Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Engineering Applications of Artificial Intelligence 17 (2004) 265–273
Engineering Applications of ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study

José Vieira[a,*], Fernando Morgado Dias[b], Alexandre Mota[c]

[a] Escola Superior de Tecnologia de Castelo Branco, Departamento de Engenharia Electrotécnica, Av. Empresário, Castelo Branco 6000, Portugal
[b] Escola Superior de Tecnologia de Setúbal, Departamento de Engenharia Electrotécnica, Campus do IPS, Estefanilha, Setúbal 2914-508, Portugal
[c] Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, Aveiro 3810, Portugal

Received 1 September 2003; received in revised form 22 December 2003; accepted 8 March 2004

Abstract

This article presents a comparison of artificial neural networks and neuro-fuzzy systems applied for modelling and controlling a real system. The main objective is to model and control the temperature inside of a kiln for the ceramic industry. The details of all system components are described. The steps taken to arrive at the direct and inverse models using the two architectures: adaptive neuro fuzzy inference system and feedforward neural networks are described and compared. Finally, real-time control results using internal model control strategy are presented.

Using available MATLAB software for both algorithms, the objective is to show the implementation steps for modelling and controlling a real system. Finally, the performances of the two solutions were compared through different parameters for a specific real didactic case.

Keywords: Temperature control; Fuzzy hybrid systems; Artificial neural networks; Applied neuro-fuzzy control; Model-based control and real-time control

1. Introduction

The purpose of the present paper is to compare, using a didactic case, two solutions for modelling that became very popular in the last decades: artificial neural networks (ANN) and neuro fuzzy systems (NFS). This comparison is made using available MATLAB software.

The field of ANN has crossed different stages of development. One of the most important steps was achieved when Cybenko (1989) proved that they could be used as universal approximators. A negative stage had been brought two decades earlier by the book of Minsky and Papert called Perceptrons (Minsky and Papert, 1969), where among other examples it was shown that a single layer of perceptrons could not represent a simple function like the Exclusive OR. This negative phase was overcome when algorithms for training of multilayer ANN where proposed in the decade of the 1980s. Since then much work has been done regarding ANN and their application to many different fields. A reasonable slice of this work has been in the modelling and control field where ANN hold the promise of being capable of producing non-linear models and controllers, being able to work under noise conditions and being fault tolerant to the loss of neurons or connections.

The field of NFS starts in the end of the 1980s and presents a big growth in the decade of the 1990s with a large variety of different approaches. These approaches mix the ANN with fuzzy inference systems (FIS) in three ways: cooperative, concurrent and fused. The most common architecture is the fused NFS that uses neural networks ideas just to learn some internal parameters of a fixed structure (Nauck et al., 1997). The adaptive neuro fuzzy inference system (ANFIS) belongs to the fused NFS, it was introduced by Jang (1992) and it is able to approach any linear or non-linear function (universal approximator) (Jang, 1993).

The ANN are widely used in model and control of many practical industry non-linear process applications

*Corresponding author. Electronics and Telecommunications, University of Aveiro, Campus Santiago, Aveiro 3810, Portugal. Tel.: +351-234-370-381; fax: +351-234-381-128.
E-mail addresses: zevieira@est.ipcb.pt (J. Vieira), fmdias@est.ips.pt (F.M. Dias), alex@det.ua.pt (A. Mota).

0952-1976/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2004.03.001

some with on-line model tuning (Ngia and Sjöberg, 2000) and some with off-line model tuning (Lightbody and Irwin, 1997; Bloch et al., 1997). The NFS are also widely used, some with on-line model/controller tuning (Fink et al., 2001) and some with off-line model/controller tuning (Kovacevic and Zhang, 1997; Zhang and Kovacevic, 1998; Vieira and Mota, 2003). Since the ANN and NFS are so frequently used, this paper proposes the discussion and comparison of these two off-line approaches through several specific parameters (interpretability, structure complexity, accuracy, time training and model based control results) for a specific didactic application.

The present didactic study is a reduced scale prototype kiln for the ceramic industry, which is non-linear and will be working under measurement noise. Using available MATLAB software for both algorithms, this article shows the details of the implementation steps for modelling and controlling a real system and compares the performances of the two solutions through different parameters.

To test the models in real-time control action the internal model control (IMC) strategy was used.

2. The kiln

Non-linearity and noise have always been a major problem in control systems. This type of kilns are non-linear systems because their temperature depends not only on the heating control variable but also on the exchange of heat with the exterior world and the present system also has measurement noise because of the type B thermocouple used.

The system is composed of a kiln, electronics for signal conditioning, power electronics module and a Data Logger from Hewlett-Packard HP34970A to interface with a personal computer (PC) connected as can be seen in Fig. 1.

Through the Data Logger bi-directional real-time information is passed: control signal supplied by the controller and temperature data for the controller. The temperature data are obtained using a thermocouple.

