



Quality prediction in pulp bleaching: application of a neuro-fuzzy system

Rui Pedro Paiva^a, António Dourado^{a,*}, Belmiro Duarte^b

^a *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*

^b *Companhia de Celulose do Caima, SA, P 2250 Constância, Portugal*

Received 31 October 2002; accepted 2 July 2003

Abstract

In chemical industries, as paper pulp, quality control is a decisive task for competitiveness. Bleaching is a determinant operation in the quality of white pulp for paper. Quality prediction is decisive in quality control. However, the complexity of the bleaching process (and in general of industrial processes), its nonlinear and time-varying characteristics does not allow to develop reliable prediction models based on first principles. New tools issued from fuzzy systems and neural networks are being developed to overcome these difficulties. In this paper a neuro-fuzzy strategy is proposed to predict bleaching quality by predicting the outlet brightness. Firstly, a fuzzy subtractive clustering technique is applied to extract a set of fuzzy rules; secondly, the centers 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 can be a basis for interpretable models, more intuitive for operators.

© 2003 Published by Elsevier Ltd.

Keywords: Pulp industry; Fuzzy modeling; Neural-networks modeling

1. Introduction

Pulp and paper industry is actually subjected to a very high demand concerning quality. Pulp bleaching, one of its sub-processes, is a nonlinear process for which there are still poorly understood phenomena. It is a sequential process influenced by a large number of variables for which their individual influences are not well known. First principles models with a sufficient level of accuracy do not exist. This leads to the development of other kind of approaches, such as fuzzy systems (Yager, 1995), neural networks and neuro-fuzzy systems (Mitra & Hayashi, 2000; Shi & Mizumoto, 2001). Fuzzy systems are general approximators, capable of accurately representing nonlinear processes and including a priori existent knowledge about the process (Pedrycz & Waletzky, 1997). Fuzzy systems can also be developed

from plant data, but in this case techniques for establishing and tuning the fuzzy rules are needed. One way to deal with these needs is through fuzzy clustering (Delgado, Gomez, Karmeta, & Martin, 1997; Mitra, Pal, & Mitra, 2002). However, one must deal with the problem with the *curse of dimensionality* of the rule base. Several authors have faced this problem (see for example, Wang and Rong, 1999; Runkler, 1998; Duch & Adamczak, 2001). One important characteristic of the bleaching process is its time-varying transport delay, which is dealt with in this work.

The strategy is developed in two phases: first, subtractive clustering is applied to extract a set of fuzzy rules; then, the membership functions (centers and widths) are learned by a neural network trained by backpropagation. The obtained results are promising, showing the potential of the technique. Furthermore, it describes the plant by a set of linguistic (fuzzy) rules close to the natural human language and intuitive for the operators.

This paper is organized as follows: Section 2 describes briefly the pulp bleaching plant. Section 3 describes the

*Corresponding author. Tel.: +351-239-790000; fax: +351-239-701266.

E-mail addresses: ruipedro@dei.uc.pt (R.P. Paiva), dourado@dei.uc.pt (A. Dourado).

two-phase algorithm for fuzzy prediction. Section 4 is devoted to the description of a possible strategy for dealing with the variable time delays. Finally, Section 5 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. Pulp bleaching plant

Bleaching is done to decolorize the lignin present in wood fibers. 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.

The present plant uses 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 Fig. 1.

2.1. The process

The cooking section of Companhia de Celulose do Caima, S.A. is based on a sulphite process, with the pulp produced achieving higher optical and bleachability standards at expenses of smaller physical properties as the viscosity. Such a trade-off is taken into account in designing the bleaching sequence, which is simple than typical sulphate process based mills.

