ELSEVIER

Journal of Process Control 10 (2000) 35–41

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Fuzzy neural networks for estimation and fuzzy controller design: simulation study for a pulp batch digester

K. Belarbi* K. Bettou A. Mezaache

Institute of Electronics University of Constantine Route de Ain el Bey Constantine Algeria

Received 9 June 1998; received in revised form 7 October 1998; accepted 3 March 1999

Abstract

A structural implementation of a fuzzy inference system through connectionist network based on MLP with logical neurons connected through binary and numerical weights is considered. The resulting fuzzy neural network is trained using classical back-propagation to learn the rules of inference of a fuzzy system by adjustment of the numerical weights. For controller design training is carried out off line in a closed loop simulation. Rules for the fuzzy logic controller are extracted from the network by interpreting the consequence weights as measure of confidence of the underlying rule. The framework is used in a simulation study for estimation and control of a pulp batch digester. The controlled variable the Kappa number a measure of lignin content in the pulp which is not measurable is estimated through temperature and liquor concentration using the fuzzy neural network. On the other hand a fuzzy neural network is trained to control the Kappa number and rules are extracted from the trained network to construct a fuzzy logic controller.

Keywords: Fuzzy neural network; Fuzzy estimation; Fuzzy control; Pulp batch digester

1. Introduction

Fuzzy logic controllers (FLC) based on the theory of fuzzy logic have found many applications in a variety of processes [1]. Usually rules for control are obtained from experienced operators through a knowledge acquisition process. However there have been attempts to construct systematic methods for designing FLC. Probably the first such a method was the PI fuzzy logic controller of MacVicar-Whelan [2] where the rules of inference are given in a fixed look-up table and the designer has to adjust scaling factors using trial and error. Lately however there has been growing interest in developing more systematic approaches [3]. These approaches can be classified into two broad classes: connectionist approaches and what may be called direct approaches. In the latter after fixing the shape of the membership functions and the number of the fuzzy set the search for optimality is carried out directly either in the space of the adjustable parameters of the membership functions [4] or in the space of both rule base and adjustable parameters [567] . Due to the complexity of the search space the method of optimisation has been almost exclusively the genetic algorithm. Connectionist approaches are various but they seek to build neural networks that realise the fuzzy inference systems leading to the so-called fuzzy neural network (FNN) [8]. The methodologies vary according to the structure of the network. In [9] Jang uses adaptive network based fuzzy inference system (ANFIS) to train a fuzzy controller based on rules of the Sugeno type. Temporal backpropagation is used to find the optimal weights. Ishibushi and Tanaka [10] proposed an FNN with fuzzy weights and biases and triangular membership functions. A generalisation of the method where membership functions can be of any shape was introduced by Lin and Lu in [11] who propose an improved learning algorithm. Hayachi et al. [12] use an FNN with crisp input and output fuzzy weight and triangular fuzzy sets to model a fuzzy controller. The FNN is then trained to learn the fuzzy control rules using a special learning algorithm based on fuzzy-valued cost function. In [13] the authors present a FNN with crisp input and fuzzy weights and use the so-called fuzzy backpropagation for learning. The implementation of Pedrycz [14] uses logic

* Corresponding author. Tel.: +213-492-3236; fax: +213-492-3237.

0959-1524/00/$ - see front matter
PII: S0959-1524(99)00038-4

based neurons that perform fuzzy and/or operations using t-norm and s-norm with numerical weights.

In this work a multi-layered perceptron is built to realise the rules of inference for a fuzzy system. In the proposed approach only the consequence weights are to be adjusted during learning. A simulation study is presented where the network is trained to learn the rules of inference through back propagation for the estimation and control of a pulp batch digester.

Batch pulp digesters convert wood into pulp through delignification. The most common process is the so-called Kraft process where a mixture of sodium hydroxide and sodium sulfide called the white liquor is used to separate the cellulose from the undesired lignin contained in wood. The reaction is usually stimulated by heating through recirculation in an external heat exchanger. The degree of delignification is measured by the Kappa number and the objective is to achieve a certain target Kappa number despite all possible variations. The controlled variable is then the Kappa number and the manipulated variables are usually the temperature and flow rate of the recirculationg liquor [15]. The control objective is to take the final value of the Kappa number as close as possible to a target value. There are however two major problems in the on line automatic control of the pulp digesters [16]: the inability to measure the Kappa number on line and the lack of understanding of the process. Various methods have been considered for the estimation and the control of the Kappa number [16 17]. Due its peculiarities the pulp digester is an ideal candidate for the application of fuzzy inference systems. Here a simulation study is presented for the control and estimation of the Kappa number of a batch pulp digester. In the following the construction of rules for a fuzzy inference system through the FNN is introduced and an application to the pulping process with results of simulation is presented.

