Comparison between Different Control Strategies using Neural Networks

Fernando Morgado Dias and Alexandre Manuel Mota

Abstract—The work reported in this article concerns the temperature control loop of a reduced scale prototype kiln. The kiln's characteristics are studied and presented.

Non-linearity and noise are two characteristics of the plant as far as control is concerned. Direct and inverse models of the kiln are implemented using Feedforward Neural Networks and using these models Direct Inverse Control, Additive Feedforward Control and Internal Model Control are implemented and their results compared. These structures provided from classical control are used almost unchanged except for Internal Model Control, which needs to be adapted. Details about this implementation are given. The utility of each of the control strategies is outlined in the conclusions.

Index Terms— Feedforward Neural Networks, Direct Inverse Control, Additive Feedforward Control, Internal Model Control and Measurement Noise.

I. Introduction

The field of Neural Networks (NN) has known different stages of development. One of the most important steps was achieved when Cybenko [2] proved that they could be used as universal approximators. A negative stage was brought by the book of Minsky and Papert called Perceptrons[9], 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 NN where proposed in the decade of the 80s.

Since then much work has been done regarding NN and their aplication to many different fields.

A reasonable slice of this work has been in the modelling and control field where NN 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.

Many theoretical results have been presented regarding systems without noise and many others regarding systems with simulated noise, but it still remains work to be done applying NN to real systems with noise and this is where the present work fits.

The system under control is a reduced scale prototype kiln and the work reported concerns the implementation of the temperature control loop. The modelling of this kiln is part of an interdepartmental project at the University of Aveiro, which will lead to the control of the atmosphere inside the kiln using one loop for temperature control and another for air/oxygen ratio control.

The kiln's characteristics are studied and presented. Direct and inverse models of the kiln are implemented with Feedforward Neural Networks (FNN) and using these models Direct Inverse Control, Additive Feedforward Control and Internal Model Control are tested and their results compared. The utility of each of the strategies is outlined in the conclusions.

II. The System

Non-linearity and noise have always been a major problem in control systems. Kilns are non-linear systems because their temperature does not depend 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, cooling system and a Data Logger from Hewlett Packard HP34970A to interface with a Personal Computer (PC) connected as can be seen in figure 1.

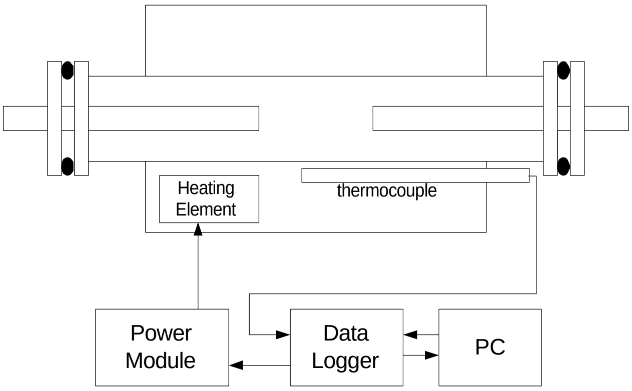

Fig. 1. Schematic of the modules composing the system.

F. M. Dias is with the Escola Superior de Tecnologia do Instituto Politécnico de Setúbal, Campus do IPS, Estefanilha 2914-508 Setúbal, Portugal Tel: +351 65 790000, Fax:+351 65 721869, Email: fdias@est.ips.pt

A. M. Mota is with the Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, 3810 Aveiro, Portugal, Tel: +351 34 370381, Fax:+351 34 381128, Email: alex@det.ua.pt

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

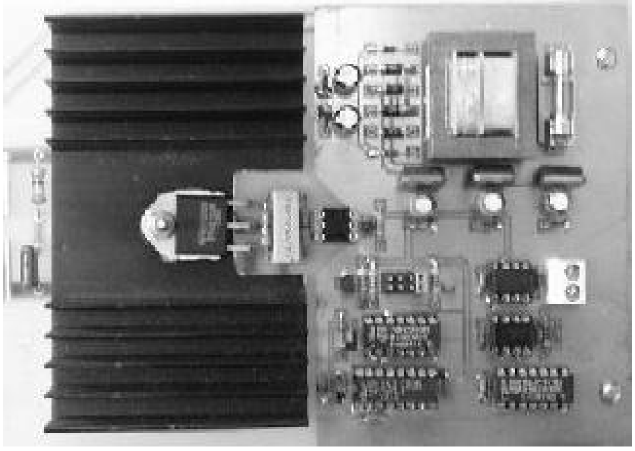

Fig. 2. Picture of the power module.