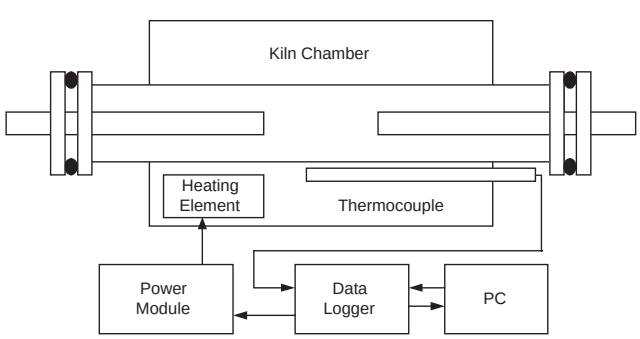

Fig. 1. Modules that compose the system.

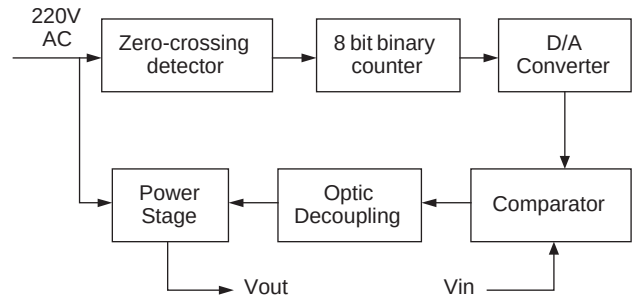

Fig. 2. Block diagram of the power module.

The power module receives a signal from the Data Logger, with the resolution of 12 bits (0–4.095 V imposed by Data Logger), which comes from the controller implemented in the PC, and converts this signal in a pulse width modulation (PWM) signal of 220 V applied to the heating element.

The signal conversion is implemented using a sawtooth wave generated by a set of three modules: zero-crossing detector, binary 8 bit counter and D/A converter. The sawtooth signal is then compared with the input signal generating a PWM type signal.

The PWM signal is applied to a power amplifier stage that produces the output signal. The signal used to heat the kiln produced this way is not continuous, but since the kiln has an integrator behavior this does not affect the functioning.

The block diagram of the power module can be seen in Fig. 2.

The Data Logger is used as the interface between PC and the rest of the system. Since the Data Logger can be programmed using a protocol called Standard Commands for Programmable Instruments (SCPI), a set of functions have been developed to provide MATLAB with the capability to communicate through the RS-232C port to the Data Logger.

Using the HP34902A (16 analog inputs) and HP34907A (digital inputs and outputs and two Digital to Analog Converters) modules together with the developed functions it is possible to read and write values, analog or digital, from MATLAB. A picture of the system can be seen in Fig. 3. The kiln is in the center and at the lower half are the prototypes of the electronic modules.

3. Identification and modelling

In the identification phase, the observation of the step response of the kiln in several reference temperatures and using the *lipschit* function (Nørgaard, 1996b) to determine the lag space, a conclusion was achieved: the kiln can either be considered a first or a second-order system followed by a time delay.

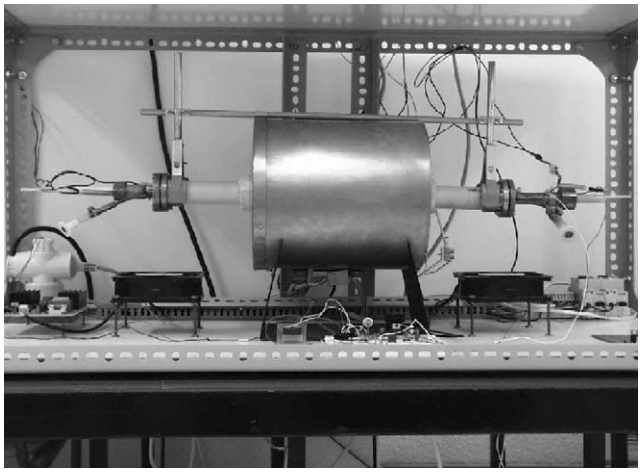

Fig. 3. Picture of the kiln and electronics.

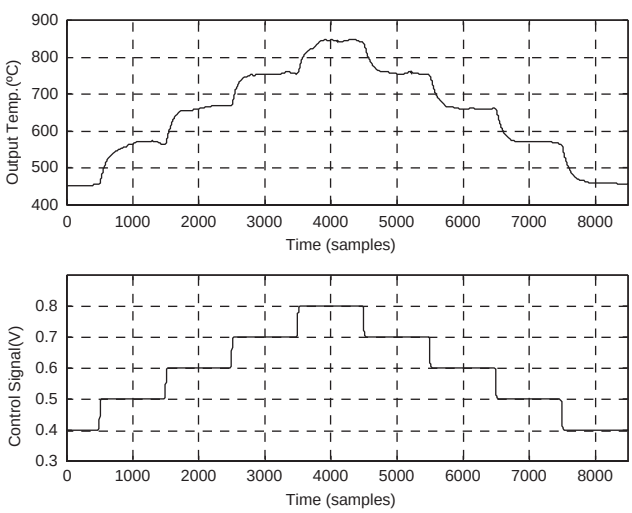

Fig. 4. Open-loop step response of the kiln.