2. Constructing rules of inference system through the FNN

In this section a scheme that can built rules for a fuzzy system through learning is presented. The fuzzy system is modelled as a fuzzy neural network which is trained to learn the rules of inference.

2.1. The basic network

When building a fuzzy model some parameters are chosen by the designer according to his knowledge of the system while others are left free to be tuned. In this work the width of the universes of discourse the number of fuzzy sets associated with a given linguistic variable and the shape of the membership functions are all set by the designer. The objective is to find an adequate set of rules. The fuzzy inference system is cast as a fuzzy neural network which is trained through back-propagation to learn the rules of inference. The network used here is based on the implementation of[14] and as shown in Fig. 1 has five layers. Nodes at layer one are input nodes with crisp input and crisp output. No computation is done in this layer. Nodes at layer two compute the value of the membership function. Each input variable x_i is fuzzified using N_i linguistic values. The output of these nodes is the membership function $\mu_{Aij}(x_i,)$ with fuzzy subsets $A_{ij.}$. Nodes in layer three perform the fuzzy AND operation through the algebraic product:

$$T(x, y) = xy \tag{1}$$

At a given node in the AND layer there is one incoming arc from all the fuzzy subsets associated with one input variable the output of an AND neuron can thus be written as:

$$y_{\mathrm{AND},k} = \prod_{i=1}^{i=n}\left(\sum_{j=1}^{j=N_i} w^i_{jk}\mu_{A_{ij}}(x_i)\right) k = 1 \dots na \tag{2}$$

$$\text{with } \sum_{j=1}^{j=N_i} w^i_{jk} = 1, \qquad \forall i, k \tag{3}$$

where na is the number of AND neurons and w^i_{jk} is the fixed binary (01) weight between neuron k of the AND layer and neuron j of layer two associated to input variable i. There are no weights to be adjusted at this layer. At layer four rules with the same consequence are integrated through the fuzzy OR operation which is implemented using the algebraic sum:

$$S(x, y) = x + y - xy \tag{4}$$

The output of the OR neuron is thus given by:

$$y_{\mathrm{OR},l} = S_k^{na}\left(T\left(y_{\mathrm{AND},k}, w_{kl}\right)\right) \qquad l = 1 \dots I_i \tag{5}$$

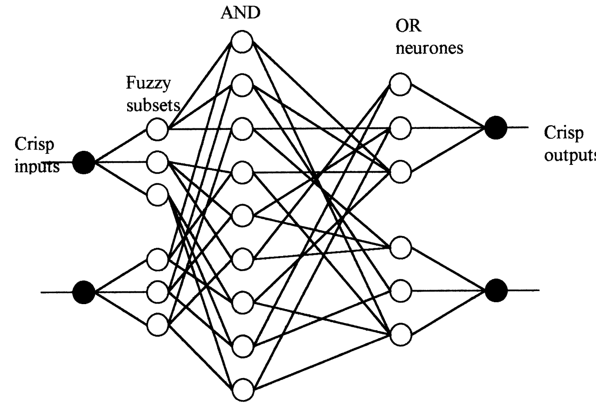

Fig. 1. Fuzzy neural network.

$$0 < w_{kl} < 1 \qquad \forall k, l \tag{6}$$

where w_{kl} is weight associated to arc linking node k of the AND layer and l of the OR layer and I_i is the number of fuzzy sets associated with output variable $y_i, i = i \ldots m$. Finally nodes in layer five realise the defuzzification of fuzzy subsets into crisp output using the center of gravity formula.

Since we only seek to derive rules of inference the parameters left for adjustment are the consequence weights that is those associated with arcs linking the AND to the OR neurons.