The power module, which can be seen in figure 2, receives a signal from the controller implemented in the Personal Computer (PC), which ranges from 0 to 4.095V and converts this signal in a power signal of 220V applied during a period of time proportional to the input signal.

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 capability to communicate through the RS-232C port to the Data Logger.

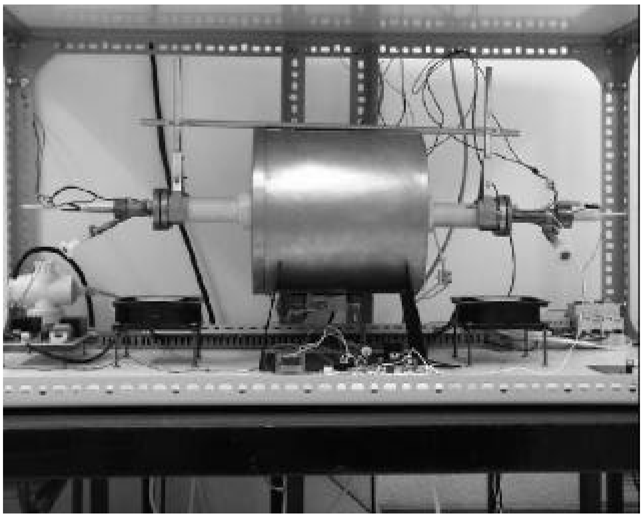

Fig. 3. Picture of the kiln and electronics.

Using the HP34902A and HP34907A modules together with the developed functions it is possible to read and write values, analog or digital from MATLAB. The first module provides 16 analog inputs and the second provides digital inputs and outputs and two Digital to Analog Converters.

A picture of the system can be seen in figure 3. The kiln can be seen in the centre and at the lower half the prototypes of the electronic modules and the cooling fans.

III. Identification

Starting from the characteristics already identified about this system: non-linearity and noise, the use of Neural Networks seemed to be straightforward and FNN were used because their simplicity and powerful algorithms available for training. The use of past samples of input and output provides a memory effect in FNN that enlarges their field of use.

Because of the measurement noise all the data was filtered using a simple first order filter with multiple iterations. Care was taken to avoid phase distortion and to choose appropriate cut-off frequency.

To proceed for the identification phase the data to be used had to fulfil one requisite: frequency and amplitude spectrum wide enough [3].

With this concern direct and inverse models for the system were produced using Feedforward Neural Networks (FNN) of one hidden layer, with linear output and Auto-Regressive with eXogenous input (ARX) architectures.

Many structures have been proposed for training [5], [6], [16] and [17], mostly for inverse models. In the present work the structures used are depicted in the block diagrams that can be seen in figures 4 and 5 because of their simplicity.

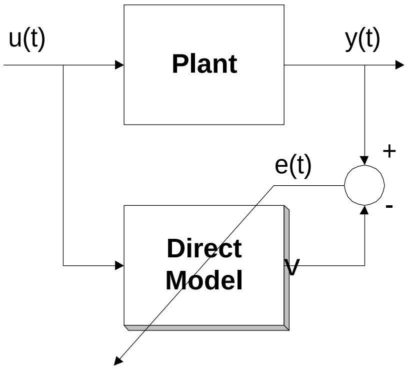

Fig. 4. Forward model training block diagram.

The lag space was analysed searching for correlations and two regressors of output and input were used.

As there is no rule to determine the ideal number of neurons in the hidden layer, a wide range was tested to search for the best solution. The best solutions were obtained using four neurons for the direct model and five for the inverse model.

The models were trained with Levenberg-Marquardt algorithm because of its fastest convergence.

When the quality of the models was considered to be “good”, the models were used for inverse control simulation and later used in the control strategies presented in the next section.