After cooking the wood with acid for delignification, washing and screening the pulp, the bleaching stage is ready to start. The bleaching sequence comprises two

TCF stages, with the first one occurring in two towers sequentially connected and the second one taking place in a third tower. In the first stage, the EOP stage (Extraction with NaOH , Oxygen and Hydrogen Peroxide), the chemical agents added are: sodium hydroxide, to guarantee continuous higher values of pH and dissolve lignin, oxygen (O) used for chromophores oxidation, and hydrogen peroxide (P), with the aim to selectively cut the lignin chains. In the second stage, the P stage, only hydrogen peroxide is used. Sodium hydroxide promotes alkaline extraction, being generally designated as E, thus leading to a bleaching sequence based on two sequential EOP-P stages. The retention times are about 240 min in stage 1 (EOP) and around 150 min in stage 2 (P).

The bleaching reactions (oxidation of chromophores and splitting of lignin chains) are carried out inside the towers, with the chemical agents added in mixers located before each stage. To improve the bleaching efficiency the temperature of pulp is raised at the entry, and all processes occur at about 80°C . To displace ionic charges accumulated in pulp, it passes through washers located after each of the stages. In the second stage, only hydrogen peroxide is used. Sodium hydroxide promotes alkaline extraction, being generally designated as E, thus leading to a bleaching sequence based on two sequential EOP-P stages. The retention times are about 240 min in stage 1 (EOP) and around 150 min in stage 2 (P).

After the P stage, the pulp is washed in washer 4 and is then conducted to the drying section, where it stays for about 1 h. The total bleaching time, from washers 1 and 2 until dried pulp is obtained takes about 8 h.

The most important quality variable of the final product is brightness, and the bleaching section has prominence in achieving higher-quality standards

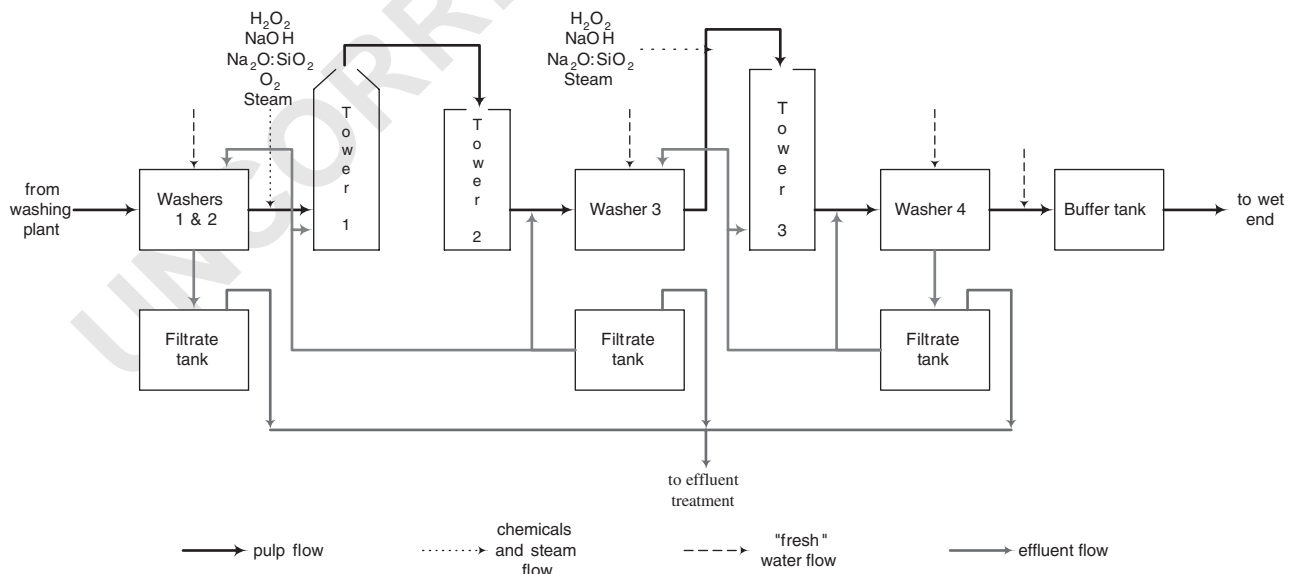


Fig. 1. The EOP/P sequence.

required by customers. Normal disturbances are inlet brightness, inlet pulp consistency and lignin contents, with the variables used to control the plant being the flows of hydrogen peroxide fed to both stages, and flow of sodium hydrogen fed to the first. In normal operation mode, the inlet brightness is around 55°ISO, at the end of first stage is within the range 80–84°ISO, and increases between 4°ISO and 5°ISO in the last tower.