2.2. Learning scheme

Training of the network is performed using classical backpropagation. The basic configuration for training the network depends on its final use. If the FNN is to be used as estimator or plant emulator usual schemes for training neural nets given input output data can be used. On the other hand if the network is to learn to control a plant various structures can be used [18]. Here we propose a closed loop off line learning procedure as shown in Fig. 2 with classical error backpropagation. In order to back propagate the error a network mapping the input output functionality of the system is used as design model. The objective is to minimise the mean square error of the output. Taking into account the logical operators introduced above and the center of gravity formula for defuzzification and using the chain rule the error rate with respect to adjustable weight w_{kl} in the case of triangular fuzzy sets is given by:

$$\frac{\partial E}{\partial w_{kl}} = (y_{i'} - y_{di'}) \frac{m_{i'}j - y_{i'}}{\sum_{i=1}^{I_i} y_{\mathrm{OR},i}} y_{\mathrm{AND},k} \prod_{i=1}^{i=na} \left(1 - y_{\mathrm{AND},i} w_{il}\right) \qquad \forall k, l \tag{7}$$

where $m_{i'j}$ are mid points of output triangular equally spaced fuzzy sets associated with output $y_{i'}$ and $y_{di'}$ is the training data vector.

Once properly trained rules can be extracted from the FNN according to a simple heuristic. The procedure of rule extraction from the FNN used here is based on the fact that the adjustable weights can be interpreted as a measure of confidence of the underlying rule of inference. Hence among all arcs outgoing from one AND neuron the one with the highest weight is selected and the underlying rule is put in the look-up table of the FLC.

3. Simulation study: an application to the estimation and control of the pulp digester model

In order to illustrate the effectiveness of the procedure it is now applied on simulation for the estimation and control of the pulp digester. The control of the pulp batch digetster is a nonlinear problem since there is no operating point where one can obtain a linear model for design. Moreover the controlled variable the Kappa number is not directly measurable and must be inferred from other measurements. These are usually the temperature of the mixture in the digester and the effective alkali. Recently novel approaches have been proposed for the control and monitoring of the batch digester. In particular Venkateswarlu and Gangiah [19] used measurements of the temperature and effective alkali in conjunction with the extended Kalman filter to manipulate the cooking temperature for the control of the Kappa number. On the other hand the research team at Auburn University [16 20] used an extended Kalman filter with four states measurements to estimate the Kappa number and applied nonlinear model based predictive control with the recirculation temperature and flow rate as control variables.

In this work a simulation study of the estimation and control of the Kappa number is carried out using the framework outlined in the previous sections. The estimation of the Kappa number is implemented through a fuzzy neural network while the controller is a fuzzy logic controller obtained after rule extraction from a trained FNN. Results are presented based on the application of

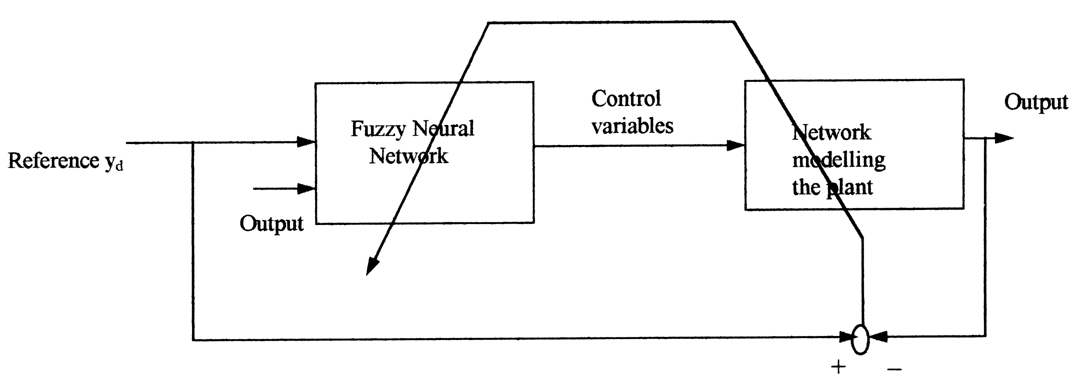

Fig. 2. Training of the controller network.

the proposed schemes to a simulation of the nonlinear model of Williams et al. [21] also in [16].

