A Comparison between a PID and Internal Model Control using Neural Networks

Fernando Morgado Dias
Departamento de Engenharia Electrotécnica,
Escola Superior de Tecnologia do Instituto Politécnico de Setúbal,
Campus do IPS, Estefanilha, 2914-508 Setúbal, Portugal
Tel: +351 265 790000, Fax:+351 265 721869, Email: fmdias@est.ips.pt

and

Alexandre Manuel Mota
Departamento de Electrónica e Telecomunicações,
Universidade de Aveiro, 3810 Aveiro, Portugal
Tel: +351 234 370383, Fax:+351 234 381128, Email: alex@det.ua.pt

Abstract

Nonlinearity and noise have always been a major problem for control systems. The majority of classical controllers fail to achieve good control performance under these conditions. Proportional Integral Differential controllers constitute on the other hand a rule of thumb for controlling systems almost independently of the mentioned problems.

The present article describes an application of temperature control for a reduced scale prototype kiln where two different solutions are proposed: an Internal Model Control using Neural Networks and a PID tuned using a Genetic Algorithm with a Neural Network model of the plant.

Both solutions lead to very good control performance though the PID optimisation is dependent of the reference used while the Neural Network due to its generalization capabilities is independent of the signal used for training as long as the signal has enough information about the system being modelled.

The availability of the Neural Network model is crucial for both solutions: for implementing the stable feedback control loop of Internal Model Control and for optimising the PID parameters with a Genetic Algorithm.

Keywords: PID, Internal Model Control, Feedforward Neural Networks, Measurement Noise and Genetic Algorithm.

1. Introduction

Nonlinearity and noise have always been a major problem in control systems. The majority of classical controllers fail to achieve good control performance under these conditions. Proportional Integral Differential controllers constitute on the other hand a rule of thumb for controlling systems almost independently of the mentioned problems.

The present article describes an application of temperature control for a reduced scale prototype kiln where two different solutions are proposed.

The first one uses a PID controller whose parameters are tuned using a Genetic Algorithm (GA). The fitness measure is given by testing the PID in a Neural Network (NN) model of the kiln.

The second solution is the classical Internal Model Control implemented using Neural Networks.

The field of NNs has known different stages of development. One of the most important steps was achieved when Cybenko [1] proved that they could be used as universal approximators. Other important steps were taken by developing suitable algorithms for training NN like backpropagation and the adaptation of the Levenberg-Marquardt algorithm to use with NN.

A reasonable slice of this work has been done 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.

Genetic Algorithm (GA) or Genetic searching algorithm is a function optimisation technique based on the principles of evolutionary genetics and the natural selection process [2] after the pioneering work of Holland [3].

The original goal was to study the adaptation phenomena in nature, but his work was later used for optimisation techniques based in a fitness function, corresponding to the survival of the fittest principle.

Since the initial work many new operators have been proposed and many improvements were introduced but crossover, mutation and elitism are solutions that are present in almost any aplication of GA for optimisation. Crossover originates a new member for the population, by a process of mixing genetic information from both parents and raises the question of how to select the parents for a fastest growing of the fitness of the population. Among many other solutions, the selection can be done with the roulette method, by tournament, random and elitist [3] [4].

Mutation is a process by which a percentage of the genes are selected in a random fashion and changed. Elitism corresponds to keeping the best members of the population to the next generation to guarantee that there is a continuous maximization of the fitness function.

GA optimisation is especially useful when there is no deterministic solution for the problem or the range of solutions is too wide for an exhaustive search and local minimum can be acceptable. It is also important that this is a global optimisation method.

2. The Plant

The plant 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.

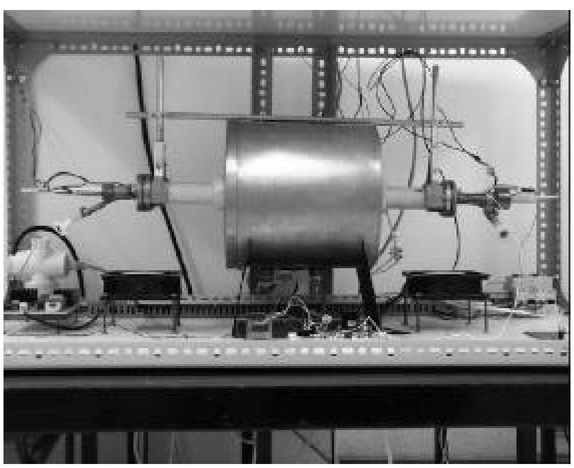

Fig. 1. Picture of the kiln and electronics.

