DIRECT INVERSE CONTROL OF A KILN

Fernando Morgado Dias[1], Alexandre Manuel Mota[2]

[1]Escola Superior de Tecnologia de Setúbal, Departamento de Engenharia Electrotécnica,Campus do IPS, Estefanilha, 2914-508 Setúbal, Portugal
Tel: +351 65 790000, Fax:+351 65 721869, Email: fdias@est.ips.pt
[2]Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, 3810 Aveiro, Portugal
Tel: +351 34 370383, Fax:+351 34 381128, Email: alex@det.ua.pt

Abstract: This paper presents an application of Neural Networks to the control of a real system with measurement noise. The system is a kiln and the steps taken to arrive at direct and inverse models are described. The results of controlling the kiln with a Direct Inverse Control strategy are presented.

Keywords: Neural Networks, Feedforward Networks, Modelling, Direct Inverse Control, Measurement Noise.

1. INTRODUCTION

The present paper describes the modelling and control of a reduced scale prototype kiln under measurement noise.

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 two loops for temperature control, and for air/oxygen ratio control.

The kiln is heated by an electrical resistor driven by a power controller and the temperature is measured by a B type thermocouple. The sensor and the actuator are connected to a Hewlett Packard HP34970A Data Logger that supplies real-time data to **MATLAB** using the RS232C serial line. The Data Logger though a helpful tool limits the measurement to temperatures superior to 300ºC and the thermocouple introduces measurement noise which makes identification more complex. This approach allows the use of the entire **MATLAB** powerful environment together with real-time capability (Mota, et al., 1998).

The kiln is completely closed as can be seen on figure 1, which is an outside view. This kiln operates around 750ºC having as a superior limit of operation 1000ºC.

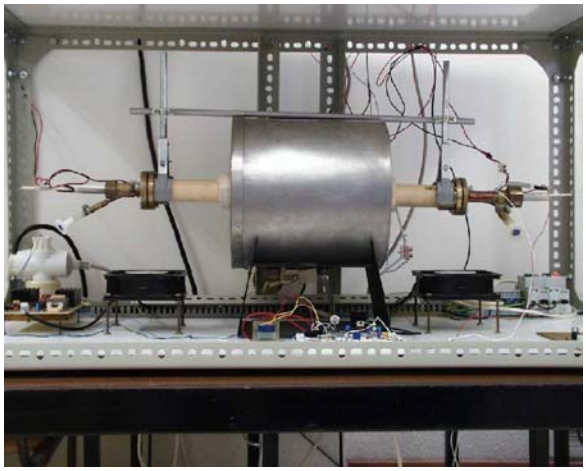

Fig. 1. Picture of the kiln and electronics.

A schematic view of the kiln can be seen in figure 2.

Considering that the kiln's behaviour is nonlinear, Neural Networks were considered to be an interesting approach to control the temperature.

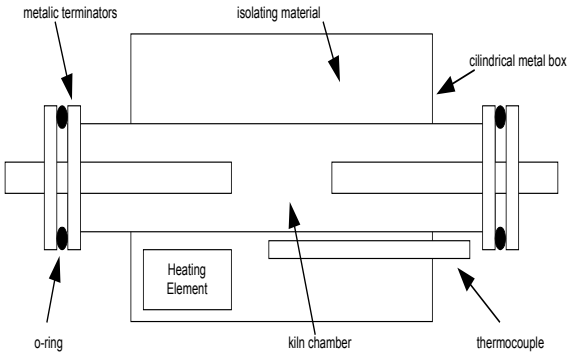

Fig. 2. Schematic view of the kiln.

2. FEEDFORWARD NEURAL NETWORKS

A Feedforward Neural Network (FNN) is a layered structure, which can include non-linearity. The basic element of a FNN is the neuron that is shown schematically in figure 3.

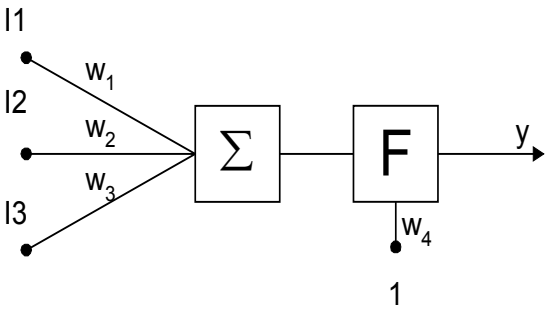

Fig. 3. Neuron structure.

The neuron implements the general equation:

$$y = F(\sum_{i=1}^{n} Ii.wi) \qquad (1)$$

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.

