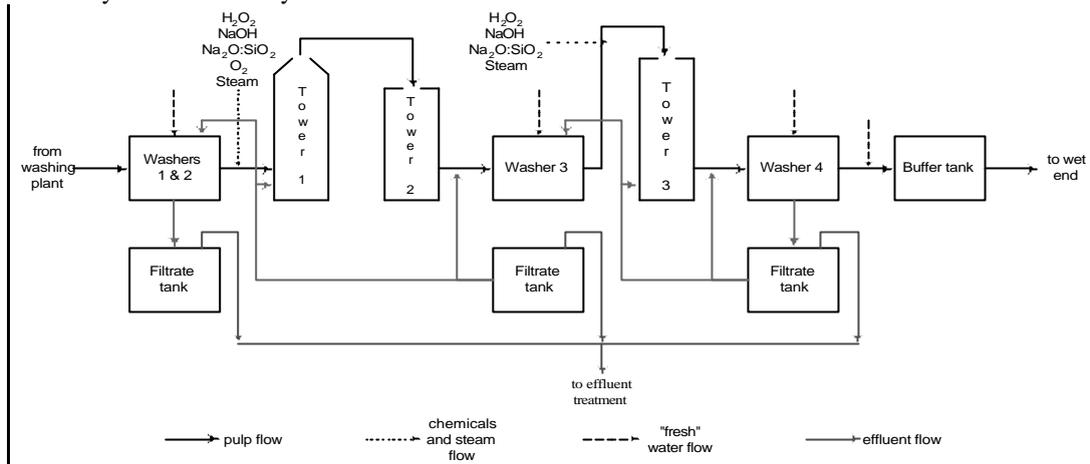


Figure 2: The EOP/P sequence

and pH. Therefore, these are the input variables used to model the plant. Some experiments were carried out with the full set of variables. However, the inclusion of those variables did not bring any better results (actually, some cases happened to worsen the model).

The fuzzy inference system is obtained from the input-output measurements using subtractive clustering and tuning the membership functions with the algorithm in section 2.2. The sampling interval was defined in the mill as one hour; this sampling interval seems to be sufficient since the system's dynamics are very slow. Simulations were carried out with $N=976$ training samples. The parameter r_a for subtractive clustering was defined with the value $r_a=0.35$, leading to $R=53$ rules. Figure 3 presents the training results and Figure 4 shows model validation. There, the continuous line represents real process data, whereas the dashed line represents the model output. The variable depicted is the final brightness achieved, as described in section 3.1.

For the training data, the root mean square error (rmse) was equal to 0.165. However, for the validation data the rms error is slightly higher: $rmse = 0.254$. We can, therefore, conclude that the model obtained does not have satisfactory generalisation capabilities. Some possible reasons for that are noise in measurements, inadequate sampling intervals or inconsistent training and validation sets, resulting from the variable time delay of the system. As stated above, the total pulp residence time varies from 7 to 10 hours (depending on the inlet pulp flow), according to the experts. The described technique seems not to be able to satisfactorily cope with this situation. Thus, a strategy for capturing the effect of the variable time delay is needed. The inlet pulp flow and the levels in the towers mainly influence this delay. Consequently, it is the authors' opinion that including those variables in the model would make it possible to capture the transport delays. Including those variables and extra input variable regression would give the neuro-fuzzy system enough information to find a proper structure, i.e., select the right past input from the regression set, based on measurements for the levels and inlet pulp. However, as can be seen, this scheme was not

completely successful, perhaps due to the absence of measurements for the level in tower 1. Yet, if one has present the difficulties and uncertainties related to the pulp bleaching plant and the (strong) presence of noise in the industrial environment, an rms error of 0.254, can be found satisfactory.

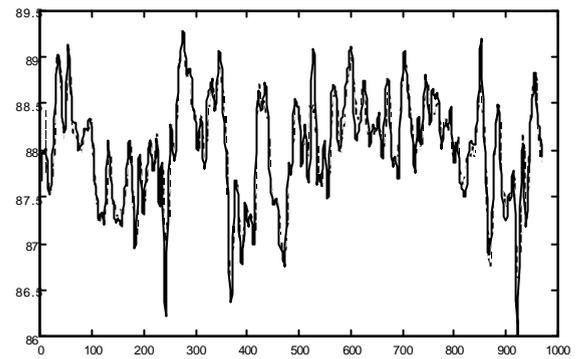


Figure 3. Training results: — Real Process Data
--- FNN output

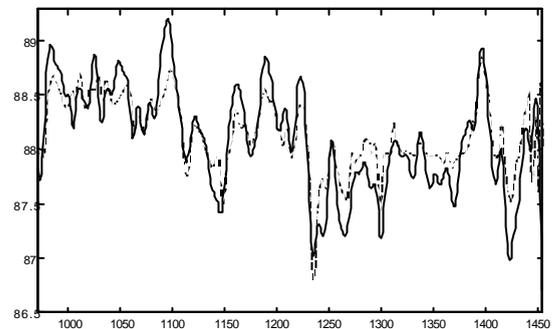


Figure 4. Validation results: — Real Process Data
--- FNN output

6 Conclusions

In this paper, a neuro-fuzzy modelling scheme was presented. The model is obtained in two-phases. In the first one, subtractive clustering was applied in order to obtain a set of fuzzy rules. Then, a fuzzy neural network is trained to optimally tune the membership parameters using backpropagation.

This approach is intended to model a pulp bleaching plant. However, some problems were encountered, which limited the accuracy of the obtained model. Those problems seem to come, fundamentally, from the variable time delay of the system, which is not accurately captured by the model.

Another important factor to take into account is related to the fact that the system may be time varying. To cope with this, an online learning scheme is presently under investigation.

References

- [Chiu94] Chiu, S. "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems*, Vol. 2, No. 3, (1994).
- [Caima94] Caima's Work Group on Best. "TCF bleaching at Caima: past and future", *Companhia de Celulose do Caima*, Internal Report (1994).
- [Duarte95] Duarte, B. "Bleaching plant fuzzy model", *Companhia de Celulose do Caima*, Internal Report (1995).
- [Lin95] Lin, C.-T. "A neural fuzzy control scheme with structure and parameter learning", *Fuzzy Sets and Systems*, No. 70, 183-212, (1995).

Acknowledgements

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