An electrical resistor driven by a power controller heats the kiln 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.

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

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

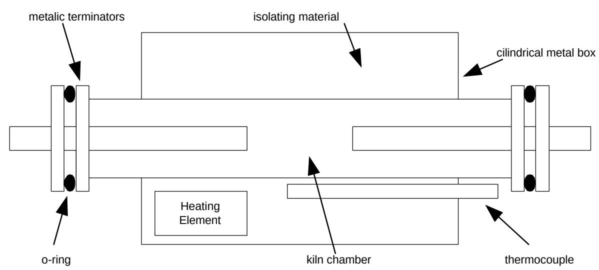

Fig. 2. Schematic view of the kiln.

3. Identification

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 [7]. The latter solution brings the problem of correlation between input and output, which might lead to the loss of system identifiability [10].

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, one of the concerns was to have an amplitude spectrum of the input signal wide enough to allow for correct nonlinear identification[10].

The sampling period used was of 30s.

Direct and Inverse models were trained according to details given in [11].

During the identification and control tasks the NNSYSID [8] and NNCTRL [9] toolboxes for MATLAB were used.

4. The Internal Model Control Loop using Neural Networks

Internal Model Control (IMC) is a structure that allows the error feedback to reflect the effect of disturbance and plant mismodelling. In fact it can be shown that a good match between forward and inverse models is enough to have good control and with this structure disturbance's influence is also reduced.

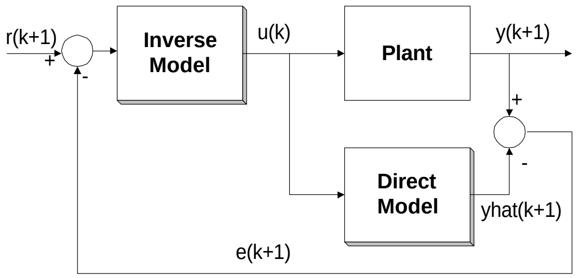

Fig. 3. Structure for Internal Model Control. The signal r(k) is the reference, u(k) the control signal, y(k) the output signal, yhat(k) the estimate of the output and e(k) the error between the output and the estimate.

instead of:

$$u(k) = g\begin{bmatrix} r(k+1), y(k),..., y(k-n_y+1) \\ u(k-t_d),..., u(k-n_u-t_d+1) \end{bmatrix}$$

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 IMC structure resulting from the first equation can be seen in figure 4.
In the present work IMC structures were tested with and without these adaptations. The direct translation of the classical IMC worked like Direct Inverse Control and in the presence of disturbances was unable to recover, while with these changes IMC works properly as can be seen in section 7.