2.2. Brightness analysis for quality classification

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

The time delay from inlet brightness to outlet brightness is 7–8 h, 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 h 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 time delay of 6.5–7.5 h is assumed.

3. The FARX predictor structure and subtractive clustering

Given a set of operating data, a fuzzy auto-regressive with exogenous variable (FARX) structure is derived, in order to model the bleaching plant. The system is described by a set of R fuzzy rules of Mamdani (1977) inference type (1):

R_i :

if $y_1(k)$ is A_{11i} and \dots and $y_q(k - n_q)$ is A_{qn_qi}
and $u_1(k - d_1)$ is B_{11i} and \dots and

$u_p(k - m_p - d_p + 1)$ is B_{pm_pi}

then $\hat{y}_1(k + 1)$ is C_{1i} and \dots and $\hat{y}_q(k - n_q + 1)$ is C_{qi} , (1)

where q denotes the number of system outputs, p represents the number of inputs, and the parameters $n_1 \dots n_q, m_1 \dots m_p, d_1 \dots d_p$ are related to the system order and discrete pure time delay. A_{jki}, B_{jki} and C_{ji} are the fuzzy sets (linguistic values) for each output and input variables, defined by their membership functions: $\mu_{A_{jki}}, \mu_{B_{jki}}, \mu_{C_{ji}}, i = 1, 2 \dots R$. Still in (1), \hat{y}_q denotes the predicted system output for q th output variable.

The parameters $n_1 \dots n_q, m_1 \dots m_p, d_1 \dots d_p$ are properly chosen on the basis of prior knowledge or by comparison of different values in terms of some criteria.

Assuming this problem is already treated, the issue is: (i) to obtain a set of rules of type (1); (ii) to adjust the parameters of the membership functions using data collected from

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

where N is the number of data samples available for the identification purpose and θ is the regression vector.

3.1. Subtractive clustering

In order to obtain a set of g fuzzy conditional rules capable of representing the system under study, clustering algorithms are particularly suited, since they 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 explosion of the rule base, a problem known as the “curse of dimensionality”. Some researchers use grid-based partitioning methods, combined with network pruning. However, based on previous work (Paiva, 1999), it is our opinion that the results are not as good as the ones obtained from clustering techniques, for the following reasons: rule-based explosion is avoided in clustering; in grid-partitioning methods with network pruning, the wrong nodes may be deleted if the network is not optimized; however, optimization of a large dimension network is very time consuming; finally, the network must be reoptimized after the deletion of nodes.

In this paper, Chiu’s subtractive clustering is applied (Chiu, 1994). This scheme possesses some interesting advantages, especially in a neuro-fuzzy identification context. In fact, subtractive clustering is an efficient algorithm, which does not require optimization, being for this reason a good choice for the initialization of neuro-fuzzy networks. Fuzzy c -means and other optimization-based clustering techniques would lead to excessive computer work because they perform an unnecessary optimization phase prior to network training. Also, progressive clustering and compatible cluster merging algorithms are computationally expensive and need metrics for validation of individual clusters (Davé & Krishnapuram, 1997). Therefore, despite their potential, they are too complex for a simple initialization of a fuzzy neural network (FNN).

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 Ψ (2). Assuming that all data points are normalized in each dimension, the data set Z is bounded by a hypercube. In the algorithm, each point is seen as a potential cluster center, 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 $r_a > 0$ define the neighborhood radius for each cluster center. Therefore, the potential associated to each cluster depends on its distance to all of the points, leading to clusters with high potential where the neighborhoods are dense.

