Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Application of an artificial neural network to on-line control of a process

D. Williams,[a] J.T. Evans,[a] J.B. Gomm[a] &
P.J.G. Lisboa[b]

[a] *Control Systems Research Group, School of Electrical and Electronic Engineering, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK*

[b] *Department of Electronic Engineering and Electronics, University of Liverpool, P.O. Box 147, Liverpool L69 3BX, UK*

ABSTRACT

Although the development of linear control theory is well established, real industrial processes contain significant non-linearities that give limited credence to the optimal performance of controllers over a wide operational region. Consequently, the development of techniques that enable the design of a controller suitable for operation with a non-linear process would be beneficial.

This paper describes the development and implementation of an on-line, one-step-ahead, optimal predictive controller incorporating a neural network model of the process. The scheme is based on a Multi-Layered Perceptron neural network as a modelling tool for a real non-linear, dual tank, liquid level process. The model validation techniques are described as well as the choice of network structure and topology. The ability of the trained neural network to represent both a simulation of the process, modelled from first principles, and the actual process is investigated. The implementation, of the optimal control algorithm, to both the simulation and the real process are described. Results are presented to illustrate the steady-state and transient performance of the control scheme.

INTRODUCTION

The development of linear controllers for linear systems has been established for many years. However, the control of more realistic non-linear systems still provides engineers with challenging problems. The approach of linearising a model of the process about its operating point and developing a suitable linear controller is of limited applicability. The alternative approach of developing a

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

non-linear controller for operation over the non-linear regions has not been established to an acceptable degree. The potential development of a non-linear model, using a neural network, and the consequent design of a suitable controller provides a possible way forward for the evolution of controllers for the future. This paper describes the development and on-line implementation of a standard predictive controller employing a neural network model to describe the non-linear dynamics of a laboratory process.

Neural Networks are not new and have been around since the mid 1950's but it is only recently, due to the development of improved training algorithms and electronic hardware for implementation, that interest has been rejuvenated. Their application into AI areas, vision systems, speech and sound systems has recently broadened into the field of control systems. The inherent ability of neural networks to model non-linearity, to generalise from the initial data set, to overcome noisy signals and to be robust to data inconsistencies raises interest in their development for modelling and control of real industrial processes.

A number of approaches are possible for controller design, depending on the type of process. Real processes may vary with time and contain non-linearities. The design of controllers for linear systems is well established and documented. However, the design of suitable controllers is not so well established for non-linear and time-varying processes and is currently receiving interest. In this paper, the objective is to design, develop and implement a controller, incorporating a neural network model, specifically for a non-linear process.

One approach, that can be used, is to develop a linear model of the process about its steady-state operating conditions, the process representation only being valid for a limited range outside its steady-state operating point. A number of different linear control algorithms can be developed using this linear model. However if the operating point was changed and the non-linearities were a factor, then the design process would again be required to redesign or retune a controller to meet a required process specification. An alternative strategy would be to design a specific non-linear controller for the process and its range of operation. A preferred solution would be a single process representation that accurately models the process dynamics over the entire region of process operation. This would enable a single control strategy to be implemented. It is now widely accepted that a non-linear process may be accurately modelled by an artificial neural network (ANN). The availability of such a model permits the implementation of a number of control strategies. A main disadvantage of a neural network representation is its lack of physical characteristics and in this respect has similarities with process black-box input-output modelling approaches. One of the significant advantages in using neural networks is that

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

the model may be achieved without deep process knowledge and can be accomplished with the availability of adequate process data.

The inherent ability of neural networks to capture the non-linear dynamics of a process enables the design of a suitable controller for process operation over the defined non-linear region. A number of investigations into the combined use of control strategies in conjunction with neural networks have been recently published (e.g. Miller *et al.* [1]). The application of an Internal Model Control (IMC) strategy to the control of a simulated non-linear system, that was invertible, was reported by Hunt and Sbarbaro [2]. The IMC approach has the disadvantage of the requirement of both a forward and inverse model of the plant, the latter may not always be possible to achieve and introduces additional training requirements for a neural network. Model reference control using neural networks has also been investigated in simulation (Narendra and Parthasarathy [3]) with success, although a number of conditions were assumed. In these studies, it was decided to employ a Neural Network Predictive Controller (NN-PC) as a first attempt strategy because only one neural network representing a forward model of the process is required and the approach has some advantages over alternative strategies. The method has been successfully used in a simulation example (Willis *et al.* [4]) and benefits from not needing an inverse process model. The combination of neural network model and predictive controller has also been investigated on a simulation of a distillation column (Montague *et al.* [5]). The NN-PC algorithm is based on an iterative solution of a conventional cost function. The objective of the algorithm is to find the control signal (or process input) that minimises the cost function. The cost function embodies an integral-square-error function on the difference between required and network model output and is tempered by a weighting factor which penalises excursions of the actuator.