FNN are a category within the Artificial Neural Networks (ANN). As the name itself says, in FNN the signal flows only from input to output while in ANN there are other ways in which the network can be connected such as using feedback and lateral connections. A typical structure can be seen in figure 4.

The Multi Input Single Output FNN in figure 4 implements the following general equation:

$$y = F_1(\sum_{j=1}^{nh} w'_{j1} f_j(\sum_{l=1}^{nI} w_{lj} I_l)) \qquad (2)$$

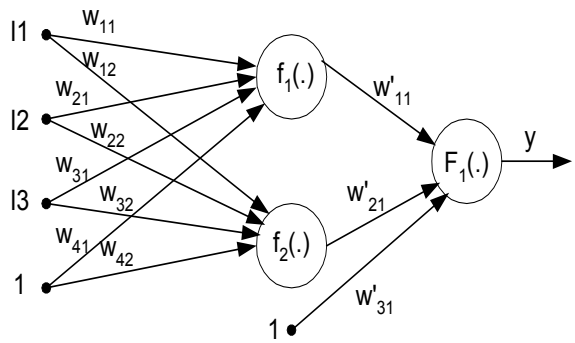

Fig. 4. Feedforward Neural Network structure.

Artificial Neural Networks have been a field of fast development over the last two decades mainly because of their ability to model nonlinear plants.

Since ANN have been proven to be universal approximators by (Cybenko, 1989) many algorithms have been developed in order to achieve control of nonlinear plants, among which it is worthwhile to point out the well know Backpropagation and the most effective Levenberg-Marquardt (Hagan, et al., 1996).

The evaluation of the state of art of Neural Networks (NN) for Control pointed out that most of the work done until now has concerned simulated systems and not real data as are examples (Nørgaard, 1996a; Ke et al.; Andersen 1998). In some cases noise is even simulated to approximate the simulations to the real case (Sørensen, 1994), but the fact is that it somehow remains to be proved the usefulness of NN for Control in real systems under noise conditions.

3. IDENTIFICATION AND CONTROL STRUCTURES

Before reporting the work that has been done, it is important to introduce the structures used for identification and control with NN to improve understanding of the solutions implemented.

3.1 Identification

Forward Modelling. The procedure for training a forward model consists of placing a NN in parallel with the plant as depicted in figure 5. Here the error resulting from the mismatch between plant and model is used to change the weights of the NN through an appropriate algorithm.

Inverse Modelling. Inverse models are very important since they are part of many control structures.

The simplest approach is the direct method which is closely related to forward modelling. A block

diagram of this type of training can be seen in figure 6.

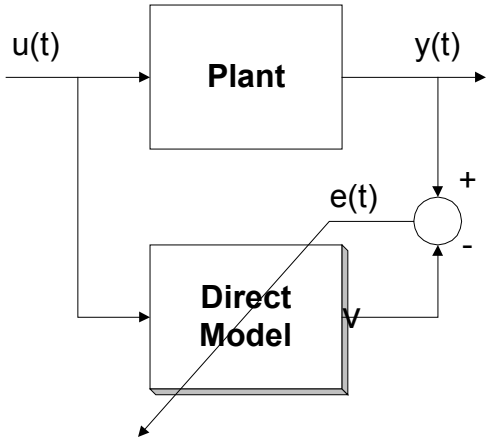

Fig. 5. Structure for direct model training.

However simple, this approach has some drawbacks (Hunt, 1992):

- The learning procedure is not goal directed. As the model is not trained in the same situation in which it will be used after training, the structure of training is said not to be goal directed.

- In situations where the mapping is not 1:1 an incorrect inverse can be obtained.

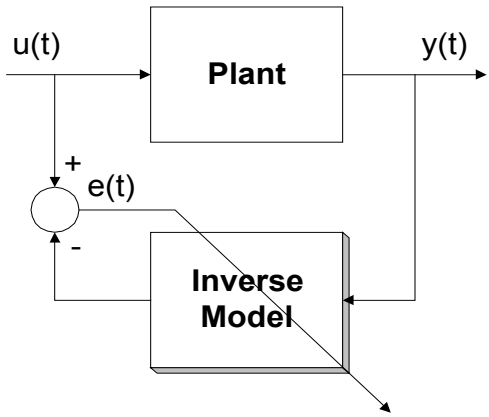

Fig. 6. Structure for inverse model training.

To avoid these problems another way of training inverse models is present in the literature (Nørgaard, 1996a; Hunt, 1992): Specialized inverse model training. The models and plant are connected according to figure 7.