3.1. Estimation of the Kappa number

A fuzzy neural network is used for the estimation of the Kappa number. The measured variables are the reaction temperature T and the effective alkali Al which are input to the FNN whose output is the estimated Kappa number K_{p_e}. Three linguistic variables LOW MEDIUM and HIGH with triangular membership functions symmetrical partitioning and evenly spaced midpoints (given in Table 1) are used for all input output signals. The network is trained to learn the mapping between the input variables the digester temperature and alkali and the output variable the Kappa number. The training data set is obtained through simulation of the nonlinear model of [21]. The model is run with the same initial conditions and parameters as in [16] and correspond to what will be called here the nominal case. The inputs are chosen so that they sweep all their domain of variation. Table 2 gives some of the data used. The computed values of the reaction temperature the effective alkali and the Kappa number are used to train the network. In real life however measured values should be used and the Kappa number is inferred off-line using for instance empirical models. Simulation time is 2 h corresponding to one batch time with sampling rate at 0.025. Initial weights are set to zero and the learning rate in the back propagation process is varied heuristically from 0.001 to 0.0001 (these features are used throughout the study for all the training procedures i.e. estimator controller and emulator). Training was stopped after 900 presentations of the 80 training values corresponding to 72000 iterations with a root mean square error of 0.503. Once the learning process is terminated the network is directly implemented as an estimator without rule extraction. However when the estimator is implemented on-line midpoints of the membership functions associated with the Kappa number (as given in Table 1) should not be kept constant since the initial value of the Kappa number depends only on raw wood quality and not on temperature and alkali. This is taken into account by selecting the parameters of the membership functions according to raw wood quality at the beginning of the batch of wood change and then fine tuning the choice batch after batch relying on operator knowledge. This is a key factor in the proposed estimating scheme that would not have been possible if for example neural networks were used.

Table 1
Midpoints for the triangular fuzzy sets

Variable	a_1	a_2	a_3
T_r	350	410	460
F_r	0.5	0.75	1
t' in K°	353	415	480
Alc	16	25	34
K_p	40	110	180
Error	−10	0	+10
Error increment	−2	0	+2

3.2. Control of the Kappa number

In this paragraph an FLC is designed for the Kappa number control. The inputs to the controller are the error $e(k)$ between reference and estimated Kappa number and its increment $\Delta e(k)$ and the outputs the recirculation temperature T_r and flow rate F_r. All variables are fuzzified with three fuzzy subsets LOW MEDIUM HIGH and the same features as above with midpoints as in Table 1. The FLC is modelled as an FNN as described above with inputs $e(k)$ and $\Delta e(k)$ and outputs T_r and F_r. There are thus 54 weights to be adjusted. The training is carried off line. A neuro fuzzy plant emulator trained before hand is used as design model. Its inputs are the two control variables and the previous value of the Kappa number with the Kappa number as output. The cost function to be minimised when training the FNN controller is given by

$$E(k) = \frac{1}{2}\left(K_{p_r}(k) - K_{p_e}(k)\right)^2 \tag{8}$$

where K_e is the estimated Kappa number and K_r is the reference trajectory taken as in [18]:

$$K_{p_r}(k+1) = K_{p_e}(k) - \frac{(Kp_e(k) - Kp_d)}{(80-k)} k = 1\ldots 79 \tag{9}$$

Kp_d is the target Kappa number equal to 50. Training was stopped after 90000 iterations with an RMSE of 0.9010. Rules are then extracted giving the rule base in Table 3.

Table 2
Data for the batch digester

Parameter	Value
Maximum recirculation temperature T_r	250°C
Maximum recirculation temperature T_r	80°C
Maximum recirculation flow rate F_r	0.5 m/h
Minimum recirculation flow rate F_r	1 m/h
Maximum change of T_r per sampling period	10°C
Maximum change of F_r per sampling period	0.1 m/h
Initial Kappa number	172.5
Initial batch temperature	80°C
Initial alkali	35

4. Results

Applications of the FLC and the estimator designed above are applied to simulations of the nonlinear model of the pulp batch digester. The model comprises 14 equations and a number of reaction parameters. The first test was carried out for the nominal case (from which the data for the training procedures were taken). Fig. 3 shows the Kappa number trajectories. It can be seen that not only the Kappa number reaches its target values but that its estimated value is always close the true value which was calculated using the total lignin contents [16]. In order to empirically analyse the robustness of the FLC simulation tests were carried out. Two types of tests are considered.