From the possible array of network structures, it was decided to employ a Multi-Layered Perceptron (MLP) network, trained using the standard back-error propagation (BEP) algorithm (Werbos [6], Rumelhart and McClelland [7]). This combination has been shown to be able to reliably represent a real non-linear process. Although a number of different network architectures have been investigated, the MLP has been utilised because it has been shown that these networks can represent non-linear functions provided sufficient hidden units are incorporated (Hornick *et al.* [8]). Another benefit is that the network is suitable for supervised training. In the field of process control, it is possible to obtain input/output data directly from a process or its simulation, hence providing the required data for both training and validation. The main disadvantages of the MLP and BEP algorithm are the relatively high volume of process data required for training and the long training time, compared to other structures and training algorithms, necessary to achieve a good model.

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

LIQUID-LEVEL PROCESS

The choice of process was moderated by a number of considerations. The process had to be accessible, safe and could be readily modelled by conventional techniques. Consequently, it was decided to use a relatively simple process as an initial test-bed because the process provided well-known non-linearities and could be readily simulated.

The process (Fig.1) consists of two plexiglass cylinders 1.2 m high with a cross-section of 0.0154 m^2. The fluid is pumped into the system by a 'Stuart' pump. The input flow-rate to the first tank is controlled by a pneumatic valve, which has a safety by-pass. The tank outlet restrictances can be adjusted manually. All pipework is standard 25.4 mm copper. The measurement of level is by standard DP-cell with P/I conversion for transmission to the PC. The control signal would be developed in the PC and transmitted to the pneumatic valve via a DAC and I/P converter.

The data acquisition and controller output signals are acquired and implemented through 'Blue Chip Technology' input/output interface cards in an IBM PS/2 model 30 computer. The control algorithm was implemented in the Quick Basic programming language and the sampling time employed was 36 secs.

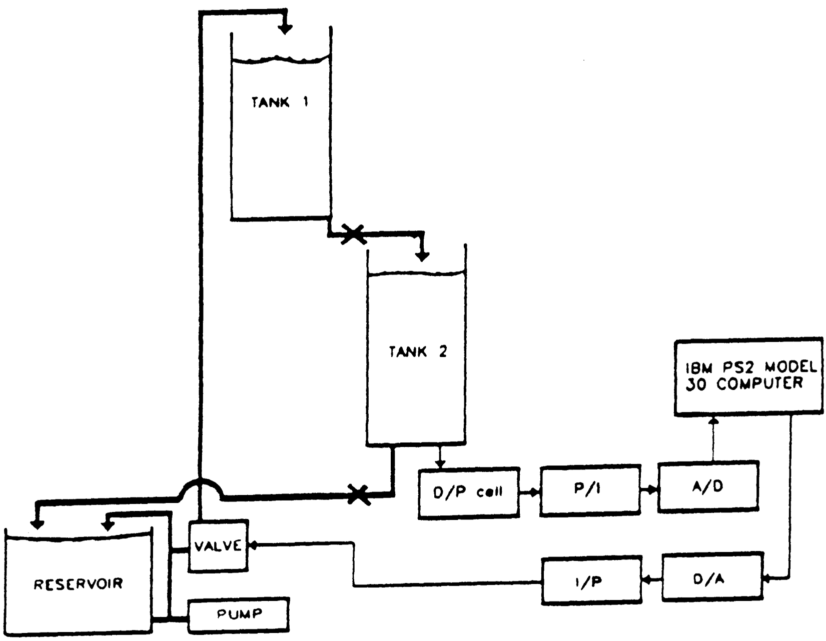


Figure 1 Process diagram

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

PROCESS SIMULATION

The on-line development of process models and controller can be time consuming and inefficient. The first step in the development of a suitable controller, was to develop a conventional mathematical model of the liquid-level process and utilise it as a test bed. Hence, it enables a thorough investigation of the principles involved prior to on-line evaluation.