About the characteristics of the kiln it can be said that the nominal temperature operating range is from 300°C to 1000°C, the nominal power is about 1 kW, the rise and fall times are different and the static system gain varies depending on the value of the temperature of the kiln as shown in the step open-loop response illustrated in Fig. 4.

The kiln shows non-linearity, due to the different behavior in the heating and in the cooling phases. This is because of the energy losses of the kiln structure to the exterior that are dependent of the temperature in the kiln.

The identification data have been chosen to respect two important requirements: frequency and amplitude spectrum wide enough (Jang, 1992; Sørensen, 1994). Therefore, the train signals are acquired in open loop and the control signal was generated in the working range (0–1.5 V) by a pseudo-random function. Finally, with a sampling period (h) of 30 s, the operation of

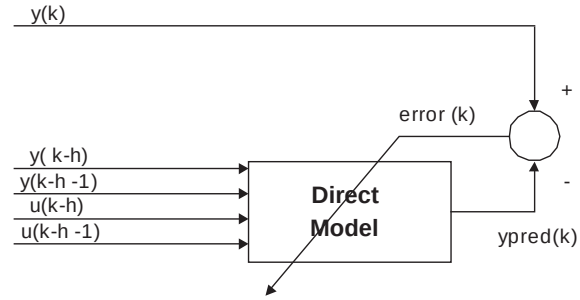

Fig. 5. Structure for direct model training (first and second-order system).

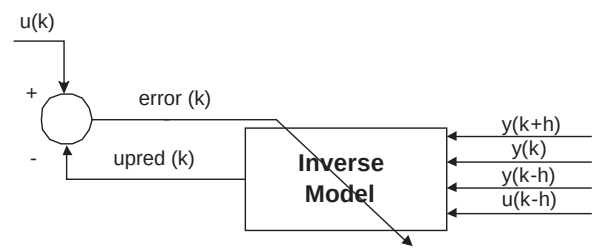

Fig. 6. Structure for inverse model training (first and second-order system).

collecting data was made. As the measurements have noise the train and test signals were filtered from high-frequency noise. From train/test signals, it can be confirmed that the rise and fall times are different, the static system gain varies depending on the value of the temperature of the kiln and the time delay of the system is constant and equal to one sample ($h = 1$).

The used training structures for direct and inverse models are described in Figs. 5 and 6. These structures are the most common solutions for training models and are described in several articles (Pradeep and Ash, 1988; Hunt and Sbarbaro, 1991). Considering the analysis done the kiln direct and inverse models were obtained with two different groups of regressors. First, considering the kiln a first-order system and second, considering the kiln a second-order system.

Considering that $k = n^*h$, where k is the time instant, n is the iteration and h the sampling time, the prediction of the temperature (direct models of the system) at time k, $y\text{pred}(k)$ is given by Eqs. (1) and (2), for first and second-order approaches:

$$y\text{pred}(k) = f(y(k-h), u(k-h)), \tag{1}$$

$$y\text{pred}(k) = f(y(k-h), y(k-h-1), u(k-h), u(k-h-1)). \tag{2}$$

To obtain the inverse models, the prediction of the control signal at time k, making $k = k + h$ in Eqs. (1) and (2), $u\text{pred}(k)$ is given by Eqs. (3) and (4) for first and

second-order approaches:

$$upred(k) = f(y(k+h), y(k)), \tag{3}$$

$$upred(k) = f(y(k+h), y(k), y(k-h), u(k-h)). \tag{4}$$

Considering the kiln's behavior, ANN like feedforward neural networks (FNN) and NFS systems like ANFIS were considered good approaches for building the direct and inverse non-linear models of the kiln.

3.1. ANN

ANN are artificial and simplified models of the neurons that exist in the human brain. They can be used as a black box approach to create models of systems profiting of the facility to model non-linear (as well as linear) systems. Their ability relies on the quality of the signals used for training and the performance of the training algorithms and their parameters do not contain information that can be directly understood by the human operator or that can easily be related to the physical properties of the system to be modelled. FNN are a subtype of ANN in which the only connections allowed between neurons are feedforward, i.e. there are no lateral or feedback connections.

FNN Architecture: A FNN is a layered structure, which can include non-linearity. The basic element of a FNN is the neuron that is shown in Fig. 7.

The neuron implements the general equation

$$y = F\left(\sum_{i=1}^{n} li.wi\right), \tag{5}$$

where usual functions for F are sigmoidal, linear and hard limit. A FNN is composed of an input layer, one or more hidden layers with one or more neurons and an output layer where frequently the neurons are linear. The multi-input single output FNN in Fig. 8 implements the following general equation:

$$y = F_1\left(\sum_{j=1}^{nh} w'_{j1} f_j\left(\sum_{l=1}^{nl} w_{lj} l_l\right)\right). \tag{6}$$

A typical structure of FNN can be seen in Fig. 4.