After computing the potential for each point, the one with higher potential is selected as the first cluster center. Let z_1^* be the center 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 center of the cluster (4):

$$P_i \Leftarrow P_i - P_1^* e^{-\beta \|z_i - z_1^*\|^2}, \quad (4)$$

where $\beta = 4/r_b^2$ and $r_b > 0$ represent the radius of the neighborhood for which significant potential reduction will occur. The radius for reduction of potential should be to some extent higher than the neighborhood radius to avoid closely spaced clusters. Typically, $r_b = 1.5r_a$. Since the points that are closer to the cluster center will have their potential reduced in a stronger way, the probability for any of those points to be chosen as the next cluster is lower. This procedure (selecting centers and reducing potential) is carried out iteratively, until the stopping criteria is reached, and is illustrated by Fig. 2.

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

One property of the algorithm is that the number of clusters to obtain is not pre-specified. However, it is important to note that the radii parameter 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 higher 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. However, despite the fact that some radii values should be tested, this parameter gives an initial hint on the number of clusters necessary (Paiva, 1999). This constitutes an important advantage over optimization-based and other classes of clustering algorithms, when little information is known regarding the best number of clusters. Another advantage of subtractive clustering is that the algorithm is noise robust, since outliers do not

significantly influence the choice of centers, due to their low potentials.

By the end of clustering, a set of g 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 centers of the membership functions for that variable are obtained by projecting the center of each cluster in the corresponding axis. As for the widths, they are obtained on the basis of the neighborhood radius, r_a , defined while performing subtractive clustering. Since Gaussian membership functions are used, their standard deviations are computed by

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

3.2. Parameter tuning by fuzzy neural network

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

The structure of the FNN is presented in Fig. 3. This structure can be found in Lin (1995). There, Gaussian membership functions are used. In the present work, two-sided membership functions (Fig. 4) are used, which are characterized by being asymmetric and containing a plateau, as a generalization of conventional functions. The increased flexibility of the generalized functions leads to better results, as a result of the increased flexibility.

The FNN 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):

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

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 centers of a two-sided Gaussian, σ_{ijL} and σ_{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