The non-interacting, dual tank liquid level process (Fig.1) with non-linearities in the outlet flow rates exhibits features typical of many industrial processes, and is made more realistic by the inclusion of an unmeasured state: the height of liquid in tank 1. The non-linear differential equations describing the process are:

$$C_1\frac{dh_1}{dt}=k_v u-\frac{\sqrt{h_1}}{R_1} \tag{1}$$

$$C_2\frac{dh_2}{dt}=\frac{\sqrt{h_1}}{R_1}-\frac{\sqrt{h_2}}{R_2} \tag{2}$$

where h_1 and h_2 are the liquid levels in tanks 1 and 2 respectively, u and h_2 are the process input and output, k_v is the valve gain, C_1, C_2, are the cross-sectional areas of tanks 1 and 2 and R_1, R_2 are the outflow pipe restrictances of tank 1 and tank 2, respectively.

These equations were implemented in a continuous simulation package ACSL (Evans *et al.* [9]) which provides a range of Runge-Kutta routines to solve differential equations with non-linear features. The data set, for the development of the neural network model, was achieved by disturbing the input flow rate, via the valve input u in equation (1).

DEVELOPING THE ANN FOR PROCESS MODELLING

In the development of a neural network model, it is important to select the optimum network topology for this application and determine the parameters for quicker and more accurate training. The lack of analytical methods results in the need to undertake experiments to determine near optimum network design. This section describes the rationale and consequent development of a network suitable to represent the laboratory process.

The conventional MLP network structure with back-error propagation, using the standard sigmoidal non-linearity as the activation function for each of the

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

neurons in the hidden and output layers, has been used for the development of the model of the process. The data was pre-conditioned using the technique of spread encoding (Evans *et al.* [9,10], Lisboa [11]) which involves spreading each input data value over a prescribed number of network input nodes using a Gaussian distribution. Each value, in the distribution, is encoded between 0.1 to 0.9. A similar and reverse procedure is applied on the network output nodes to gather the distributed data and decode to the correct analogue value.

The network can be structured as either a 'predictor' or as a 'model' as illustrated in Fig.2. In this figure it can be seen that in the 'model' structure the network outputs are delayed and fed back as inputs to the network. When the network is configured as a 'predictor', then the past values of process input and output data are used for network inputs. Comparative investigations were undertaken between the two structures in order to produce a representation of the process that is as reliable and robust as possible. The number of input and output nodes was determined in a similar way to conventional black-box, discrete input-output modelling. The process is considered to be characterised by the NARX (Non-linear, Auto-Regressive, eXogenous) model defined by:

$$y(t) = F\left[\, y(t-1),..,y(t-n_a), u(t-k),..,u(t-k-n_b) \,\right] + e(t) \qquad (3)$$

where F is some unknown non-linear function, y and u are the process outputs and inputs respectively, e is a Gaussian distributed white noise sequence, k is the process deadtime, n_a and n_b are the number of past output and input data used in the model structure. The well-defined dynamics of the liquid-level process simplified the selection of the NARX model structure to be a 2nd order model with one sample delay between input and output (i.e. n_a=2, n_b=1 and k=1). Hence, the four network inputs consisted of two past outputs and two past inputs from the process and the network output was the process output at time,t. The application of the spread encoding results in each data value being spread

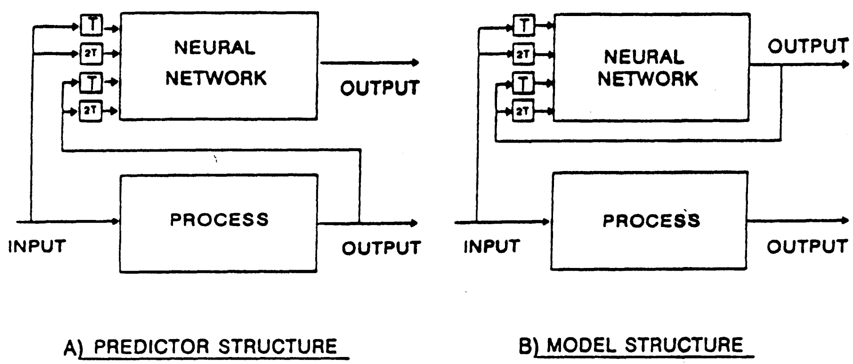


Figure 2 Predictor and model structures for neural network operation

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

over 6 nodes. As a consequence, the network input/output structure is 24 input nodes (past process data) and 6 output nodes (present process output).

The liquid-level process has been simulated using both classical linear and non-linear equations and was utilised for initial investigations into the ability of a network to adequately represent the process. It was necessary, in the development of the model, to investigate not only the number of network inputs and outputs but also the number of hidden nodes and the necessary precision between desired and actual network outputs. At this time, analytical techniques have not been developed to enable network designers to specify the optimum number of nodes in the hidden layer, or even whether there should be additional layers. Consequently, it was necessary to conduct experiments to determine the number of hidden nodes, the required mean square error (MSE) value and the number of training passes (Lisboa *et al.* [12]) and details are presented later.

