NEURAL NETWORKS FOR THE DEVELOPMENT OF INTELLIGENT CONTROLLERS

Marzuki Khalid
Department of Control Engineering
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
Jalan Semarak
54100 Kuala Lumpur

Correspondence address:

Dept. of Information Science & Intelligent Systems
Faculty of Engineering
University of Tokushima
Tokushima-shi, 770
Japan
(Received 19 March 1992/Accepted 8 July 1992)

RINGKASAN : *Kertaskerja ini menekankan pembangunan pengawal-pengawal cerdik dengan menggunakan rangkaian neural "backpropagation" untuk dua jenis sistem kawalan suhu iaitu sebuah pembakar industri mini dan sebuah loji air panas. Kaedahnya adalah mudah dan sesuai untuk digunakan dengan senangnya kepada berbagai jenis sistem kawalan stabil. Dari maklumat masukan-keluaran proses-proses tersebut, rangkaian neural ini mempelajari model dinamik songsang untuk digunakan sebagai pengawal kepada loji masing-masing. Untuk membuktikan kebolehan pengawal-pengawal neural ini, beberapa ujikaji dilakukan dan dibandingkan dengan pengawal-pengawal konvensional. Keputusan ujikaji menunjukkan pengawal-pengawal neural tersebut adalah lebih baik kawalannya, mudah untuk direkabentuk, dan tidak perlu dilaraskan (tuning).*

ABSTRACT : This paper addresses the development of intelligent self-learning controllers using back propagation neural networks for two kinds of temperature control processes: a miniaturized model of an industrial furnace and a water bath. The method does not require the use of any prior conventional controller or knowledge regarding dynamics which can be easily applied to a wide variety of stable control systems. The neural networks learn the inverse dynamic models from the open-loop input-output behaviour of the process plants. Once trained, they are configured as direct controllers to the plants similar to a conventional feedback control scheme. A novel feature of the neuro-controllers is that they can be continuously trained on-line by specialized learning, to further improve their performance. Experimental results show that the neuro-controllers compare well with conventional feedback controllers, giving improved performance and robustness, and are easier to implement with no requirement for tuning when confronted with moderate load disturbances.

KEY WORDS : Backpropagation neural networks, intelligent controllers, input-output behaviour, self-learning, inverse dynamic models, real-time control.

INTRODUCTION

Learning through experience is one natural course of human life. Human intelligence allows us to generalize and categorize. For example, we can easily identify a bird by generalizing when we see one, even though we may have never seen its kind before. Intelligent machines are currently being developed to learn in a similar way with the advent of artificial neural networks from the behavioural characteristics of processes. Modelled after the human brain, artificial neural networks consist of many interconnected simple non-linear elements called neurons, as shown in Figure 1. The strengths of the interconnections are denoted by parameters known as weights and are gradually adjusted to improve performance, depending on the task given. The neural network learns from the adjustment of these weights, step by step, typically to minimize some objective function. This ability to learn is one of the main advantages of neural networks and offers the possibility of solving many engineering problems in a much simpler way.

Of the various neural network paradigms, backpropagation is the most widely used. Its simple learning and update procedure has made it the mainstay of neuro-computing which has served a wide variety of applications. Neural networks have excelled in the field of pattern recognition and they are currently being studied to solve problems related to control. In pattern recognition they are generally trained to memorize and recall the patterns, whereas, in control they are trained to learn the relationship of the patterns. Recent results have shown that, at least, in principle, neural networks can represent most classes of continuous functions with bounded inputs