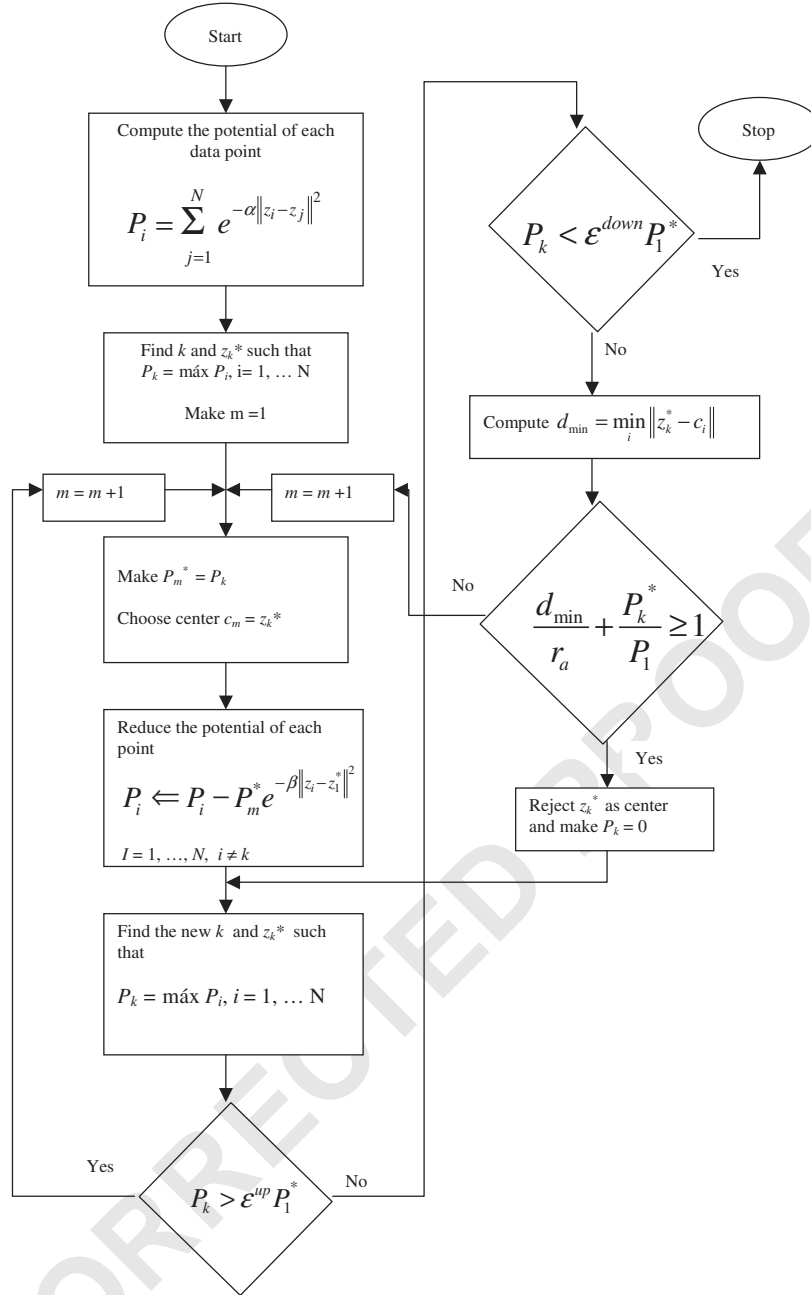


Fig. 2. Flowchart of the subtractive clustering algorithm.

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:

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

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 appointed for layer 3, a truncation operator, namely *max*, was used:

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

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 center

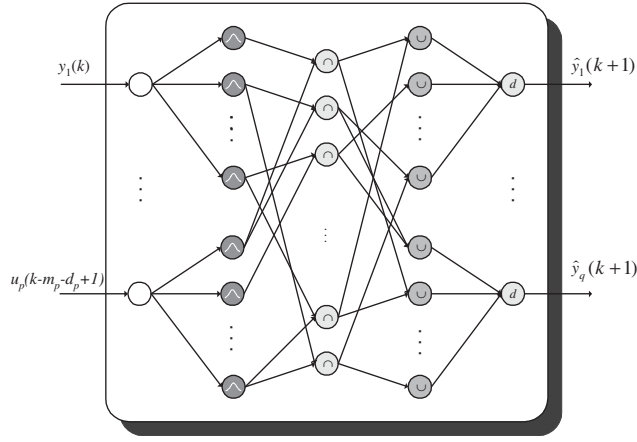


Fig. 3. Structure of the fuzzy neural net.

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

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

$$\delta_l^{(4)} = \sum_m \delta_m^{(5)} \frac{[(c_{lmL}\sigma_{lmL} + c_{lmR}\sigma_{lmR}) \sum_r (\sigma_{rmL} + \sigma_{rmR}) - (\sigma_{lmL} + \sigma_{lmR})a_l^{(4)} \sum_r (c_{rmL}\sigma_{rmL} + c_{rmR}\sigma_{rmR})]}{[\sum_r (\sigma_{rmL} + \sigma_{rmR})]^2}. \quad (15)$$

As in layer 2, c_{lmL} , σ_{lmL} , c_{lmR} , and σ_{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 optimization of the centers 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 minimize 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

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

where lr is the learning rate.