CONTROL ALGORITHM

It was decided to use a Predictive Control strategy in this investigation. The objective of the algorithm is to find the value of control input, u(t), at time step t=1 that minimises the cost function:

$$J=\sum_{t=1}^{N_1}[y_r(t+1)-Y_{nn}(t+1)]^2+\sum_{t=1}^{N_u}\lambda[u(t)-u(t-1)]^2 \tag{4}$$

where λ is a weighting factor, y_r is the required process output, Y_{nn} is the network output, u is the control signal, N_1 is the prediction horizon and N_u is the control horizon.

The neural network model is used to predict future output responses of the process, $Y_{nn}(t+1)$, and these values are compared with the future desired values of the outputs within the cost function. The objective of this function is to minimise the error between the required set point and predicted process output, subject to a weighting on the control signal. The algorithm determines the next controller output to be applied to the process.

The major advantage of Predictive Control is that it incorporates predictions for a number of future time steps to the horizon. This strategy enables the model-based control system to anticipate where the process is heading. Values of manipulated variables are computed to ensure that the predicted response has certain desired characteristics. One sampling period after the application of the current control action, the prediction response is compared to the actual response. Using corrective feedback action for any errors between actual and predicted responses, the entire sequence is repeated at each sample instant.

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

The control objective is to have the corrected prediction approach the set point as closely as possible. Only the first prediction is implemented. The advantage of this procedure, is that it gives early detection of modelling errors of disturbances and approximately corrects for them. One problem with the control law, is that it can result in excessively large changes in the manipulated variable. An approach to counteract this is to penalise the change in movements of the manipulated variable, u, in the cost function, with the weighting factor λ in equation (4).

NETWORK TOPOLOGY AND TRAINING

The selection of the type of training data for the network was investigated. In order to capture the dynamics of a non-linear system, both the magnitude and frequency response of the process must effectively be captured. Standard frequency response testing is inadequate unless undertaken for a range of sinusoidal amplitudes. Investigations also highlighted inadequacies in PRBS excitation due to there only being two excitation levels which did not excite the neurons over their operational span (Lisboa *et al.* [12]). Consequently, it was decided to use a Random Amplitude Signal (RAS) to provide a rich excitation signal for process operation over a wide non-linear operational region and produce suitable data for network training and validation.

It is important to ensure that the MLP network is trained over a wide operating range not only to capture the non-linear dynamics but also to ensure it is wider than the proposed operating range of the process. To achieve successful representation of a process, it is important to provide adequate excitation over the operational region. Process output data obtained with a RAS input will produce fewer values at the outer regions of the process excursions. Consequently, it is important to test the network in a slightly smaller region than in which it was trained.

The training data for the neural network model was collected from both the conventional mathematical simulation of the process and also real data from the laboratory process. In each case, the process was excited by a random amplitude signal applied to the process input flow rate via the valve. A set of 1000 input-output data points were obtained from the simulation and 300 from the laboratory process, due to process operational constraints.

In order to determine the optimum number of hidden nodes, a series of MLP networks with different complements of hidden nodes were trained to find the acceptable minimum MSE without the computational time becoming excessive. These tests resulted in 6 hidden nodes being selected as suitable for the network. Hence, these investigations resulted in a network with 24 input nodes, 6 nodes in the hidden layer and 6 output nodes.

The network topology was then fixed and tests continued to compare the relative merits of the 'predictor' and 'model' approaches for network operation. Both network configurations are illustrated in Fig.2. The ability of the trained network to represent the process was tested by comparing the steady-state and transient responses of the network outputs with that of the process. These tests consisted of a number of different excitation signals, which included a PRBS, RAS, sine and a set of step changes, to fully prove the simulation capability of the network over a wide operating region.

The evaluation of these validation tests resulted in the reappraisal of network training. It was important to ensure that the training activity produced an accurate representation of the process. The initial values of the gain and momentum terms in the BEP algorithm were 0.9 and 0.6 respectively. The investigations showed that the minimum MSE could be further reduced by decreasing the gain and momentum values to 0.4 and 0.15 after 70 passes of the complete data set. It was found that initially, a high gain is advantageous to speed initial network convergence. However, at later stages, a lower value is beneficial in reaching the global minimum of the search space. The network was trained to a MSE of 0.8623 with 200 training passes, the gain and momentum were initially 0.9 and 0.6 respectively and were reduced as the training proceeded to final vales of 0.01 and 0.001 respectively. The ability of the network to represent the process was evaluated by testing its performance with a series of test signals, to verify both the steady-state and dynamic performance, and is described in the next section.