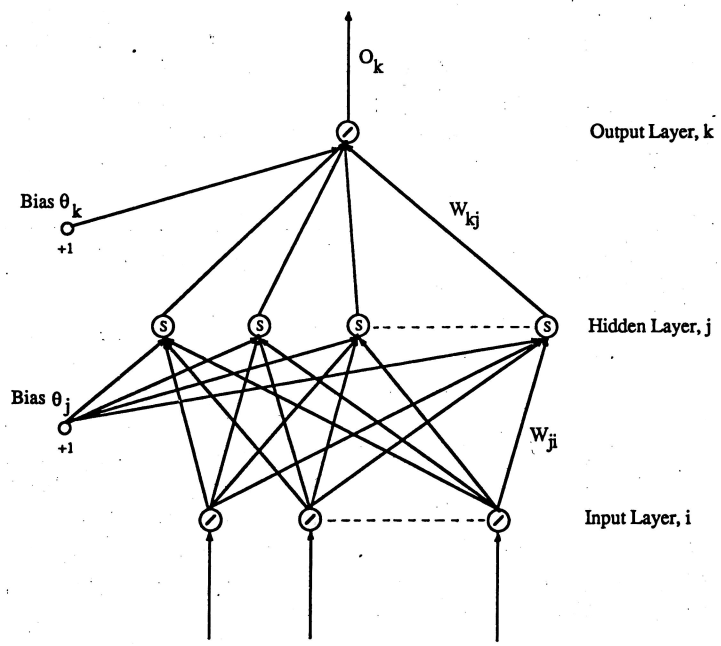

Figure 1. *A multilayered backpropagation neural network with one layer of hidden neurons. Sigmoid functions are used in the hidden neurons and linear functions are used in the input and output layer neurons*

and outputs to arbitrary precision (Cybenko, 1988; Hecht-Nielsen, 1989; Hornik *et al.*, 1989). In addition, the non-linear mapping ability of the neural networks has also made it possible to solve highly non-linear control problems where traditional control approaches have failed. Since there have been numerous simulation studies (Psaltis *et al.*, 1988; Miller *et al.*, 1990; Jordan and Rumelhart, 1991) showing the viability of neuro-control, it would seem appropriate to begin to apply these methods to realistic control problems to determine how well they work in practice and, if not, where they need further refinement.

This paper addresses the development of self-learning controllers using backpropagation neural networks for two different temperature control systems: a miniaturized model of an industrial furnace and a water bath process, without the use of any prior conventional controller or knowledge regarding their dynamics. The method carried out to develop the neural network-based control systems can be applied to a wide variety of open-loop stable control systems using only the open-loop input-output behaviour of the plants. The neural networks are first trained off-line to learn the inverse dynamic models of the plants using the general learning architecture as proposed by Psaltis *et al.*, (1988). By back-propagation of the performance error, which is the error between the plant output and the desired output, the neural network controllers can be continuously trained in an on-line way (specialized learning) to further improve their performance. Although the location of the plant imposed some difficulty of backpropagating the performance error, Saerens and Soquet (1989) have shown that the plant's partial derivatives may be approximated by their sign. This can be done when the orientation in which the control parameters influencing the outputs of the plant is known, without any decrease in performance. The performances of the neuro-controllers are compared with conventional feedback controllers, namely the proportional-plus-integral-plus-derivative controllers (PID) implemented on the same processes, respectively. This paper has been organized as follows. In the next section, the back-propagation neural network algorithm with some modification for the control purpose is briefly discussed. A brief description of the two temperature control systems are given in the following section. The development of the neuro-controllers is also discussed. The results of the experimental studies are compared and discussed in the section that follows.

BACKPROPAGATION ALGORITHM FOR CONTROL

The backpropagation algorithm has been extensively derived and discussed in many literatures which include Rumelhart *et al.*, (1986), and Pao (1989). In this section, the backpropagation algorithm is briefly described with some modifications which can be used to develop self-learning controllers for a class of single-input single output control systems using the scheme as shown in Figure 2. A multilayered backpropagation neural network consists of at least one hidden layer (Figure 1). The input layer consists of linear unit function neurons i, in which their outputs (o_i) are connected to the inputs (s_j) of the hidden layer neurons j, through the gradually updated weights (w_{ji}) as follows:

$$s_j = \sum_i o_i w_{ji} \tag{1}$$

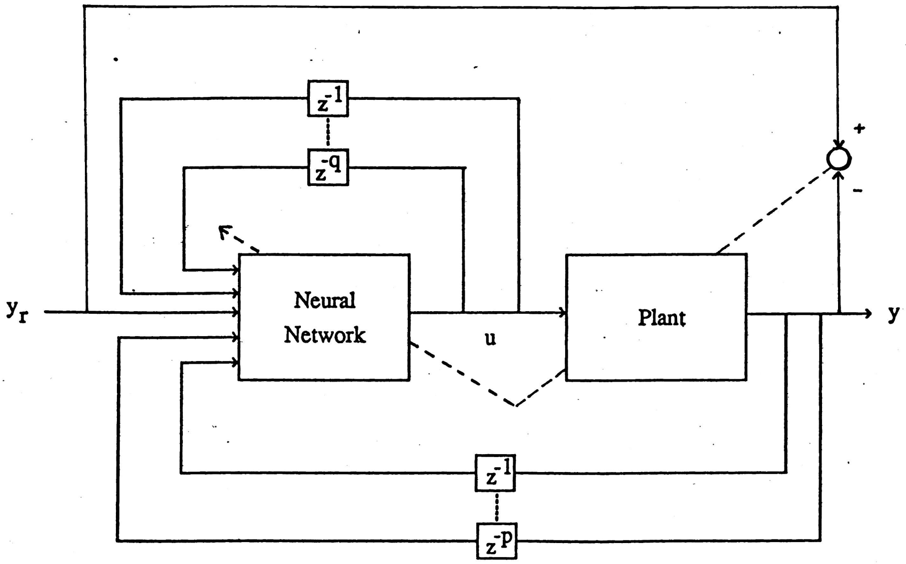

Figure 2. *A feedback neural network control scheme. Delayed plant outputs and inputs are used apart from the desired plant output as the input vector elements of the neural network. Specialized on-line learning is carried out by backpropagation of the performance error as shown by the dotted line*

For the general learning scheme, the elements of the input vector of the neural network should consist of the present plant output and some past plant outputs and inputs. The target patterns should consist of the corresponding plant inputs. Determination of the number of past plant outputs and inputs to be used may be done by trial and error, in order for the neural network to learn the true inverse dynamics of the plants successfully. Once the network has been trained the input vector element of the present plant output must be replaced by the desired plant output or the reference value, whereas, the rest of the elements remain the same.

The latter configuration should also be adopted for the on-line specialized learning scheme. In a simple study, Khalid and Omatu (1992) have shown that the use of different input vector elements did have an effect on the neural network in learning the true inverse dynamics of a particular plant.

Each output (o_j) of the hidden layer neurons is activated by a semilinear activation function known as the sigmoidal logistic function as in the following equations:

$$o_j = f\,(s_j + \theta_j) \tag{2}$$

$$f\,(s_j + \theta_j) = \frac{1}{1 + \exp\{-(s_j + \theta_j)\}} \tag{3}$$

where θ_j is a bias term which may be added to improve the convergence properties of the network. The outputs of the hidden layer neurons are connected to the neurons in the output layer k, through another set of weights (w_{kj}) as follows:

$$s_k = \sum_j o_j w_{kj} \tag{4}$$

where s_k is the input to the output layer neuron. The output of the output layer neuron is used as the actual control input to the plant where the sigmoid function is replaced by a saturating linear function. The output if free to adapt to the real values of the target control input, as follows:

$$o_k = s_k + \theta_k \tag{5}$$

where θ_k is the bias term for the output layer neuron. If a sigmoid function is used at the output layer neuron then the output of the neural network has to be normalized. This is because a sigmoid function has saturating binary values of 0 to 1. If the control signal is needed to have bipolar values, i.e., +1 (when on) and -1 (when off), then hyperbolic-tangent signal functions should be used instead of the sigmoid functions in the hidden layer neurons (Kosko, 1992). Backpropagation neural networks involve two steps. The first step involves Eqs. (1)-(5) and are termed as the forward propagation step. The second step is called the backward propagation step in which the neural network learns.