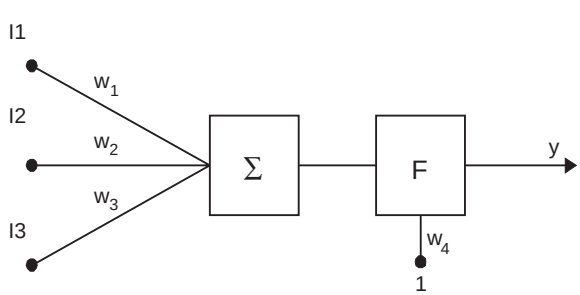

Fig. 7. Neuron structure—the bias input is shown connected to the activation function to show its effect on biasing the activation function.

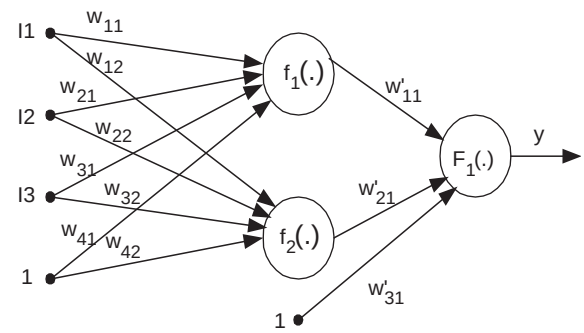

Fig. 8. FNN structure.

Table 1
Characteristics of FIS and ANN

ANN	FIS
Black box	Interpretable
Learning for scratch	Making use of linguistic knowledge and heuristics

Learning Algorithms of FNN: Many algorithms have been developed to use with FNN like the well-known Backpropagation or the most effective Levenberg–Marquardt. The algorithms developed or adapted for the use with FNN are based on minimizing a criterion (which is most frequently based in the error between the desired and the obtained output). Most of them are based on derivative calculations of the error as a mean to minimize it. The Levenberg–Marquardt algorithm was chosen because of the robustness and fastest convergence.

3.2. NFS

A FIS can use human expertise by storing its essentials components in a rule base, and perform fuzzy reasoning to infer the overall output value. The derivation of if-then rules and corresponding membership functions depends, a lot, on the a priori knowledge about the system. However, there is no systematic way to transform experiences and knowledge of human experts to the knowledge base of a FIS. There is also a need for adaptability or some learning algorithms to produce outputs within the required error rate. On the other hand, ANN learning mechanism does not rely on human expertise. Due to the homogenous structure of ANN, it is difficult to extract structured knowledge from either the weights or the configuration of the ANN. Table 1 summarizes the characteristics of FIS and ANN.

FIS and ANN are complementary which induce the appearance of the NFS that take advantage of the capacity that FIS have to store human expertise knowledge and the capacity of learning of the ANN. A common way to apply a learning algorithm to a FIS is

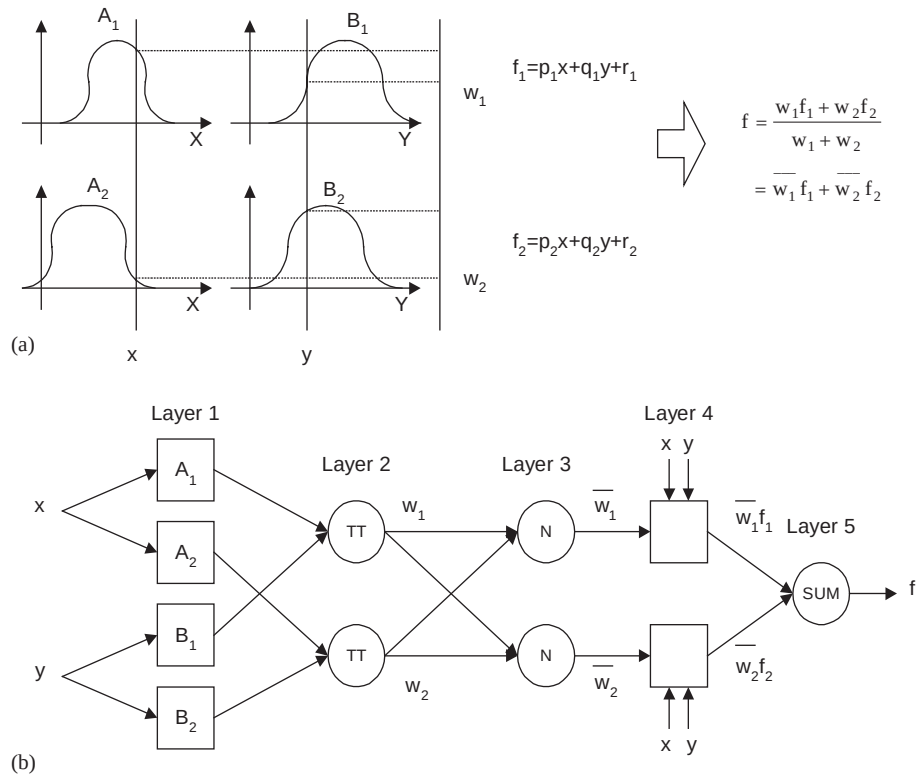

Fig. 9. (a) A two–input first-order Sugeno fuzzy model with two rules; (b) equivalent ANFIS architecture.

to represent it in a special ANN like architecture, which is what we have in ANFIS.