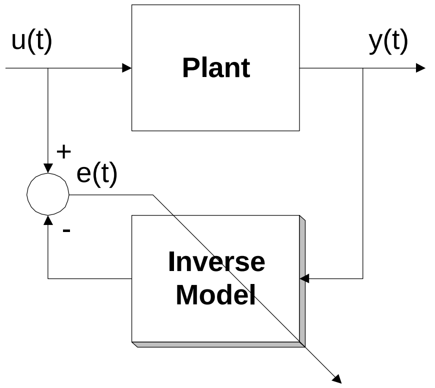

Fig. 5. Inverse model training block diagram.

One common problem that arises during training is overtraining or overfitting. This corresponds to having the FNN modelling not only the features of the system but to an undesirable extent also the noise [16].

The overtraining problem has been an open topic for discussion motivating the proposal of several techniques like Regularization [12], Early stopping [14] and pruning - Optimal Brain Damage [11] and Optimal Brain Surgeon [10]. In the present work both models were trained using early stopping.

During the identification and control tasks the NNSYSID [4] and NNCTRL [5] toolboxes for **MATLAB** were used.

IV. Control Structures

The control structures used in the present work are: Direct Inverse Control, Additive Feedforward Control and Internal Model Control.

A. *Direct Inverse Control (DIC).*

Direct inverse control is the simplest solution for control that consists of connecting in series the inverse model and the plant as can be seen in figure 6.

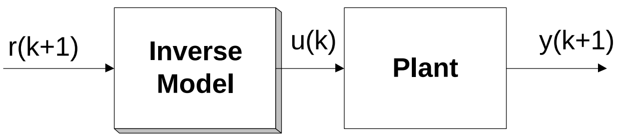

Fig. 6. Structure for Direct Inverse Control. The signal r(k) is the reference, u(k) the control signal and y(k) the output signal.

Considering that the Plant's behaviour is described by P(k) and I(k) describe the inverse model, the relation between input r(k) and output y(k) is:

$$y(k) = \frac{P(k)}{I(k)}.r(k) \qquad (2)$$

using H(k) to note the transfer function:

$$H(k) = \frac{P(k)}{I(k)} \qquad (3)$$

Thus the output y(k) will only follow the reference signal r(k) if I(k) is an accurate inverse of P(k).

B. *Additive Feedforward Control (AFFC).*

The principle of additive feedforward control is quite simple: add to an existing (but not satisfactory functioning) feedback controller an additional inverse process controller. The principle of AFFC is illustrated in figure 7.

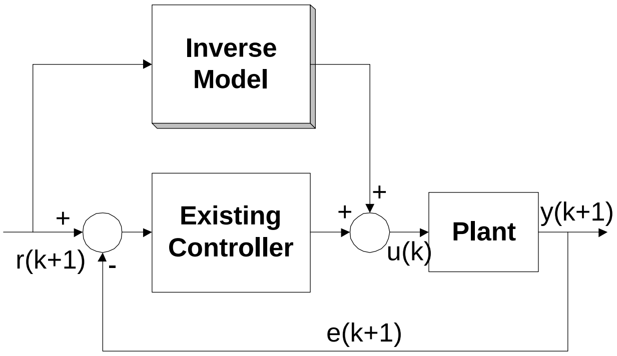

Fig. 7. Structure for Additive Feedforward Control.

The additive feedforward control strategy offers the following important advantages [7]:

- Data collecting can be done using the existing closed loop, avoiding plant stopping for data collection and facilitating the access to good quality data.
- There is no need for opening the existing control loop nor during training neither during the introduction of the additive controller.

The condition for AFFC to work properly is the same needed in DIC: the inverse model needs to be accurate.

C. *Internal Model Control (IMC)*

Internal Model Control is a structure that allows the error feedback to reflect the effect of disturbance and plant mismodelling.

From the block diagram of figure 8 it is possible to derive the following expression (considering the Direct model described by D(k)):

$$H(k) = \frac{P(k).I(k)}{1 - I(k).D(k) + P(k).I(k)} \qquad (4)$$

Since y(k)=H(k).r(k), if the Direct and Inverse model have a good matching I(k).D(k) equals 1 and H(k) reduces to 1, making y(k)=r(k). The conclusion is that in IMC a good match between forward and inverse models is enough to have good control. It can also be shown that IMC reduces disturbance’s influence.