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

Layer 5: In this layer, the centers and widths of the output membership functions are updated. This is conducted by (12)–(14)

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

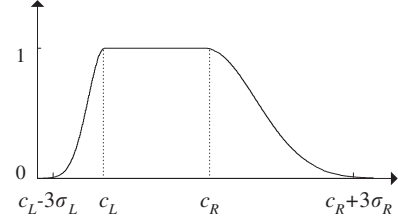


Fig. 4. Two-sided Gaussian membership functions used.

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

$$\frac{\partial E_m^{(5)}}{\partial \sigma_{lmL}} = -\delta_m^{(5)} \times \frac{c_{lmL} a_l^{(4)} \sum_r (\sigma_{rmL} + \sigma_{rmR}) - a_l^{(4)} \sum_r (c_{rmL} \sigma_{rmL} + c_{rmR} \sigma_{rmR})}{[\sum_r (\sigma_{rmL} + \sigma_{rmR})]^2}. \quad (14)$$

Layer 4: In this layer, there are no parameters to update. Therefore, only the error signals (δ) need to be computed for backpropagation (15):

Layer 3: As in layer 4, only the error signals need to be computed (16):

$$\delta_k^{(3)} = \sum_l \delta_l^{(4)} \frac{\partial a_l}{\partial a_k^{(3)}}. \quad (16)$$

Layer 2: In layer 2, the centers and widths of the input membership functions are updated according to Eqs. (17)–(21)

$$\frac{\partial E_m^{(5)}}{\partial c_{ij}} = \left(\sum_k \delta_k^{(3)} \frac{\partial a_k^{(3)}}{\partial a_j^{(2)}} \right) \frac{\partial a_j^{(2)}}{\partial c_{ij}}, \quad (17)$$

$$\frac{\partial E_m^{(5)}}{\partial \sigma_{ij}} = \left(\sum_k \delta_k^{(3)} \frac{\partial a_k^{(3)}}{\partial a_j^{(2)}} \right) \frac{\partial a_j^{(2)}}{\partial \sigma_{ij}}, \quad (18)$$

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

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

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

4. Results

Some of the measured variables are not sufficiently excited. 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 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 3. The sampling interval was defined in the mill as 1 h; 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 $g = 53$ rules. Fig. 3 presents the training results and Fig. 4 shows model

validation. There, the continuous line represents real process data, whereas the dashed line represents the model output. The variable is depicted is the final brightness achieved, as described in Section 2.

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: $rmse = 0.254$ (Figs. 5 and 6). It can, therefore, be concluded that the model obtained does not have satisfactory generalization 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 h (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

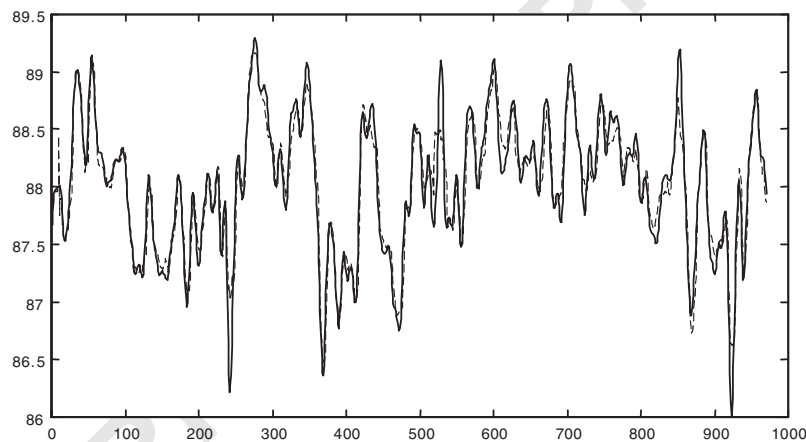


Fig. 5. Validation: (—) process, (- -) FNN output.