Table 3
Rules of base of the FLC first entry is F_r; second entry is T_r

	Er		
dEr	L	M	H
L	HH	HM	ML
M	HH	ML	LL
H	HH	LM	MH

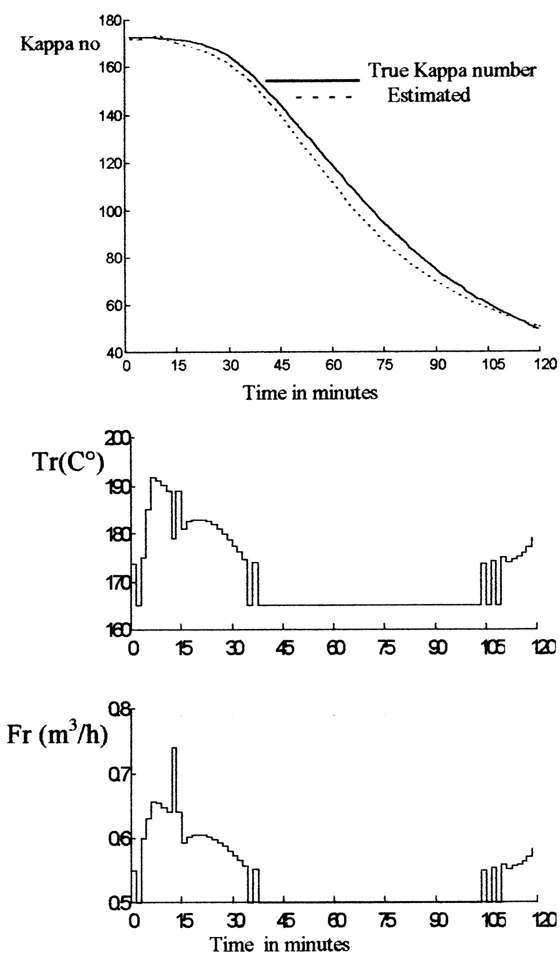

Fig. 3. Closed loop response nominal case: Kappa number and control variables.

4.1. Parameters variations [16]

The parameters taken for the simulation model are probably not exactly equal to their true values. In order to analyse the effect of these discrepancies the following tests are presented. For the sake of comparison all variations are as in [16 20]. Note however that for the estimation of the Kappa number four measurements were used therein whereas we used only two measurements.

A change of 100% in the first frequency factor of one wood component: Fig. 4 shows that no important changes are introduced in the behaviour of the closed loop system. The result is similar to that in [16] although their Kappa number approaches its true value after of 1 h.

A change of 50% in the second activation energy of one wood component: as shown in Fig. 5 the behaviour of the closed loop is not altered whereas the estimator–controller of [16] fails.

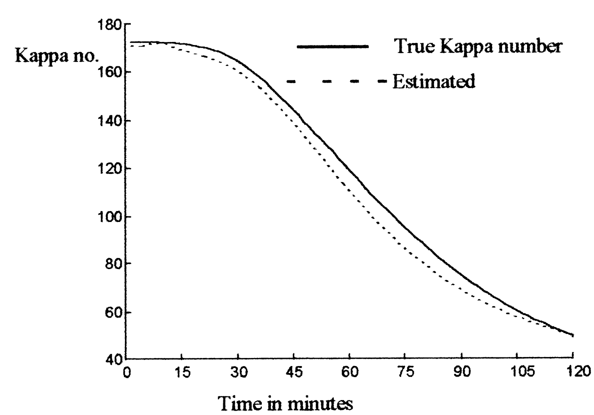

Fig. 4. Closed loop response change of 100% in A_{11}: Kappa number.

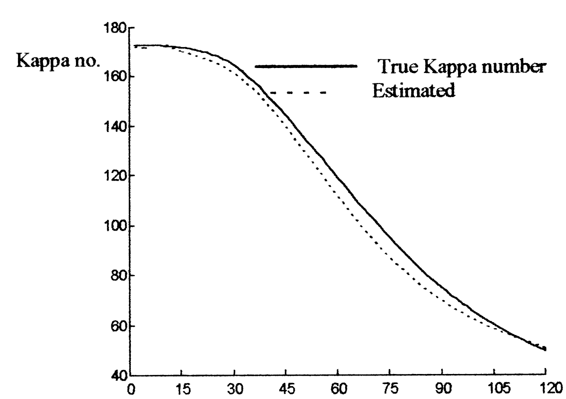

Fig. 5. Closed loop response change of 50% in E_{22}: Kappa number.