In this work ANFIS was the NFS solution chosen because of the robustness and fastest convergence.

ANFIS Architecture: The ANFIS architecture (Jang, 1993) is illustrated in Fig. 9.

Assume that the fuzzy inference system under consideration has two inputs x and y and one output z, *for example*. For the first-order Sugeno fuzzy model a common rule set with two fuzzy if–then rules is the following Rules 1 and 2, Eq. (7):

$$\begin{aligned}&\text{Rule}_1: \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ then}\\ &f_1 = p_1x + q_1y + r_1;\\ &\text{Rule}_2: \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{ then}\\ &f_2 = p_2x + q_2y + r_2.\end{aligned} \tag{7}$$

Fig. 9(a) illustrates the reasoning mechanism for this Sugeno Model and the corresponding equivalent ANFIS architecture is shown in Fig. 9(b), where nodes of the same layer have similar functions.

The output f in Fig. 9(b), can be written as

$$\begin{aligned}f &= \frac{w_1}{w_1+w_2}f_1 + \frac{w_2}{w_1+w_2}f_2\\ &= \overline{w_1}(p_1x + q_1 + r_1) + \overline{w_2}(p_2x + q_2 + r_2)\\ &= (\overline{w_1}x)p_1 + (\overline{w_1}y)q_1 + (\overline{w_1})r_1\\ &\quad + (\overline{w_2}x)p_2 + (\overline{w_2}y)q_2 + (\overline{w_2})r_2.\end{aligned} \tag{8}$$

This way an adaptive network that is functionally equivalent to a first-order Sugeno fuzzy model is constructed. From the ANFIS architecture shown in Fig. 9(b), it can be seen that when the values of the premise parameters (layer 1) are fixed, the overall output can be expressed as a linear combination of the consequent parameters (layer 4).

Learning algorithms of ANFIS: The learning algorithms are composed of two phases:

- In the forward pass of the hybrid learning algorithm, node outputs values go forward until layer 4 and the consequent parameters are identified by the least squares method.
- In the backward pass, the output errors are propagated backward and the premise parameters are updated by gradient descent method.

3.3. FNN and ANFIS structures for identification

For the FNN structure, as there is no rule to determine the ideal number of neurons in the hidden layer, a wide range of values were tested to search for the best solution for first- and second-order system models. For first-order approach models, the best solutions were obtained using three neurons for the direct model and four for the inverse model (hidden layer). For second-order approach models, the best solutions were obtained

using three neurons for the direct model and six for the inverse model (hidden layer). The models have one output neuron with linear activation function.

In the ANFIS structure, the standard ANFIS structure was used to obtain the direct and the inverse models. For first-order approach models, the structure contains four rules, two inputs with two membership functions each (bell shaped with three non-linear parameters each). For second-order approach models, the structure contains 16 rules with two membership functions each (bell shaped with three non-linear parameters each). All the models have one output that is a linear function of the consequent parameters.

Using the data collected and divided into training and test sets, direct and inverse models were identified using the latest structures parameters definition. The identification procedures were performed using MATLAB Fuzzy Logic Toolbox (MATHWORKS, 1996) tools for ANFIS models and the NNSYSID (Nørgaard, 1996b) and NNCTRL (Nørgaard, 1996c) toolboxes for MATLAB for FFN models (Nørgaard, 1996a).

Training was performed off-line for both solutions. In the FNN and ANFIS model approaches, removing the mean and dividing by their standard deviation, the train and test data sets were normalized.

3.4. Comparison of the FNN and ANFIS models

Fig. 10 shows the training (first $\frac{3}{4}$ of the points) and testing sets (last $\frac{1}{4}$ of the points) used to create the models ($y(k)$-temperature and $u(k)$-control signal).

The number of training epochs corresponds roughly to stopping the training when the minimum of the test data error was achieved. To compare the precision of the models, the mean square error (MSE) criterion was used in train and test sets. The results are shown in Tables 2 and 3 (direct and inverse models, respectively).

Filtering train/test data contributes for achieving very small values of the train and test errors and for the convergence of the gradient decent learning methods used.