The forward propagation step produces outputs which are compared to the desired target values (t_k) and this is normally measured in the form of mean-squared error (E) as follows:

$$E = {}^1/_2 \sum_k (t_k - o_k)^2 \tag{6}$$

Some researches (for example Nguyen and Widrow, 1990; Yabuta and Yamada, 1991) have shown that it is possible to realize the inverse models by minimizing other types of cost function. For the general learning scheme, the target values are the actual plant inputs, whereas, for the specialized learning scheme the target value is the desired plant output or the reference value. A modified generalized delta rule as proposed by Nagata *et al.*, (1990), has been used in training the neural networks which includes another momentum term, which is found to accelerate the convergence. Thus, the weights between the input and hidden layers are updated as follows:

$$\Delta w_{ji}(n+1) = \eta\delta_j o_i + \alpha\Delta w_{ji}(n) + \beta\Delta w_{ji}(n-1) \tag{7}$$

and between the hidden and output layers as follows:

$$\Delta w_{kj}(n+1) = \eta\delta_k o_j + \alpha\Delta w_{kj}(n) + \beta\Delta w_{kj}(n-1) \tag{8}$$

where η, α and β are the learning rate, momentum, and acceleration coefficients, respectively. The constants η, α and β must be chosen empirically. If initial randomized weights are used, it would be good to start with a large η, around 10^{-2}, depending on the number of patterns chosen. Once oscillation occurs, η should be reduced gradually. This approach ensures progressively fast convergence. A method to use adaptive learning rate for faster convergence can be found in Werbos (1991).

For the weights between the output and the hidden layers, the back error propagation signal is expressed as:

$$\delta_k = (t_k - o_k) \tag{9}$$

and between the hidden and input layers, it is expressed as:

$$\delta_j = f'(s_j) \sum_k \delta_k w_{kj} \tag{10}$$

where $f'(s_j)$ is the derivative of the function of $f(s_j)$. Note that Eqs. (6) - (10) describe the backpropagation step where the equivalent error is computed for the output layer using

Eq. (9) and then propagated backward through the layers towards the input layer using Eq. (10).

DESCRIPTION OF THE TEMPERATURE CONTROL SYSTEMS

The Miniaturized Industrial Furnace

The furnace consists of a triple-channel copper cylinder as illustrated in Figure 3 where each channel can be controlled independently. However, in this paper experiments are conducted on a single channel only. The temperature control system is developed by Omron Inc. which can be divided into the main components as shown in Figure 4. The copper cylinder is 65 mm in diameter and 175 mm in length with a heater capacity of 1200 watts. The output temperature of the cylinder is measured using a thermocouple of the K-type (TC 12-4K) and digitized via a 12-bit analog-to-digital converter (AD6940) with a resolution of 0.2°C.

A solid-state relay (Omron G3N-220B) is used to switch the heater on and off according to the control signal given. When the control signal is 100%, the heater is switched on continuously for 6 seconds in one sampling interval. A programable input-output interface card (PIO-16/16) is used as the interface between an NEC PC-9801F2 microcomputer and the sensor and relay modules. The main control program is written using Microsoft-C, while the subroutine to operate the heater is written in Assembly (Intel's 8086).

The Water Bath Process

A Yamato Science Inc. laboratory water bath (BT-15 model) is the main component of this temperature control system as illustrated in Figure 5. A schematic diagram of the experimental set-up is shown in Figure 6. The system can be divided into five main components: (i) the water bath (ii) the sensor module (iii) the programable input-output (PIO) interface board (iv) the microcomputer and (v) the actuator. A brief description of the five components follows. The capacity of the water bath is 8 litres with dimension 250x290x100 (mm^3). The water bath is heated by a 600 watts base heater which is connected to a thyristor (SI6G12S-12) circuit. To ensure

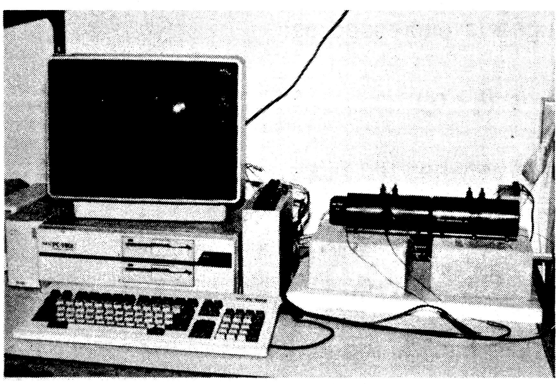

Figure 3. *The temperature control system of the miniaturized industrial furnace*