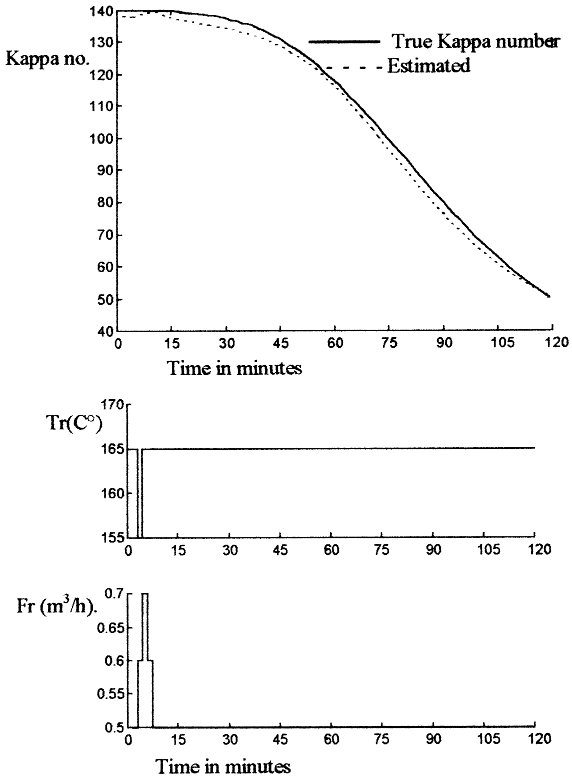

Fig. 6. Closed loop response soft wood: Kappa number and control variables.

The performance of the proposed scheme is excellent and although the FLC is quite simple with only nine rules it consistently outperformed the results obtained in [16].

4.2. Changes in wood (feed) characteristics [20]

Soft wood: the type of wood is changed from nominal to a softer type inducing a change in the initial Kappa number. To cope with this change at the estimator level midpoints of the membership functions should be altered. Since wood is softer than nominal the Kappa number is lower. A first guess of (30 90 150) resulted in a final error equal to 5. At the beginning of the second batch the midpoints were again altered to (35 87.5 140) and the results are shown in Fig. 6 this time the Kappa number reaches its target with an error of 0.355.

Hard wood: when wood is changed from soft to hard the initial Kappa number increases. Midpoints were changed to (60 150 240) resulting in a final error of 8. At the beginning of the second batch the mid-points were again altered to (50 135 220) results are shown in Fig. 7 and the final error is 0.505.

A change of the size of wood chips: with the same wood quality the size of wood chips is now doubled. The results of the simulation run are shown in Fig. 8 and are exactly as in the previous test. Note that this time no alterations of midpoints is necessary.

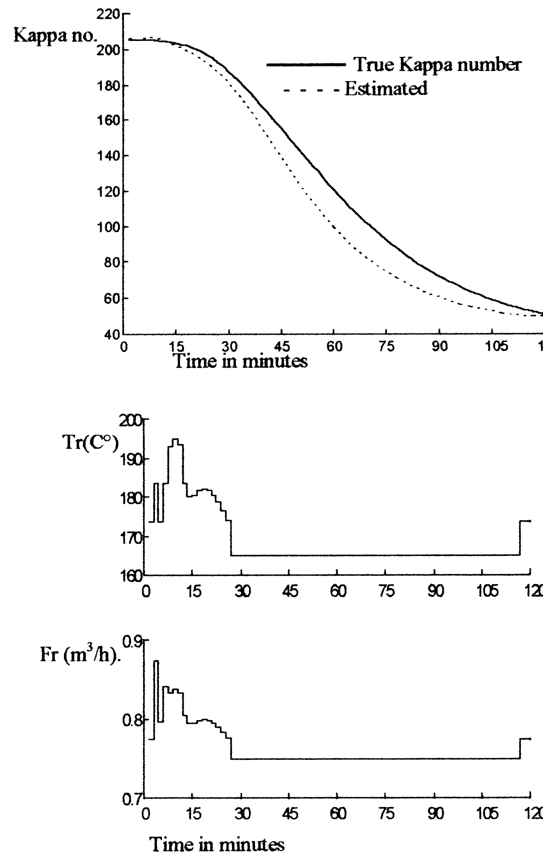

Fig. 7. Closed loop response hard wood: Kappa number and control variables.