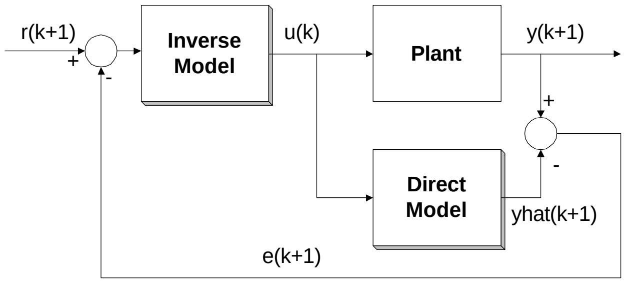

Fig. 8. Structure for Internal Model Control. The signal r(k) is the reference, u(k) the control signal, y(k) the output signal, yhat(k) the estimate of the output and e(k) the error between the output and the estimate.

D. Adapting the Control Structures to use Neural Networks Models.

The need for NNs arises when dealing with non-linear systems for which the linear controllers and models do not satisfy and the use of structures provided by classical control theory seems a straightforward strategy.

The structures presented in the sections A and B can be used with NN models without the need of major changes, but the structure used in section C needs some refinements to work properly[8].

According to the conclusions extracted from equation 4, the models need to match, that is the inverse model should be the inverse of the forward model instead of the inverse of the system. The forward model is implemented according to the following equation:

$$yhat(k+1) = f\begin{bmatrix} yhat(k),\ldots, yhat(k-n_y+1), \\ u(k-t_d),\ldots,u(k-t_d-n_u+1) \end{bmatrix} \quad (5)$$

And the inverse model:

$$u(k) = g\begin{bmatrix} r(k+1), yhat(k),\ldots, yhat(k-n_y+1) \\ u(k-t_d),\ldots,u(k-n_u-t_d+1) \end{bmatrix} \quad (6)$$

Where n_y is the number of previous output samples used, n_u is number of previous control signal samples used and t_d is the time delay of the system. The block diagram of the resulting control loop can be seen in figure 9.

This matching between the models would normally point out to specialized training, though in the present work the better results where achieved with normal training as depicted in figure 5.

V. The Real Time Control Action

The three proposed control strategies were implemented according to the details given in the previous sections and the results can be seen in figures 10 to 12.

The summary of the results is shown in table 1 in the form of Mean Square Error (MSE) for an easier comparison. The first column shows the error over the entire set point. It is worth to mention that the training data starts at 300ºC and therefore the computation of the first samples in the NN is done without past information. The other two columns refer to partial evaluations of the MSR.

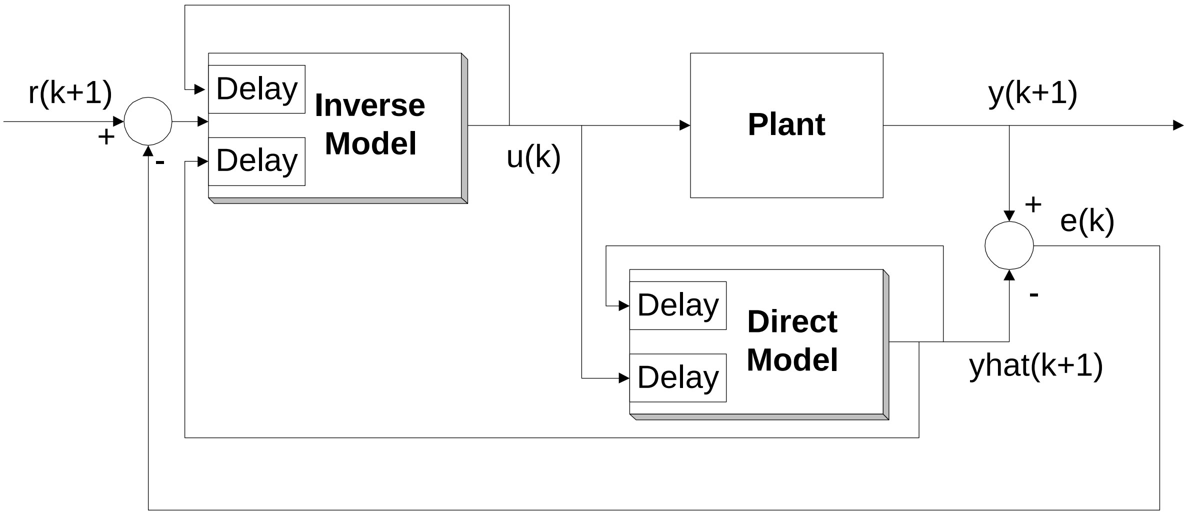

Fig. 9. Internal Model Control structure with detail of the implementation of inverse and direct model.