MODEL VALIDATION

The trained neural networks representing the mathematical model of the process and the real process were evaluated on test data sets that were not used during training. These included another random amplitude signal a pseudo-random binary sequence signal (PRBS), and finally a set of steps so as to evaluate the networks steady-state performance. The validation of the mathematical model is illustrated in Figs.3 and 4, and the real process in Figs.5 and 6.

Fig.3 illustrates the network tested on the random signal as a one-step-ahead predictor, and also as a model. The network's performance is adequate in both cases. The steady-state performance of the neural network as both a model and predictor is shown in Fig.4 for a set of step inputs. For this test the prediction accuracy as a model is noticeably poorer than as a predictor when required to predict steady states close to the outer region on which the network was trained (>70 cm). Nevertheless, the results demonstrate stable operation as a model and again, acceptable predictions of the process output with a maximum error of 4cm. The neural network that was trained to represent the real process was also evaluated on the same signals described above. Figs.5 and

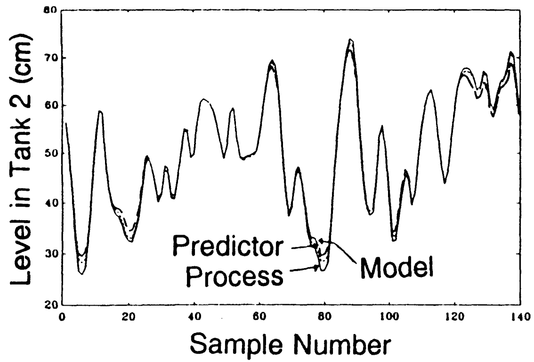


Figure 3 Validation of neural network model on a random signal (simulation)

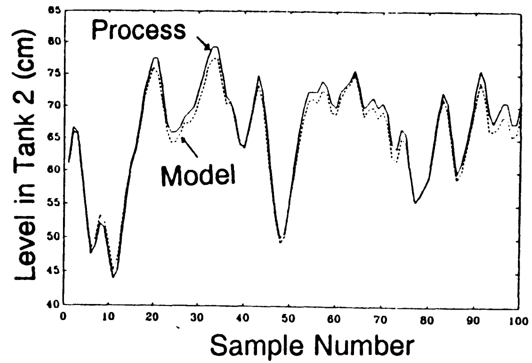


Figure 5 Validation of neural network model on a random signal (real data)

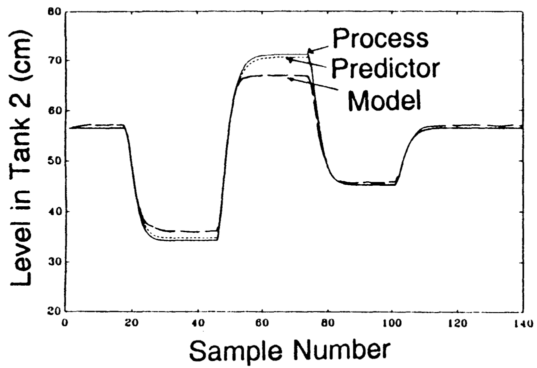


Figure 4 Validation of neural network model on step responses (simulation)

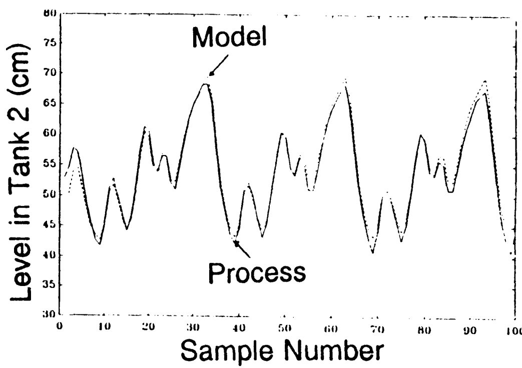


Figure 6 Validation of neural network model on a PRBS (real data)

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

6 show the results of the network trained with real process data. It is seen that in each of the cases the neural network performs adequately to real data.

Investigations into the relative merits of operating the 'predictor' and 'model' network architectures have indicated that the model form is the more useful because it can be operated independently from the process, unlike the predictor structure. An accurate model also improves the performance of the predictive controller by providing accurate future predictions of the process response. Investigations have shown that the spread encoding technique does enable an accurate model representation to be obtained with the MLP and this is illustrated in these results.

As the initial simulation training data was noise-free, investigations were also undertaken with a Gaussian distributed noise signal added to the process output, with a signal to noise ratio of 20dBs, to represent measurement noise. In each case the ability of the network to represent the process or its simulation was evaluated. The results indicated that for networks trained with or without noise added to the process output both gave acceptable results when recalled in the predictor and model structures.