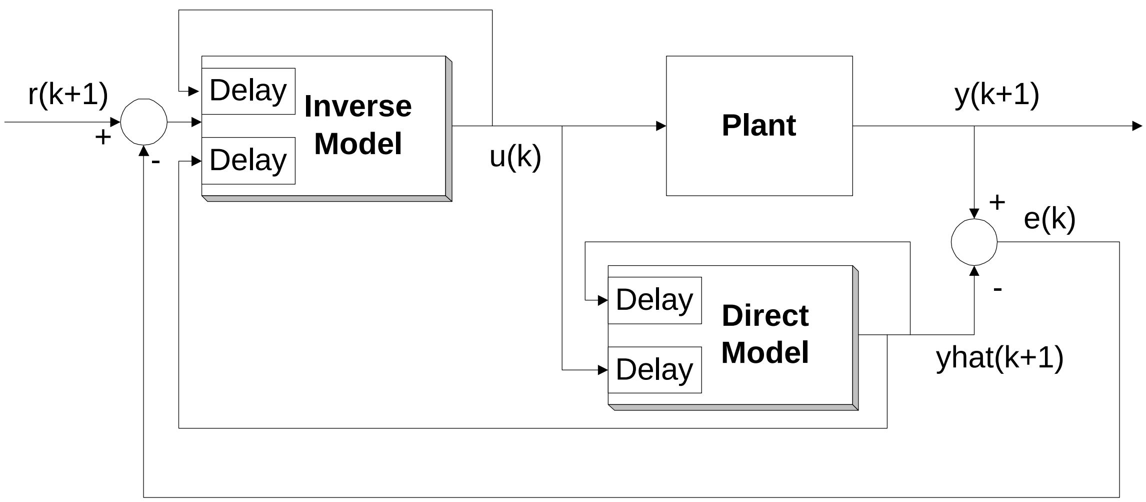

Fig. 4. Internal Model Control structure with detail of the implementation of inverse and direct model.

Adapting Internal Model Control Structure 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.
Some of structures adopted from classical control can be used directly with NN models, but IMC needs some refinements to work properly [12].
The good match between forward and inverse models, referred above translates to having the forward model outputs feedback to the input of the inverse and direct model instead of the outputs of the plant. This means that the inverse model will implement the following equation:

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

5. Proportional Integral Differential Controller Optimisation

The optimisation of the Proportional Integral Differential (PID) controller was done through the use of a genetic algorithm that uses a NN model of the plant. For each individual in the population used in the genetic search a control loop is implemented using the parameters for the PID and the model of the plant. Provided a reference for the control loop, the fitness of the solution is the Mean Square Error obtained between the output and the reference. The fittest solution is the one with the lower fitness value.
The algorithm implemented includes Crossover, Mutation and Elitism but several changes were introduced.
Using a small population, a strong elitism of 25% is assumed, crossover of one site splicing is performed.

and all the members are subjected to mutation except the elites.

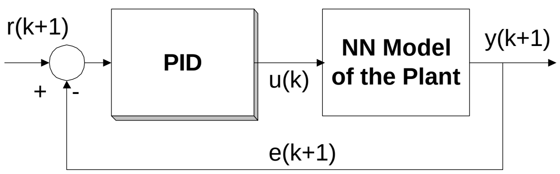

Fig. 5. Control loop implemented during the genetic optimisation of the PID controller.

The mutation operator is a binary mask generated randomly according to a selected rate that is superposed to the existing binary codification of the population changing some of the bits according to a predefined rate.

Improvements Introduced in the Algorithm

Two changes were introduced in the algorithm in order to improve the convergence. The first one regards the crossover: 50% of the population, including the elites is randomly selected with equal opportunity. The second one concerns picking the lower bits of the weighs' codification in the fittest members and change them adding or subtracting a small binary amount in order to verify the neighbourhood for a solution better than the present one. The backpropagation algorithm itself inspired this improvement since checking the neighbourhood of the present solution might lead to a lower value of the error that is difficult to obtain with the random character of the genetic search. These refinements though similar in some points to the ones presented in [2] were developed independently.

Due to the reduced number of parameters of the PID controller and the improvements introduced, in few generations the solutions converge and the optimisation is terminated.

It is worth to note that the integral gain ki in the several optimisations performed is always zero or a very small value. This can have a physical interpretation: because of the integral behaviour of the kiln, the integral action is not needed.

6. Modelization of the Plant

After the identification task, input and output sets of data are produced and will be split in to training and test sets. Both sets will be scaled removing their mean and divided by their standard deviation. The training set is used for teaching the NN while the test set serves to test the quality of the model obtained.

An Auto-Regressive with eXogenous input (ARX) model was chosen and the Levenberg-Marquardt algorithm was used for training the NN because of its fastest convergence. 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.

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

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.

In figures 6 and 7 the response of forward and inverse models to the test sets is presented.