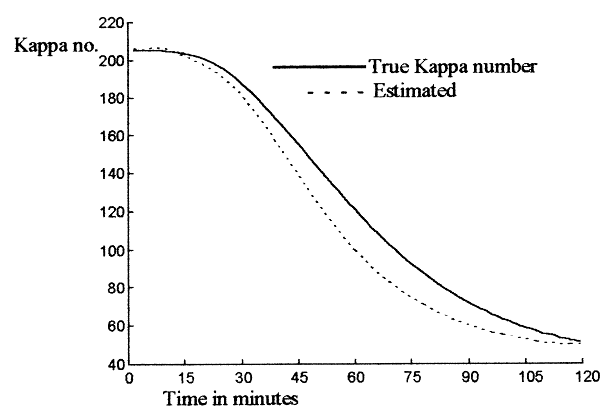

Fig. 8. Closed loop response hard wood 100% change in the size of wood chips: Kappa number.

Again the results shown for the changes in wood characteristics compare favourably with those obtained in [20]. In fact during batch time the estimated Kappa number is closer to its true value than in [20] where the results shown concern the third and the fourth batches after the change.

5. Conclusion

A fuzzy neural network that learns rules of inference for a fuzzy system through classical backpropagation has been proposed. For fuzzy logic controller design the network is trained off-line in a closed loop simulation. In order to back propagate the error signal a network model of the plant is used as a design model. Rules were extracted from the trained network to build the rule base of the fuzzy logic controller. The scheme was applied to the estimation and control of a batch pulp digester. The controlled variable the Kappa number which cannot be measured on line is estimated with the same type of fuzzy neural network but without rule extraction through the measurements of the batch temperature and the concentration of the alkali. This estimator is shown to be flexible enough to cope with initial changes of the Kappa number through alterations of the midpoints of the universe of discourse. The training of the fuzzy neural controller was carried out using a fuzzy neural model of the plant. Although the FLC was quite simple with only nine rules simulation results show good degree of robustness in the face of parameter variations and changes in operating conditions. This novel methodology involving simultaneous estimation and control using fuzzy logic gave promising results which are better those obtained elsewhere with less on line computation burden and fewer measurements for estimation.

References

[1] C.C. Lee IEEE Trans. SMC-20 (2) (1990) 404.
[2] P.J. MacVicar-Whelan Int. J. Man-Machine Studies 8 (1976) 687.
[3] D.A. Linkens H.O. Nyongesa IEE Proc. D 143 (4) (1996) 367.
[4] C.L. Karr AI Expert 6 (2) (1991) 26.
[5] D.A. Linkens H.O. Nyongesa IEE Proc. D 142 (3) (1995) 161.
[6] A. Homaifar E. McCormick IEEE Trans. Fuzzy Sets 3 (2) (1995) 129.
[7] J. Kim P. Ziegler IEEE Trans. Fuzzy Systems 4 (3) (1996) 213.
[8] J.J. Buckley Y. Hayashi Fuzzy Sets and Systems 66 (1994) 1.
[9] R.J. Jang IEEE Trans. Neural Netw. 3 (5) (1992) 716.
[10] H. Ishibushi H. Tanaka H. Okada Proceedings of the International Joint Conference on Neural Networks San Francisco CA 1993 p. 1650.
[11] C. Lin Y. Lu IEEE Trans SMC-26 (1996) 745.
[12] Y. Hayashi J.J. Buckley E. Czogala Proceedings of the International Joint Conference on Neural Networks 1992 vol II p. 412.
[13] D. Nauck R. Kruse Proceedings IEEE International Conference on Neural Networks 1993 p. 1022.
[14] W. Pedrycz Fuzzy Sets and Systems 56 (1993) 1.
[15] T. Senbon F. Hanabuchi Instrumentation Systems Fundamentals and Applications Springer Verlag 1991.
[16] J.H. Lee A.K. Datta AIChE journal 40 (1) (1994) 50.
[17] G.A. Dumont Automatica 22 (2) (1986) 143.
[18] M. Agarwal IEEE Control system magazine 1997 April 75
[19] C. Venkateswarlu K. Gangiah Ind. Eng. Chem. Res. 31 (1992) 848.
[20] A.K. Datta J.H. Lee S. Vanchinathan G.A. Krishnagopolan Proceedings of the American Control Conference Baltimore MD 1993 p. 500.
[21] T.J. Williaws C.C. Smith T. Christiansen L.F. Albright Pulp and Paper Canada 85 (5) (1984) 55.