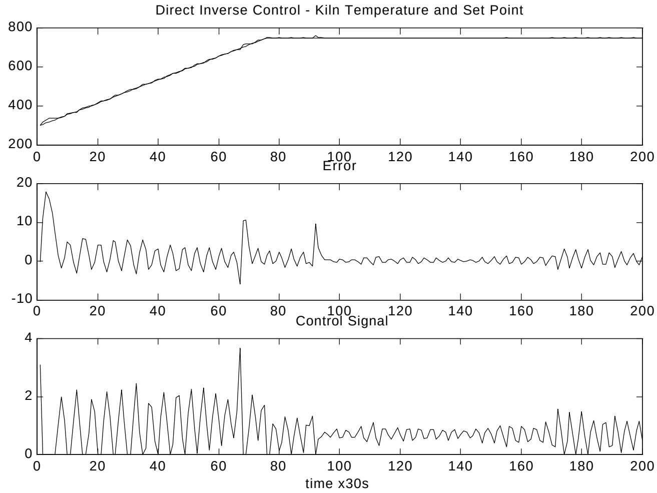

Fig. 10. Direct Inverse Control results.

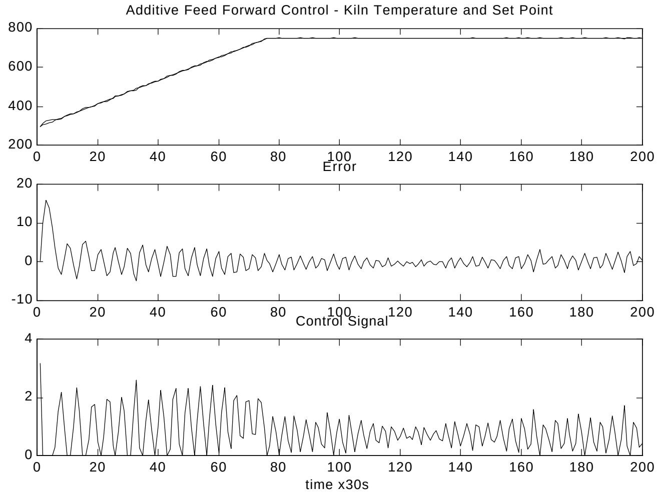

Fig. 11. Additive Feedforward Control results.

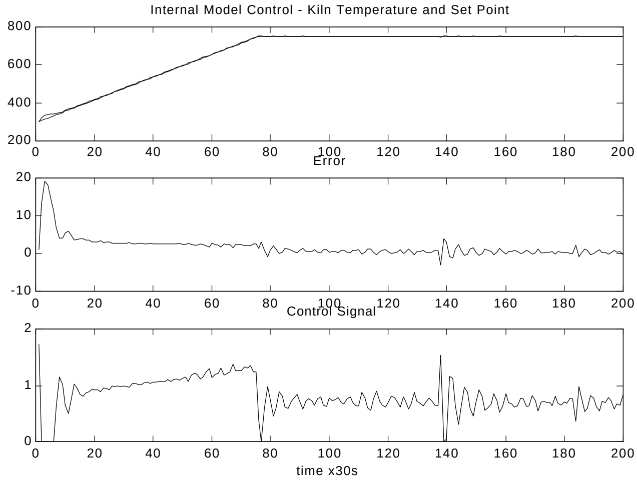

Fig. 12. Internal Model Control results.

CONTROL TYPE	TOTAL	POINTS 30 TO 200	POINTS 80 TO 200
DIC	9,67	4.62	1.97
AFFC	6.70	2.97	1.56
IMC	9.83	2.28	0.85

Table 1. Mean Square Error for the proposed strategies.