The inverse model is connected in series with the plant, but since usually the internal states of the Plant are unknown and do not allow performing the necessary calculations to report the error (between plant output and desired output) to the output of the inverse model, a direct model is placed in parallel with plant.

This structure is supposed to overcome the problems mentioned for the previous type of training because the network is trained in a situation similar to the one that the NN will assume in a control situation.

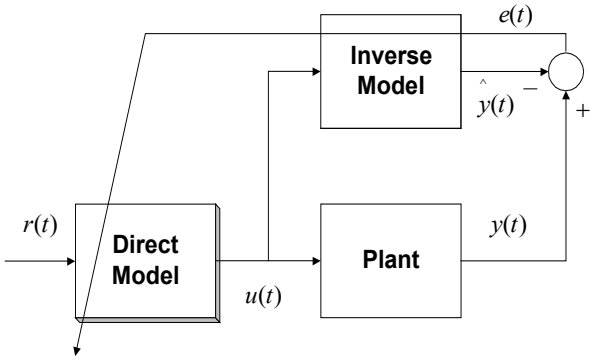

Fig. 7. Structure for specialized inverse model training.

3.2 *Control Structures*

Many control structures have been proposed, most of which were adapted from linear control. Resuming state of art for these structures is out of scope for the present article, thus only direct inverse control will be introduced.

Direct inverse control. 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 8.

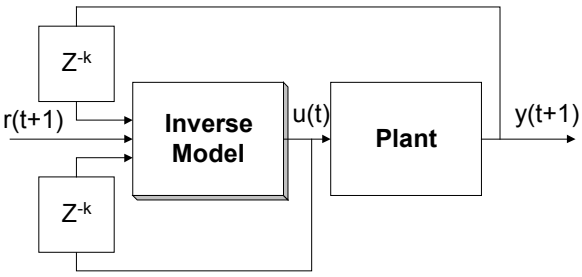

Fig. 8. Structure for direct inverse control. If the inverse model is of order superior to one, delayed samples of control and output are feedback to the inverse model input.

4. THE IDENTIFICATION PROBLEM

The first issue to deal with is which way to use to collect data. Open loop data is usually used, though in some situations closed loop data might be preferred since the plant is kept inside the range in which it is intended to operate (Nørgaard, 1996a). The latter solution brings the problem of correlation between input and output, which might lead to the loss of system identifiability (Söderström, 1989).

The data used for the identification was open loop data in which the control signal was produced in order to cover the entire operating range.

Since the kiln is nonlinear (notice that the heating is much faster than cooling since the kiln is closed, the inside temperature is protected by the metal cylindrical box and the gap between the metal box and the kiln chamber is filled with isolating material), one of the concerns was to have an amplitude spectrum of the input signal wide enough to allow for correct nonlinear identification (Sørensen, 1994).

The sampling period was fixed to 30s after some trial-and-error tests.

During the identification and control tasks the NNSYSID (Nørgaard, 1996b) and NNCTRL (Nørgaard, 1996c) toolboxes for **MATLAB** were used.

5. STRUCTURE SELECTION AND NETWORK TRAINING

After the identification task, input and output sets of data are produced (which later will be split in to training and test sets). The NN training is accomplished with scaled sets.

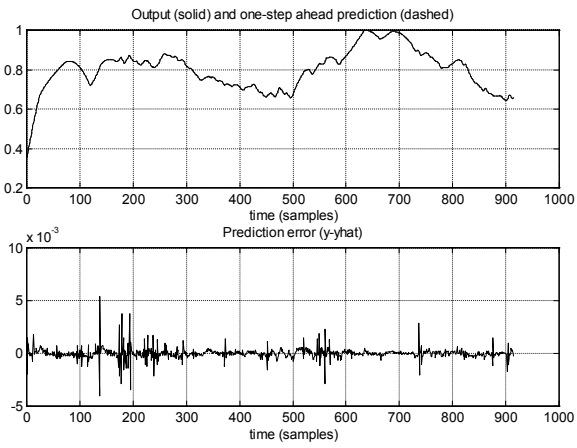

Fig. 9. Output of forward model and training set (upper half) and Error (lower half).

In order to evaluate the quality of the models obtained both a direct and inverse model were produced. This allowed performing closed loop simulation before controlling the kiln directly.

An ARX model was chosen and the Levenberg-Marquardt algorithm was used for training the NN because of its fastest convergence. Several architectures of the NN were tested and the best results were obtained using four neurons on the hidden layer of the direct model and five neurons on the hidden layer of inverse model, both models have a linear output neuron.