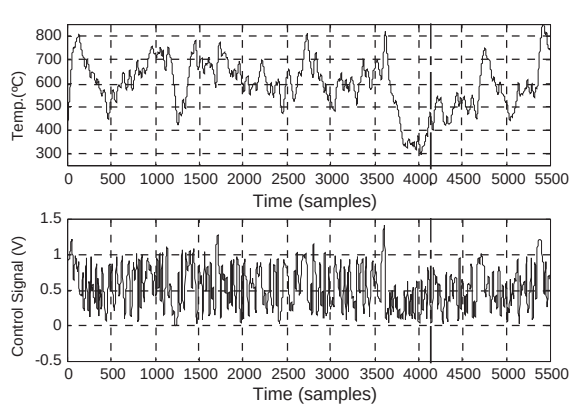

Fig. 10. Training and test data sets.

Table 2
MSE of direct models

MSE	Train	Test
Direct model with FNN1	0.233	0.266
Direct model with FNN2	5.3e-2	5.4e-2
Direct model with ANFIS1	0.195	0.194
Direct model with ANFIS2	1.3e-3	2.2e-3

Table 3
MSE of inverse models

MSE	Train	Test
Inverse model with FNN1	5.8e-3	5.1e-3
Inverse model with FNN2	1.5e-3	1.2e-3
Inverse model with ANFIS1	2.1e-3	2.2e-3
Inverse model with ANFIS2	3.3e-5	5.6e-5

(FNN1—FNN models for first-order approach, FNN2—FNN models for second-order approach, ANFIS1—ANFIS models for first-order approach, ANFIS2—ANFIS models for second-order approach).

Table 4
Complexity comparison of the direct models

Complexity comparison	FNN 1	FNN 2	ANFIS 1	ANFIS 2
No. of parameters	13	19	24	104
No. of training epochs	80	150	30	50
Training time (s)	27	85	12	531

Table 5
Complexity comparison of the inverse models

Complexity comparison	FNN 1	FNN 2	ANFIS 1	ANFIS 2
No. of parameters	17	37	24	104
No. of training epochs	15	50	1	50
Training time (s)	8	89	1	531

As can be seen, in the second-order approach the errors are smaller than the ones in the first-order approach. In general the ANFIS structure achieved smaller MSE when compared with the same order approach using FNN.

The obtained models can also be compared in terms of complexity by measuring the number of parameters, number of training epochs and the training time.

The results are summarized in Tables 4 and 5 for direct and inverse models, respectively.

From Tables 4 and 5, it can be seen that for first-order approach, the number of parameters is smaller and the training time is higher in FNN than in ANFIS. For second-order approach, the number of parameters and the training time are smaller in FNN than in ANFIS, because it was used the standard ANFIS structure (number of rules $= 2^{\text{number of inputs}}$).

4. Control structure

To test the obtained models for both approaches and both architectures, the IMC structure was used.

IMC consists of connecting in series with the plant the inverse model and in parallel with the plant the direct model. The difference between the output of the model and the output of the plant will generate an error $ef(k)$ that will be feedback (Pradeep et al., 1988; Dias Fernando and Mota Alexandre, 2001a) and subtracted to the reference input signal $r(k+h)$. Direct model implements Eqs. (1) and (2) and inverse model implements Eqs. (3) and (4). In this two last equations the $y(k+h)$ is substituted by $r(k+h)$ that is the desired temperature signal h iterations ahead. If $r(k+h)$ is not available it can be used $r(k)$ but the output y will delayed of h samples. This solution can be seen in Fig. 11.

5. The real-time control action

The controllers were directly tested in the kiln, when the simulation results were considered satisfactory.

Figs. 12 and 13 show the results of FNN and ANFIS internal model controllers for the first- and second-order approach (reference signal ($r(k+h)$, control signal $u(k)$ and error signal $e(k)$ from first-order approach and finally the same signals for the second-order approach).

A PID controller is also presented as a reference controller for the comparison of the control results. The PID was optimized using a genetic algorithm search procedure (Dias Fernando and Mota Alexandre 2001b) for a reference equal to the illustrated in Fig. 14 in order to obtain a better performance. The parameters of the PID controller are: $Td=6$, $Kp=65$, $Ki=10$ $Ti=2$ and the PID control results are illustrated in Fig. 14.

Comparing solutions always involves the minimization of some kind of metric measure function. In this particular case the measure used was the MSE function.

The MSE of the four IMC controllers for first and second-order approaches and for the PID controller are presented in Table 6 after initial stabilization (from 50 to 300 samples). These first 50 samples are not used in the cost function, because the data logger limits the measured temperature to values superior to 300°C

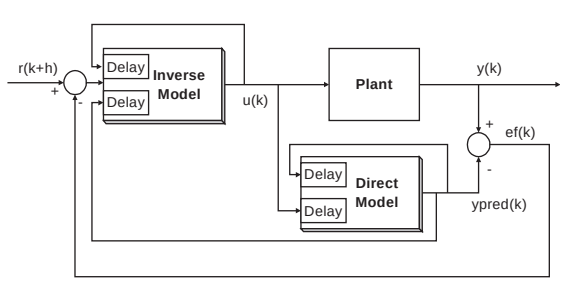

Fig. 11. Generic structure for IMC.