CONTROL RESULTS

Initially, the development and testing of the predictive controller with a neural network model (NN-PC) algorithm was achieved using the simulator, since the on-line development, implementation and evaluation of the control scheme is normally time consuming and in industry would be subject to the availability of the plant. Consequently, the liquid-level process was modelled by conventional non-linear differential equations, the network by standard high-level (FORTRAN) language statements to represent each node in the network and the optimiser by a routine which evaluated the cost function every sample time. The combined elements were incorporated in the continuous simulation package, ACSL.

The control scheme of a one-step-ahead predictor, $N_1=1$, $N_u=1$ in equation (4), is illustrated in Fig.7 and was used throughout these results. The performance of the controller was initially tested and consequently evaluated on the process simulation. A number of validatory tests were defined to ensure that the controller and neural network model were able to achieve the required steady-state and dynamic characteristics. Fig.8 illustrates the results of applying a series of set-point changes to evaluate the steady-state and transient responses. The expected effect of λ can be seen, namely that when λ is zero, the response is oscillatory but fast acting. Additionally, as λ increases, the output response is more sluggish as changes in the control signal are penalised but a much smoother control input is achieved. In Fig.9, a comparison can be made between a NN-PC controller and a well-tuned PID controller for small changes

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

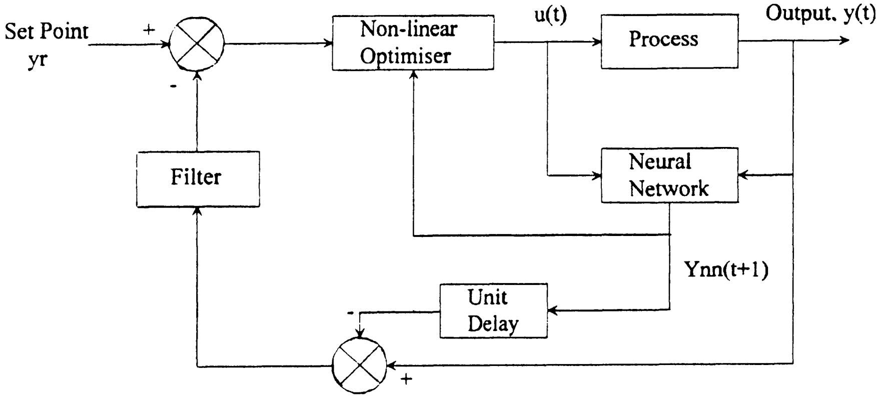
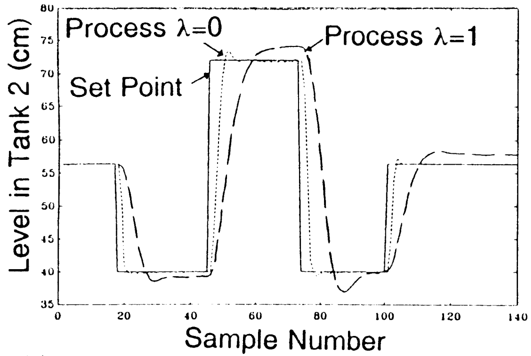


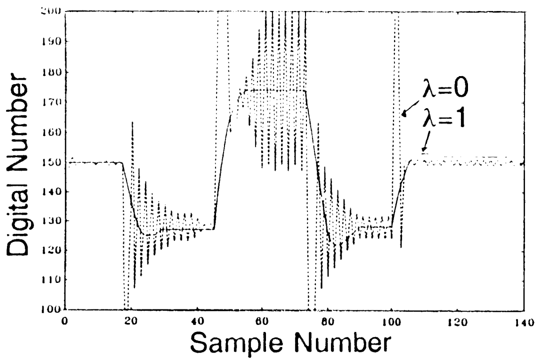
Figure 7 Predictive control scheme incorporating a neural network process model

in set-point. The results show that the PID controller marginally outperforms the NN-PC (with $\lambda = 0$) and produces a less oscillatory response. This result could be predicted since the process will be quite linear over this small region of operation. Fig.10 compares the NN-PC and PID for larger step sizes which will drive the process into non-linear regions. It can be seen that in this case the control achieved with the NN-PC is an improvement over that of the PID controller. The PID controller is now required to operate under different conditions to when it was tuned, because of the process non-linearity, and would require retuning at the new operating points.