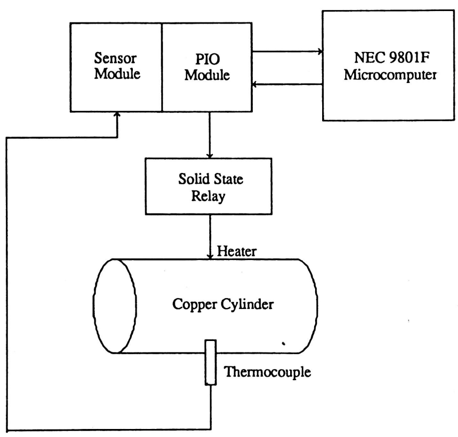

Figure 4. *A schematic diagram of the furnace control system*

even temperature distribution, a stirrer is used which can rotate at 120 rpm. The sensor module has been developed using diodes (1S1588) and high gain amplifiers (A741). It consists of a two step amplification circuit and can transform the measured temperature over the range of 0°C to 100°C into the corresponding voltage range of 0 volt to 10 volts with a resolution of 0.24°C.

The PIO interface circuit board consists of an analog-to-digital (A/D) converter, a digital-to-analog (D/A) converter and a programable peripheral interface device (PD8255A). An external clock is designed to operate the A/D and D/A converters. The clock circuit is designed using crystal oscillator and JK flip flops. The micro-computer

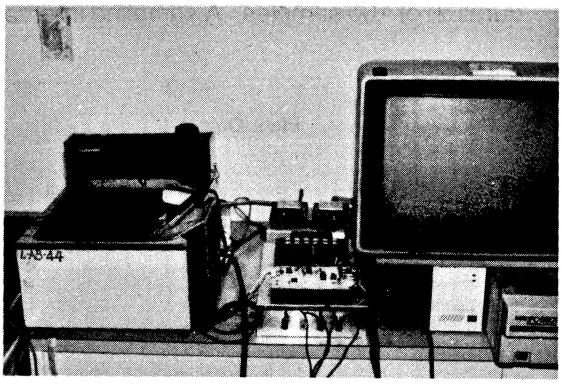

***Figure 5.** The water bath temperature control system*

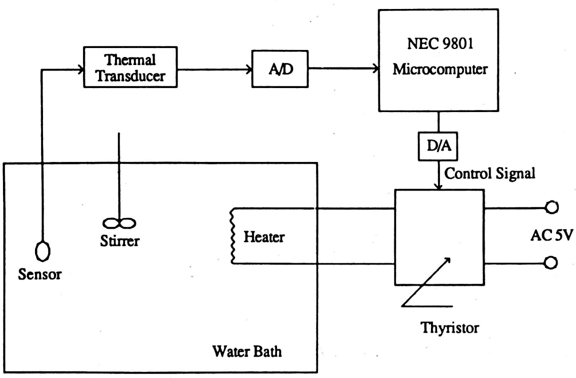

***Figure 6.** A schematic diagram of the water bath temperature control system*

used in this experiment is the NEC PC 9801F having an Intel 8086 16–bit CPU with a 10 MHz clock speed. A simple control routine is written using Microsoft–C to provide the control input to the actuator through the D/A and also to measure the output temperature.

A thyristor (S16G12S-12) is used as an actuator for the heater and is switched on and off according to the following constraints where :

$u(kT)$ = Output of the neural controller
T = Sampling interval
k = 0,1,2,3,...
V_i = Input voltage to the actuator

If $u(kT) \leq 0.0$, then $V_i = 0.0$ volt, heater is OFF. If $0.0 < u(kT) < 5.0$, then $V_i = u(kT)$ volts, heater is ON at an intermediate output. If $u(kT) \geq 5.0$, then $V_i = 5.0$ volts, and the heater is full ON.