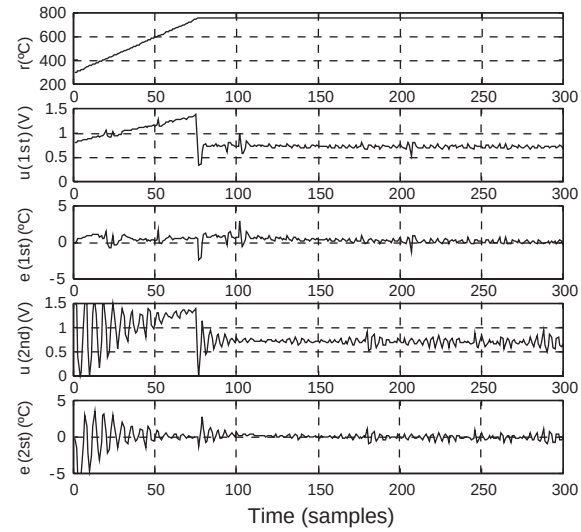

Fig. 12. IMC using FNN models for second-order approach.

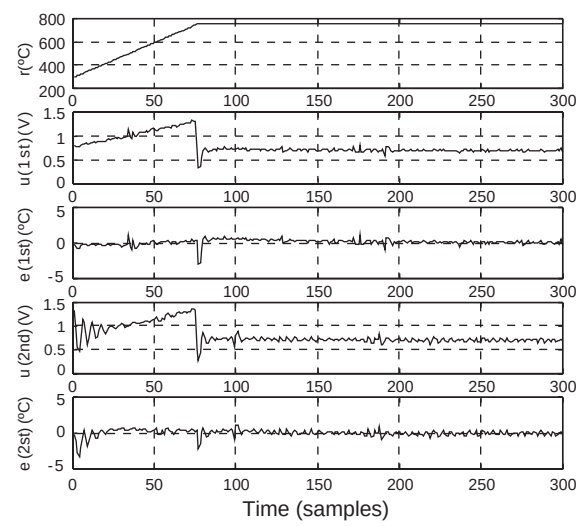

Fig. 13. IMC using ANFIS models for second-order approach.

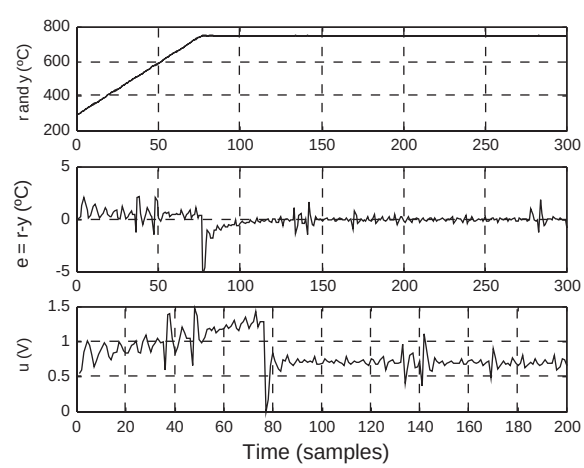

Fig. 14. PID control results.

Table 6
Comparison between the four controllers

Controller	MSE
PID Control	0.505
IMC using FNN1 models	0.321
IMC using FNN2 models	0.178
IMC using ANFIS1 models	0.186
IMC using ANFIS2 models	0.116

(due to the characteristics of the thermocouple). Therefore, to control the temperature of the kiln, a non-controlled pre-heating phase is needed to put the kiln at 300°C. The control testes do not start from steady state 300°C, but from a ramp growing temperature that reach 300°C with a velocity of about 7°C for sample. This pre-heating phase will affect the first controlled samples of the heating process, especially in the second-order approaches.

As can be seen from the Figures, after initial stabilization, the second-order approach achieves better results.

As can be seen from Table 6 the results of first-order ANFIS and second-order FNN are similar but the ones achieved by ANFIS for second-order are better than the ones achieved by FNN.

6. Conclusions

In the modelling of this kiln, the train/test errors achieved are similar in ANFIS and in FNN structure. The most significant difference can be found in the first-order approach, since the errors achieved are smaller in ANFIS than in FNN structure. The number of parameters and time training of the models are bigger in general in ANFIS than in FNN structure. This could be caused because the standard ANFIS structure was used, which is not very flexible. It could be interesting to test other structures like zero-order Sugeno type and cluster data before training to get simpler structures.

In spite of the more oscillate control signal in both second-order solutions, the IMC results show that the second-order approach gets better MSE results. The first-order approach controllers take longer to achieve a zero stationary error, because the models are not so "good". The second-order approach controllers give better results, because they can hold more information, learning all the characteristics of the kiln, so for these models the direct models and the kiln are almost equal.