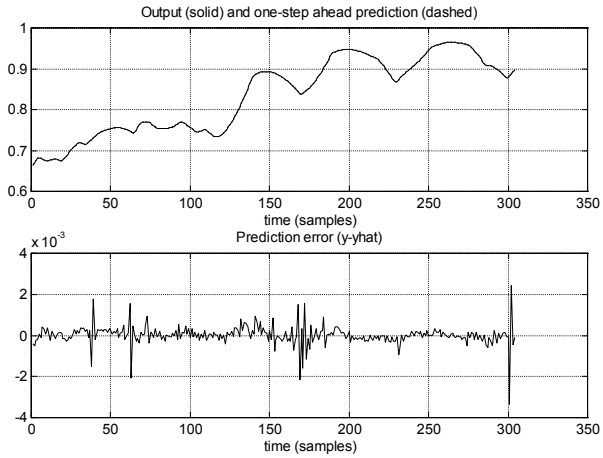

Fig. 10. Output of forward model and training set (upper half) and Error (lower half).

Forward Model responses to training and test sets can be seen in figures 9 and 10 respectively.

Data has been divided into train and test sets in order to be possible to train the NN on one set and evaluate the result of the training over the other.

As can be seen the forward model follows very closely both the train and test sets.

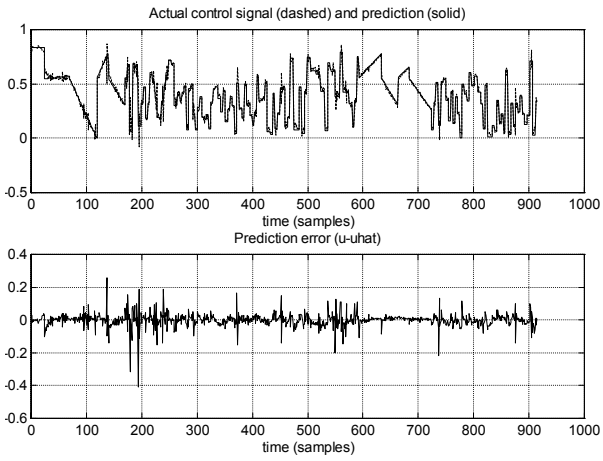

Fig. 11. Output of inverse model and training set (upper half) and Error (lower half).

Inverse model accuracy is not as good as direct model, mainly because of the kind of data used (open loop data) which in this case shows that not all the data used is inside of the operating range. Responses of the inverse model to train and test sets can be seen in figures 11 and 12 respectively.

Closed loop simulation of Direct Inverse Control can be seen in figure 13.

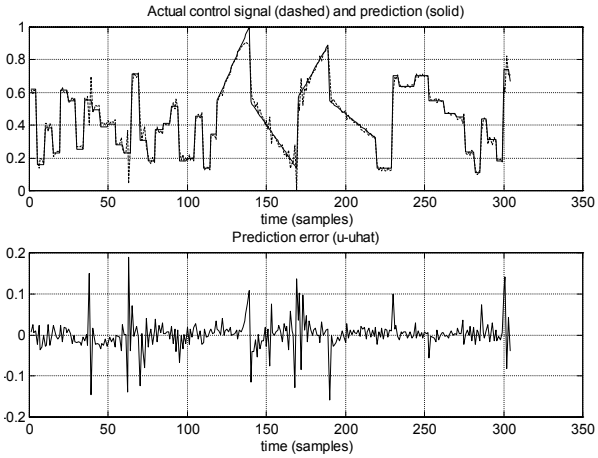

Fig. 12. Output of inverse model and test set (upper half) and Error (lower half).

Simulation of direct inverse control using both train and test sets shows that the model have captured enough information about the system exception made to the initial heating phase where data was only available starting from 300ºC.

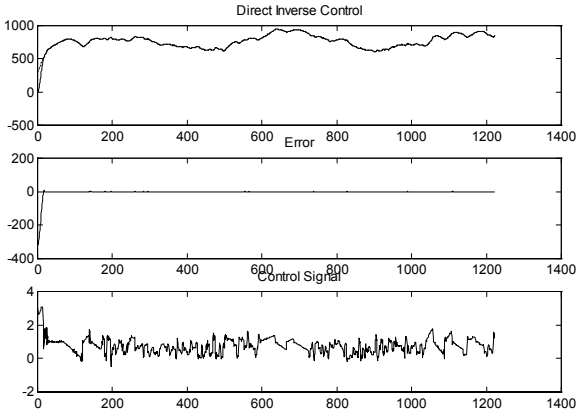

Fig. 13. Direct inverse control simulation: temperature (reference signal and actual temperature), error and control signal respectively.