VI. Conclusions

The behaviour of FNNs is not the same when dealing with noisy data: training must be shorter since less accurate information can be extracted from data, higher absolute error should be expected and the final quality of the controller is not the same as without noise. Filtering and early stopping were used to deal with this problem.

The measurement noise has a random characteristic, but for the present system its effect is more severe when building the inverse model because a small difference in a sample of the temperature measurement would require an important change in the input.

AFFC was implemented using an existing PI controller tuned manually. The interest here is to show that even with a non-optimised controller AFFC performs well.

DIControl and AFFC strategies achieved interesting results and therefore the conclusion that the quality of the inverse model is good can be extracted. The results for IMC hold the best performance except for the first values where the lack of data lower than 300ºC seems to be more harmful for this type of control.

All the control strategies have an interesting side:

- Direct Inverse Control is the simplest strategy to implement.
- Additive Feed Forward Control's most important utility is to allow for improving an existing control loop without needing for stopping normal operation.
- Internal Model Control forms a more robust control loop an thus presents better results from Mean Square Error point of view.

Internal Model Control strategy forms a very stable loop that as can be seen from the present example performs better than strategies that don't include feedback. IMC presents a handicap though for hardware implementations where space is an important factor: two models instead of one have to be implemented.

Future work will concern implementing the air/oxygen control loop, implementing the controller in low cost micro controllers and study other possibilities for hardware implementation of the controllers.

VII. References

[1] Karl Åshtröm and Björn Wittenmark, *Computer Controlled Systems*, Prentice Hall 1997.

[2] G. Cybenko, "Approximation by Superposition of a Sigmoidal Function, Mathematics of Control, *Signals and Systems*, 2 pp.492-499,1989.

[3] M. Nø rgaard, Neural Network System Identification Toolbox for MATLAB*, Technical Report*,1996.

[4] M. Nørgaard, Neural Network Control Toolbox for MATLAB*, Technical Report*,1996.

[5] K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, "Neural Networks for Control Systems-A Survey", *Automatica*, vol.28, nº6, pp1083-1112, 1992

[6] K. J. Hunt and D. Sbarbaro, "Neural Networks for Nonlinear Internal Model Control", *IEE Proceedings-D*, vol.138, no.5, pp.431-438, 1991.

[7] O. Sø rensen, Neural Networks in Control Applications *PhD Thesis, Department of Control Engineering, Institute of Electronic Systems, Aalborg University*, Denmark, 1994.

[8] G. Lightbody and G. W. Irwin, "Nonlinear Control Structures Based on Embedded Neural System Models", *IEEE transactions on Neural Networks*, vol.8, no.3, 1997.

[9] M. Minsky and S. Papert, "Perceptrons", *Cambridge, MA: MIT Press*,1969.

[10] B. Hassibi and D. G. Stork, "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon", *Proceedings of NIPS 5,164, San Mateo, California*, 164:172, 1993.

[11] Y. Le Cun, J.S. Denker and S.A. Solla, "Optimal Brain Damage, Advances in Neural Information Processing Systems", *Denver 1989, Ed. D.Touretzsky, Morgan Kaufmann*, 598:605, 1990.

[12] M. W. Pedersen and L.K. Hansen, "Recurrent Networks: Second Order Properties and Pruning", *Proceedings of Neural Information Processing Systems 7,* 673:680, 1994.

[13] R. T. Shen, "Optimal Control of Nonlinear Systems with Neural Networks". *PhD Thesis, Stanford University*, 1998.

[14] J. Sjöberg and L. Ljung, "Overtraining, Regularization and Searching for minimum in Neural Networks", *Preprint IFAC Symp. on Adaptive Systems in Control and Signal Processing, Grenoble*, France, 669:674, 1992.

[15] F. M. Dias and A. M. Mota, "Additive FeedForward Control of a Kiln using Neural Networks", *Modelling, Identification and Control MIC2001, Innsbruck Austria,* 2001.

[16] M. Nørgaard, "System Identification and Control with Neural Networks". *PhD Thesis, Department of Automation, Technical University of Denmark*, 1996.

[17] H. Andersen, "The Controller Output Error Method", *PhD Thesis, Department of Computer Science and Electrical Engineering, University of Queensland, Australia*, 1998.