DEVELOPMENT OF THE NEURO-CONTROLLERS

Multilayered neural networks with one hidden layer are used to develop the inverse dynamic models which are then configured as direct controllers to the processes. Although the estimated reduced-order mathematical models of these process plants can be derived, in this approach no prior mathematical analyses of the processes are being made. The idea of this approach is to use the self-learning ability of the neural networks based on input-output characteristics of the plants, without the need to use any initial conventional controller or prior knowledge regarding dynamics. The neural networks are trained off-line to learn the inverse dynamic models from the open-loop input-output behaviour of the processes.

In using the general learning architecture, a set of training patterns has to be obtained in prior. In a number of applications (Psaltis *et al.,* 1988; Narendra and Parthasarathy, 1990; Jordan and Rumelhart, 1991), the training patterns are selected by probing the plant input using random signals. As this technique is rather impractical for these slowly varying temperature control processes and, in addition, are open-loop stable, a ramp signal is instead directly injected in the plants within the limits of the actuator input constraints. By measuring the corresponding output values, a set of training patterns which describe the behaviour of the processes can be obtained. It must be noted that in order for the neural network to control the plant well in the region of interest, the training patterns must at least consist the values of the plant outputs within that region of interest.

Figure 7 shows the open-loop input-output characteristics of the furnace by driving a ramp input signal between 0% and 100% with an increment of 1% per sample over a duration of 100 samples where a sampling interval of 15 seconds was used. Similarly, for the water bath, a ramp signal between 0 volt and 5 volts was injected with an increment of 0.55 volt per sample over a duration of 100 samples. A sampling interval

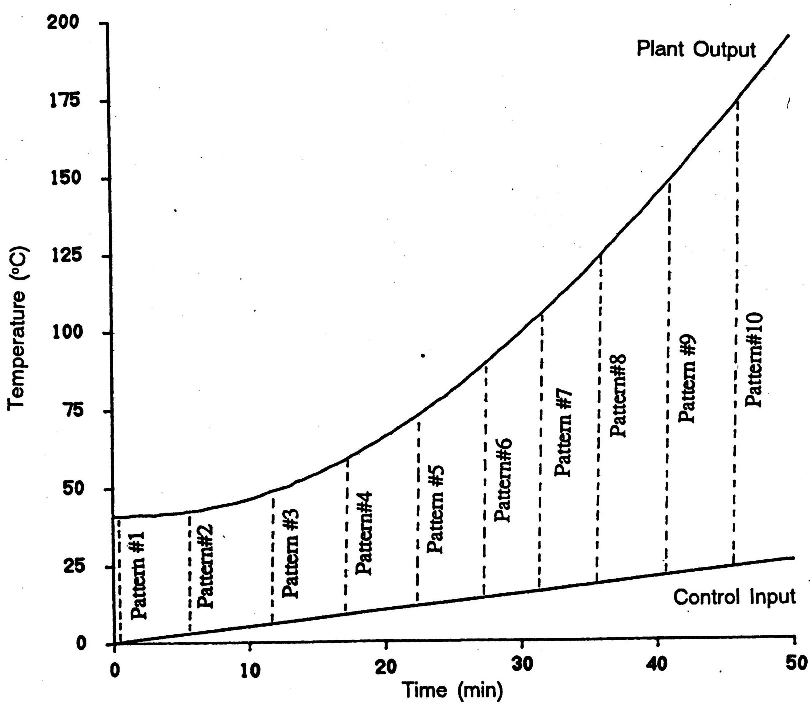

Figure 7. *Open-loop input-output behaviour of the furnace process plant. A ramp signal is injected to the actuator of the plant between 0% and 100% with an increment of 1% per sample. Ten patterns are selected for training the neural network to learn the inverse dynamic model*

of 30 seconds was used and the plant open loop input-output characteristics was obtained from the input-output measurements.

The number of training patterns, the number of hidden units and the number of input vector elements of the neural network models are all chosen by experiments as there is still no reliable method of determining these parameters systematically or automatically. Several neural network models were chosen having different input vector elements and hidden neurons which were trained on a 16 MHz IBM 386 microcomputer with a math co-processor. The convergence time is between two to seven days for each model depending on the number of hidden neurons, the number input vector elements, and the number of training patterns chosen. A large learning rate, η, was used initially at 0.01 and once oscillation occurs, it was reduced to a tenth of its previous value. The momentum terms, η and β, were chosen to be 0.8 and -0.15 respectively. For each model, training was stopped once convergence did not reduce by more than 0.001 per cent over 1000 iterations.