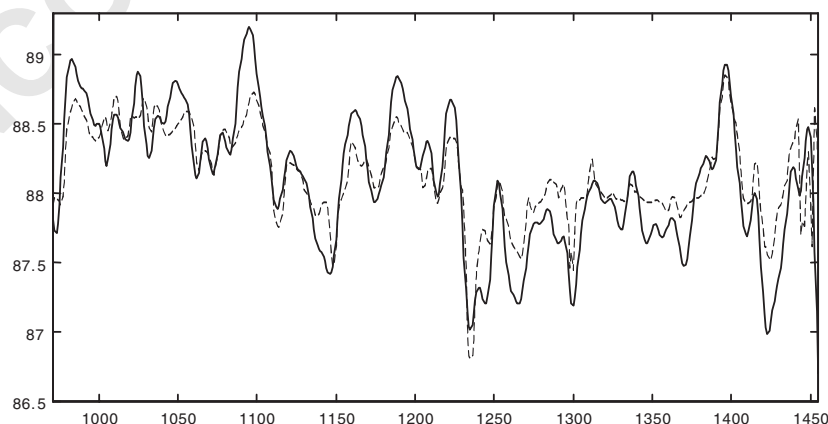


Fig. 6. Validation: (—) process, (- -) FNN output.

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 to 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.

5. Conclusions

A predictor for the output brightness of a bleaching plant (paper pulp industry), in a neuro-fuzzy framework, is developed in two phases: first, subtractive clustering is applied so as to obtain a set of fuzzy rules; then an FNN is trained to optimally tune the membership parameters using backpropagation.

Some problems were encountered related to the variable pure time delay of the process, which limited the accuracy of the obtained model. The system may also be time varying.

Another problem comes from the quality of industrial data. It seems that for a deeper foundation of the methodology, more data, with a lower sampling interval, should be used, in order to capture the short-term dynamics of the process. This means more and better instrumentation. However, the presented results illustrate the potential of the proposed methodology.

Acknowledgements

The authors thank to the anonymous reviewers whose comments have substantially improved the paper.

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

program PRAXIS XXI, contract (Alcine) PRAXIS XXI/EEI/11455/1998 and under POSI.

References

- Chiu, S. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3), 267–278.
- Davé, R. N., & Krishnapuram, R. (1997). Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Systems*, 5, 270–293.
- Delgado, M., Gomez, A. F., Karmeta, S., & Martin, F. (1997). A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling. *IEEE Transactions on Fuzzy Systems*, 5(5), 223–233.
- Duch, W., & Adamczak, R. (2001). A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks*, 12(2), 277–306.
- Lin, C.-T. (1995). A neural fuzzy control scheme with structure and parameter learning. *Fuzzy Sets and Systems*, 70, 183–212.
- Mamdani, E. H. (1977). Applications of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, C-26(12), 1182–1191.
- Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: Survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3), May.
- Mitra, S., Pal, S. K., & Mitra, P. (2002). Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks*, 13(1), 3–14.
- Paiva, R. P. (1999). *Identificação neuro-Difusa: Aspectos de interpretabilidade (Neuro-fuzzy identification: interpretability issues)*. M.Sc. thesis, Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Portugal (in Portuguese).
- Pedrycz, W., & Waletzky, J. (1997). Fuzzy clustering with partial supervision. *IEEE Proceedings of the SMC Part B*, 27(5), 787–795.
- Runkler, T. A. (1998). Automatic generation of first order Takagi-Sugeno systems using fuzzy c-elliptotypes clustering. *Journal of Intelligent and Fuzzy Systems*, 6(4), 435–445.
- Shi, Y., & Mizumoto, M. (2001). An improvement of neuro-fuzzy learning algorithm for tuning fuzzy rules. *Fuzzy Sets and Systems*, 118, 339–350.
- Wang, Y., & Rong, G. (1999). A self-organizing neural-network-based fuzzy systems. *Fuzzy Sets and Systems*, 103(1), 1–11.
- Yager, R. (1995). Fuzzy sets as a tool for modelling. *Computer Science Today*, Lecture Notes in Computer Science, no. 1000 (pp. 536–548). Berlin: Springer.

57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99
101
103
105
107
109
111