6. THE REAL TIME CONTROL ACTION

When the accuracy of the models was considered sufficient (this was also evaluated from closed loop simulations as depicted in figure 13) the model was tested on the real kiln using the Data Logger connected to MATLAB to monitor the temperature and interface the control signal to be applied.

The results can be seen on figures 14 and 15 for different references.

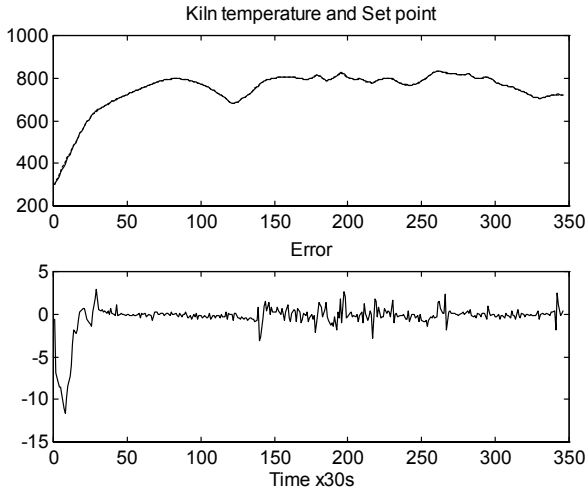

Fig. 14. Direct inverse control using as reference the training signal.

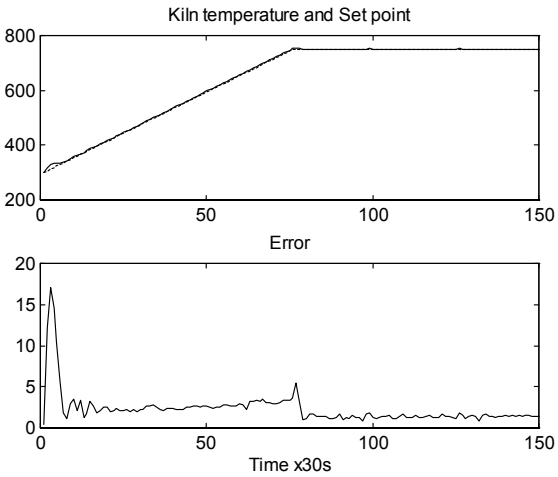

Fig. 15. Direct inverse control using a new reference.

7. CONCLUSIONS

As can be seen from figures 9 to 12 both direct and inverse model match very closely the data used for training and for testing. In most of the figures it is even difficult to distinguish both signals. The good performance showed in the Direct Inverse Control simulation is confirmed in the real control of the kiln using two different references, proving that the NN has generalisation capabilities. It can though be seen that the lack of temperature data lower than 300ºC results in some difficulties in the closed loop control.

Future work will concern implementing other control strategies such as Internal Model Control and Feedforward Control.

REFERENCES

Andersen, H. (1998), The Controller Output Error Method, *PhD Thesis, Department of Computer Science and Electrical Engineering, University of Queensland, Australia*.

Cybenko, G. (1989) Approximation by Superposition of a Sigmoidal Function, *Mathematics of Control, Signals and systems*, **2** pp.492-499.

Hagan, Demuth, Beale (1996), Neural Network Design, *PWS Publishing Company.*

Hunt, K. J., D. Sbarbaro, R. Zbikowski and P. J. Gawthrop (1992), Neural Networks for Control Systems-A Survey, *Automatica,* **28**, pp.1083-1112.

Ke, L., R. L. Tokar, B. McVey, An Integrated Architecture of Adaptive Neural Network Control for Dynamic Systems, *Neural Information Processing Systems 7*, pp. 1031-1039.

Mota, A. et al, (1998) Using SIMULINK in Real Time Control, *Proceedings of 3rd Portuguese Conference on Automatic Control*, pp.

Nørgaard, M. (1996)a. System Identification and Control with Neural Networks. *PhD Thesis, Department of Automation, Technical University of Denmark.*

Nørgaard, M. (1996)b. Neural Network System Identification Toolbox for MATLAB, *Technical Report*.

Nørgaard, M. (1996)c. Neural Network Control Toolbox for MATLAB, *Technical Report*.

Söderström, T., P. Stoica, (1989) System Identification, *Prentice-Hall*.

Sørensen, O. (1994). Neural Networks in Control Applications. *PhD Thesis, Department of Control Engineering, Institute of Electronic Systems, Aalborg University, Denmark.*