The successful development of the controller and NN model in simulation provided a good foundation for on-line tests. These evaluation tests were similar to those used in the simulation, i.e. a series of step demands in the set point. The on-line results are illustrated in Fig.11. The expected oscillatory response was replaced by an almost overdamped transient at the first set point change (from 59cm to 40cm). Further investigation of the control signal (Fig.11b) showed that although the NN-PC was generating large corrective signals, the physical limitation of the valve operating range was preventing the expected response. A more desirable control signal could be achieved by using a value of λ larger than zero at the expense of slower set point tracking, as illustrated in Fig.8b. A PID controller tuned at the initial operating point was subjected to the same series of set-point changes (Fig.12). It can be seen that the controller had the same saturation problems and furthermore the results indicate a poorer overall performance when compared with the NN-PC. It was noted that the on-line tuning of the PID controllers was tedious and time consuming, as expected, and the controllers naturally could not remain optimal over the non-linear region. The NN-PC results in simulation and practice demonstrate the improved performance that can be achieved with neural networks for control of non-linear systems.



(a) Process response



(b) Control signal

Figure 8 NN-PC with λ=0 and λ=1 (simulation)

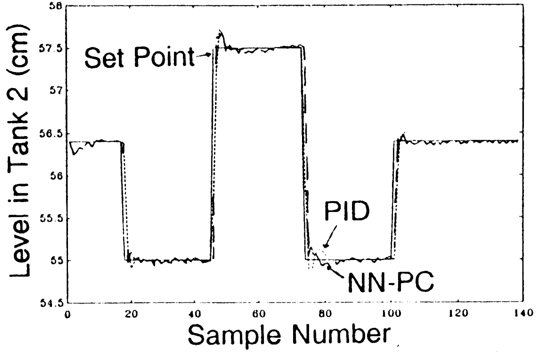


Figure 9 PID control versus NN-PC for small step demands (simulation)

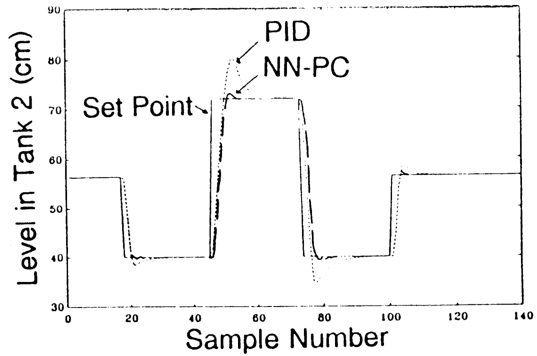
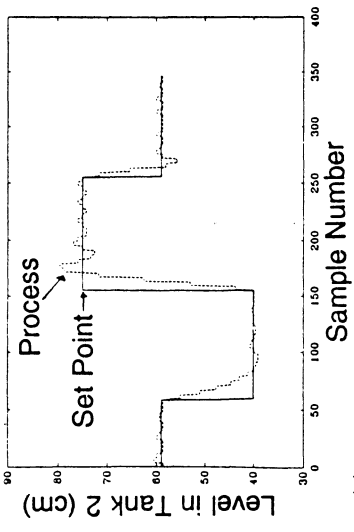
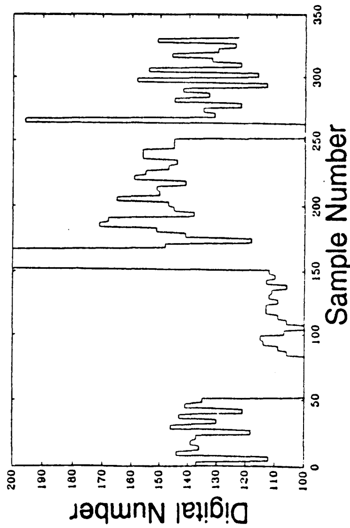


Figure 10 PID control versus NN-PC for large step demands (simulation)