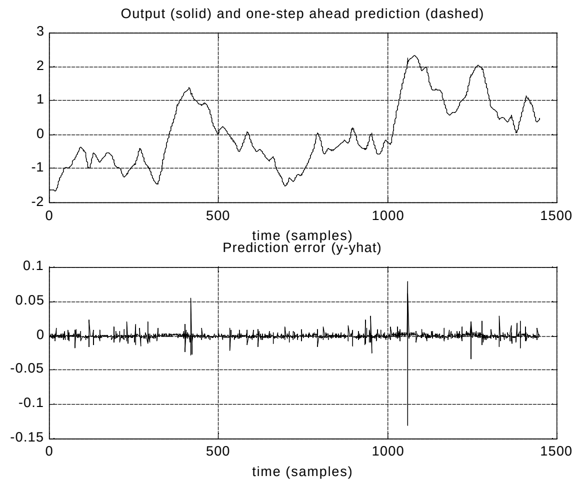

Fig. 6. Forward model's response to test sequence and error between output and prediction.

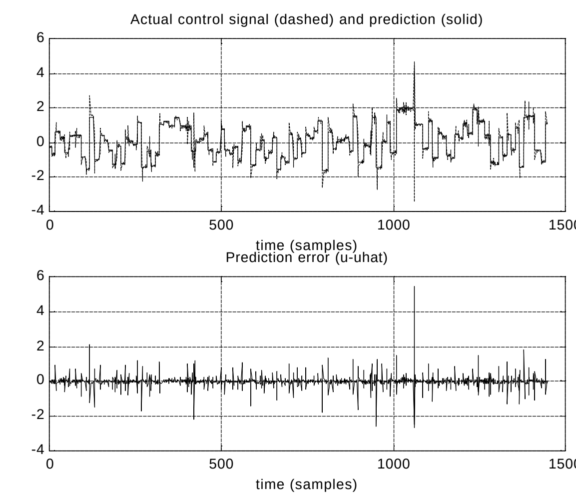

Fig. 7. Inverse model's response to test sequence and error between control signal and prediction.

The quality of the inverse model is inferior to the direct model. This is mainly because of the use of filtering, which in some situations diminishes the connection between input and output signal. This situation is more

severe to the inverse model since to the direct model the kiln itself acts as a low pass filter.
The error is small, except for some "glitches" that appear in situations where the filtering makes it difficult for the models to "understand" the relation between input and output. With this procedure a detuned model is obtained. This detuned model is better than the model obtained directly from the noisy data and both forward and inverse models detuned form a good match that is necessary for the IMC to work properly.

7. The Real Time Control Action

In this stage the solutions prepared to control the temperature loop are tested using the plant and the results are collected.
For the experiences performed with the PID controller different ranges and set points were tested. Table 1 summarizes the results and the best result achieved, corresponding to experience 5, is shown in figure 8.

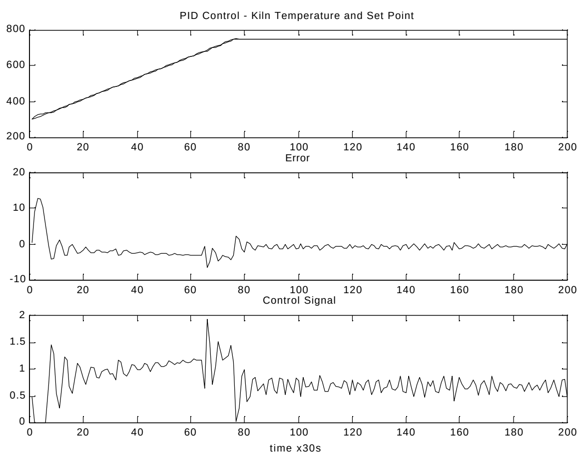

Fig. 8. Best performance obtained with PID Control.

Comparing the results presented in table 1 there are some facts that are worth to point out: Ki is always zero or very small, which confirms the integral characteristic of the kiln, the PID optimisation is dependent of the reference used during the optimisation process.
In table 2 a comparison between the results obtained with the PID and the IMC is summarized.