From the results of Table 6 the controllers with ANFIS models gives smaller errors than the equivalent ones with FNN models. Their complexity is similar for first-order models, but ANFIS becomes more complex for second-order models. For the FNN solution the use of second-order models is justified by the results obtained while for the ANFIS solution a first-order approach would be enough. Both solutions are valid options for the modelling of real systems under measurement noise.

References

Bloch, G., Sirou, F., Eustache, V., Fatrez, P., 1997. Neural intelligent control for a Stell Plant. IEEE Transactions on Neural Networks 8 (4), 910–918.

Cybenko, G., 1989. Approximation by superposition of a sigmoidal function. Mathematics of Control, Signals and Systems 2, 492–499.

Dias Fernando, M., Mota Alexandre, M., 2001a. Comparison between different control strategies using neural networks. 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia.

Dias Fernando, M., Mota Alexandre, M., 2001b. A comparison between a PID and internal model control using neural networks. 5th World Multi-Conference on Systemics, Cybernetics and Informatics, Vol. 5, Orlando, EUA, pp. 268–273.

Fink, A., Nelles, O., Fischer, M., Isermann, R., 2001. Nonlinear adaptive control of a heat exchanger. International Journal of Adaptive Control and Sig.1 Proc. 15 (8), 883–906.

Fuzzy Logic Toolbox for Matlab, Mathworks, 1996. Technical Report.

Hunt, K.J., Sbarbaro, D., 1991. Neural networks for linear internal model control. IEEE Proceedings-D, Control Theory and Applications 138 (5), 431–438.

Jang J.S.R., 1992. Neuro-fuzzy modeling: architecture, analyses and applications. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA.

Jang, J.S.R., 1993. Anfis: adaptive-network-based fuzzy inference systems. IEEE Transactions on System, Man, and Cybernetics 23 (3), 665–685.

Kovacevic, R., Zhang, Y.M., 1997. Neurofuzzy model-based weld fusion state estimation. IEEE Control Systems Magazine 17 (2), 30–42.

Lightbody, G., Irwin, G.W., 1997. Nonlinear control structures based on embedded neural system models. IEEE Transactions on Neural Networks 8 (3), 553–567.

Minsky, M., Papert, S., 1969. Perceptrons. MIT Press, Cambridge, MA.

Nauck, D., Klawonn, F., Kruse, R., 1997. Foundations of Neuro-Fuzzy Systems. Wiley, New York.

Ngia, L., Sjöberg, J., 2000. Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg–Marquardt algorithm. IEEE Transactions in Signal Processing 48 (7), 1915–1927.

Nørgaard, M., 1996a. System identification and control with neural networks. Ph.D. Thesis, Department of Automation, Technical University of Denmark.

Nørgaard, M., 1996b. Neural network system identification toolbox for MATLAB. Technical Report.

Nørgaard, M., 1996c. Neural network control toolbox for MATLAB. Technical Report.

Pradeep, B., Ash, D.R.H., 1988. Computer Process Control w/ Advanced Control Applications, second ed. revised and enlarged. Instrument Society of America.

Sørensen, O., 1994. Neural networks in control applications. Ph.D. Thesis, Department of Control Engineering, Institute of Electronic Systems, Aalborg University, Denmark.

Vieira, J., Mota, A., 2003. Smith predictor based neural fuzzy controller applied in a water gas heater that presents a large time-delay and load disturbances. Proceedings of IEEE International Conference on Control Application, Vol. 1, pp. 362–367.

Zhang, Y.M., Kovacevic, R., 1998. Neurofuzzy model based control of weld fusion zone geometry. IEEE Transactions on Fuzzy Systems 6 (3), 389–401.

José A. B. Vieira received his M.S. degrees in Electronics Engineering from the University of Aveiro Portugal in 1997. He is with the Department of Electronic and Telecommunications of the Polytechnic School of Castelo Branco and he belongs to the IEETA investigation unit in University of Aveiro.

Currently, he is working in his Ph.D. studies in Electronic Control Engineering at the University of Aveiro. His research interests include Industry Non-Linear Model Identification, Fuzzy systems, Neuro Fuzzy Architectures, Model Based Smith Predictive Control, Hammerstein and Wiener Modelling, Adaptive Control, On-line Identification, Varying Time Delay Systems.

Fernando Morgado Dias received his *Diplôme D'Études Approfondies* in Microelectronic from University Joseph Fourier in Grenoble, France in 1995 and currently teaches at the Escola Superior de Tecnologia de Setubal, Portugal and is preparing his Ph.D. degree at the University of Aveiro.

Alexandre M. N. Mota received his Ph.D. in Automatic Control Electronics Engineering from the University of Aveiro, Portugal in 1993 and he is a teacher in Electronics Department in the University of Aveiro. Currently he is president of the Portuguese Automation Control Association and he belongs to the IEETA investigation unit. He has published more than 40 papers in the field of automatic control, adaptive systems, neuro-network system modelling, fuzzy modelling, distributed control systems, linear and non-linear control.