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

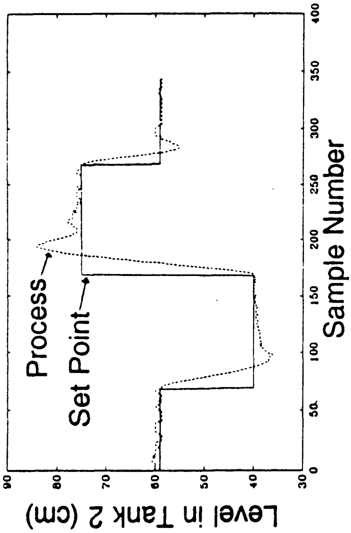


(a) Process response

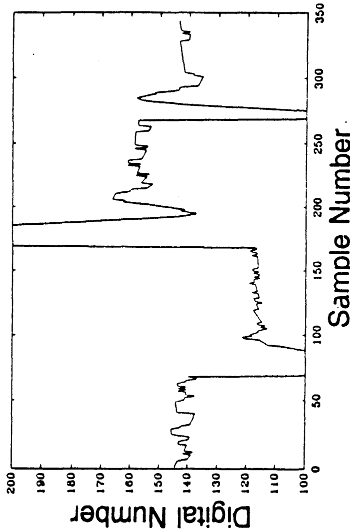


(b) Control signal

Figure 11 On-line NN-PC for $\lambda=0$



(a) Process response



(b) Control signal

Figure 12 On-line PID control

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

CONCLUSIONS

The paper has presented the development of a controller employing a neural network model to control a real process. A simulation of the process was utilised to develop the controller prior to on-line implementation. The results show that both the model and the real non-linear process can be accurately modelled by a MLP neural network. The paper also shows that it is necessary to include basic design methodology to optimise the topology of the network and enable an efficient training procedure. The investigations have concentrated on a thorough validation of the process model, by neural network, prior to developing the controller. These investigations included consideration of different network structures, the number of training passes required to achieve the necessary accuracy, testing by the inclusion of simulated measurement noise and a range of excitation signals. The results show the network is capable of modelling a real process.

The successful development of the model enabled the investigation into the operation of a standard optimal controller. Again, the philosophy was to investigate and develop the controller by simulation prior to on-line implantation. The results show successful controller operation both in simulation and on-line and an improved performance over a tuned PID controller. Further work is in progress to assess the on-line control performance when long range prediction is implemented and to make an on-line comparison of the performance of different control strategies incorporating neural networks.

ACKNOWLEDGEMENTS

The authors would like to thank the Science and Engineering Research Council (UK) for financial support of J.T. Evans.

REFERENCES

1. Miller III, W.T., Sutton, R.S., and Werbos, P.J., *Neural Networks for Control* MIT Press, Cambridge MA, 1990.

2. Hunt, K.J. and Sbarbaro, D. 'Neural Networks for Nonlinear Internal Model Control' *IEE Proc. D.*, Vol.138, No.5, pp. 431-438, 1991.

3. Narendra, K.S. and Parthasarathy, K. 'Identification and Control of Dynamical Systems using Neural Networks' *IEEE Transactions on Neural Networks*, Vol.1, No.1, pp.4-27, 1989.

4. Willis, M.J., Di Massimo, C., Montague, G.A., Tham, M.T. and Morris, A.J. 'Artificial Neural Networks in Process Engineering' *IEE Proc. D.*, Vol.138, No.3, pp. 256-266, 1991

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

5. Montague, G.A., Tham, M.T., Willis, M.J. and Morris, A.J. 'Predictive Control of Distillation Columns using Dynamic Neural Networks', pp. 231-236, *Preprints 3rd IFAC Symposium DYCORD+'92*, Maryland, USA, April, 1992.

6. Werbos, P. *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*, Ph.D. Thesis, Harvard University, 1974.

7. Rumelhart, D.E. and McClelland, J.L. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol.1, Chap.8, MIT Press, Cambridge MA, 1987.

8. Hornik, K., Stinchcombe, M. and White, H. 'Multilayer Feedforward Networks are Universal Approximators', *Neural Networks*, Vol.2, No.5, pp. 359-366, 1989.

9. Evans, J.T., Gomm, J.B., Williams, D., Lisboa, P.J.G. and To, Q.S. 'Non-linear Process Modelling with Artificial Neural Networks', pp. 637-641, *Proc. 24th SCS Summer Computer Simulation Conference*, Reno, USA, 27-30th July, 1992.

10. Evans, J.T., Gomm, J.B., Williams, D., Lisboa, P.J.G. and To, Q.S. 'Modelling a Non-linear Process using Artificial Neural Networks', *Proc. Sixth IMA Conference on Control: Modelling, Computation, Information*, Manchester, UK, 2-4th September, 1992.

11. Lisboa, P.J.G. 'Neural Networks for Control: Evolution, Revolution or Renaissance.' Chapter 2, *Application of Neural Networks to Modelling and Control* eds. Page, G.F., Gomm, J.B. and Williams, D., Chapman & Hall, London, in press 1992.

12. Lisboa, P.J.G., Gomm, J.B., Evans, J.T., Williams, D. and To, Q.S. 'Investigation into the Application of Neural Network Models for System Identification', pp. 8/1-8/5, *Proc. IEEE Colloquium on Neural Networks in Control and Modelling of Industrial Processes*, PCL, London, April, 1991.