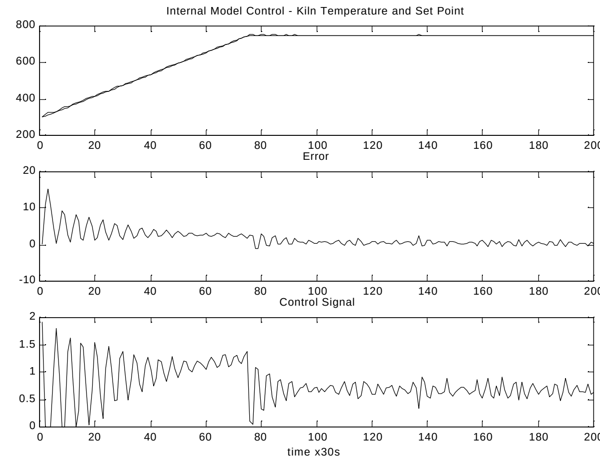

Fig. 9. Internal Model Control results.

Control strategy	MSE (total)	MSE (points 80 to 200)
IMC	8.07	0.74
PID	5.87	0.86

Table 2.- Comparison between the two control strategies.

8. Conclusions

The use of Neural Networks is crucial to both solutions implemented: Internal Model Control is implemented using forward and inverse models of the plant and the Proportional Integral Differential controller is optimised using a Neural Network model of the plant. This last solution could not be implemented directly using the plant because the temperature could go outside the safe operating range.
From the results shown in table 1 it can be concluded that the PID optimisation is dependent on the "training signal" which is in fact the reference used for the optimisation. On the contrary NN performance is quite independent of the training signal, requiring only that the signal used has enough information about the system and refers to the operating range of the plant as much as possible.
Comparing the results obtained from the two control strategies presented in table 2 it can be seen that IMC shows more difficulties in the initial phase where the NN controller needs past information about inputs and outputs that is not available, but after this stage performs better than the PID controller.
Both situations show the utility of FNN in modelling and control.

EXPERIENCE NUMBER	RANGE OF PARAMETERS	PARAMETER VALUE	TRAINING SIGNAL	SET POINT TYPE	MSE .
1	0 to 10	Ki=0.05 Kp=10 Td=9.96	ramp+stable	ramp+stable	47.86
2	0 to 10	Ki=0 Kp=10 Td=10	NNID	ramp+stable	91.84
3	0 to 25	Ki=0.004 Kp=25 Td=25	ramp+stable	ramp+stable	13.83
4	0 to 25	Ki=0 Kp=25 Td=25	NNID	ramp+stable	17.28
5	0 to 100	Ki=0 Kp=65.49 Td=5.88	ramp+stable	ramp+stable	5.87
6	0 to 100	Ki=0 Kp=68.63 Td=5.49	NNID	ramp+stable	6.34

Table 1. Control essays with the plant for different PID optimizations. The NNID signal is the signal used for training the Neural Network and MSE is the Mean Squared Error.

References

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

[2] J.-S. Yang and M. L. West, "*A Case Study of PID Controller tuning by Genetic Algorithm*", Proceedings of the IASTED International Conference on Modelling Identification and Control, Innsbruck, 2001.

[3]J. Holland, "*Adaptation in Natural and Artificial Systems*", University of Michigan Press, Ann Arbor, MI, 1975; MIT Press, Cambridge, MA, 1992

[4] G. Bartlett, "*Genie a first GA*", published in "*Practical Handbook of Genetic Algorithms, Applications*", edited by Lance Chambers, Volume I, pp31-56, CRC Press, 1995.

[5] J.-S. R. Jang, C.-T. Sun and E. Mizutani, "*Neuro-Fuzzy and Soft Computing*", Prentice Hall 1997.

[6] S. T. Welstead, "*Neural Network and Fuzzy Logic Applications in C/C++*", Wiley, 1994.

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

[8] M. Nørgaard,"*Neural Network System Identification Toolbox for MATLAB*", Technical Report, Technical University of Denmark, 1996.

[9] M. Nørgaard, "*Neural Network Control Toolbox for MATLAB*", Technical Report, Technical University of Denmark, 1996.

[10] O. Sørensen, "*Neural Networks in Control Applications*" PhD Thesis, Department of Control Engineering, Institute of Electronic Systems, Aalborg University, Denmark, 1994.

[11] F.M.Dias and A.M.Mota, "*Direct Inverse Control of a Kiln*", Proceedings of the 4th Portuguese Conference on Automatic Control, Guimarães, 2000.

[12] 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.

[13] 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

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