Each of the neural network models is then tested by configuring it directly to control the respective processes. It was found that for the furnace a neural network structure of 10 hidden neurons and two input vector elements consisting of the present plant output and one delayed plant output was suitable to realize the inverse dynamic model. A set of 10 training patterns has been used as shown in Figure 7. For the water bath process, the neural network model has eight hidden neurons with two input vector elements and a set of seven training patterns has been used (Khalid and Omatu, 1992).

RESULTS AND DISCUSSIONS

The off-line trained neural networks are configured as controllers to the respective processes (Figure 2) and on-line specialized learning is carried out by backpropagation of the performance error (dotted line in Figure 2). For each process, an experiment to follow set-point changes is conducted. For the furnace, the experiment was conducted over 200 samples with a sampling interval of 15 seconds which resulted in a 50-minute duration. In the case of the water bath, the experiment was conducted over 100 samples with a sampling interval of 30 seconds which also resulted in a 50-minute duration. The difficulty in controlling these two processes is that only positive input signals can be given.

The performance of the furnace neuro-control system with one cycle of on-line training is shown in Figure 8. While the performance of the water bath neuro-control system is shown

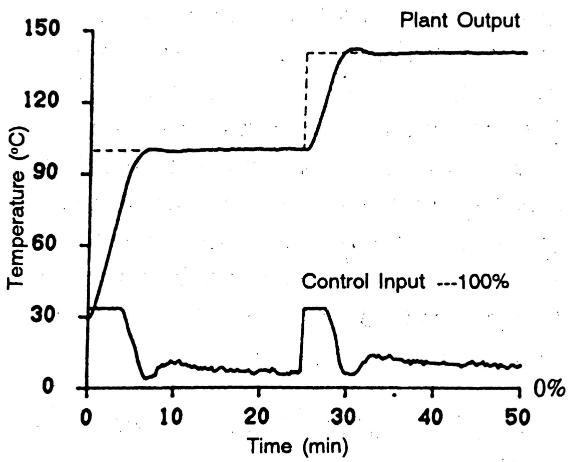

Figure 8. *Performance of the neuro-controller on the furnace temperature control system with one cycle of on-line learning with respect to set-point changes*

in Figure 9. It was observed that for the water bath, the neural network controller was able to perform very well even without on-line training (Khalid and Omatu, 1992).

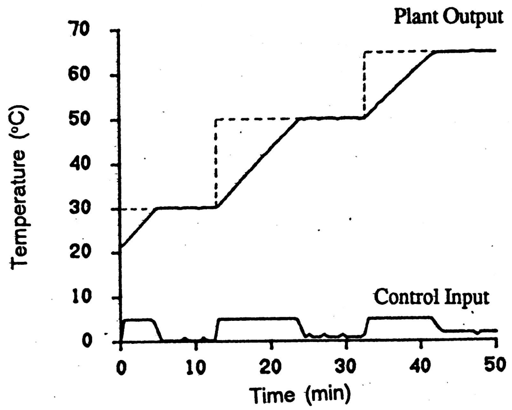

Figure 9. Performance of the neuro-controller on the water bath temperature control system without on-line learning with respect to set-point changes

However, for the furnace, the off-line general training alone is not sufficient where steady-state errors exist. This could be due to the fact that the noise level in the furnace is much higher than the water bath. With specialized on-line learning on the furnace using a large learning rate, it was observed that oscillations occurred at the plant output. When the learning rate was too small, the improvement effect was negligible. Therefore, it is important to choose the learning rate correctly. With the appropriate learning rate chosen at 0.00002, the performance of the furnace improves which is shown with the one cycle of on-line learning.

The performance of the neuro-control systems are compared to a conventional feedback control scheme implemented on each of the two processes using the popular velocity form discrete-time PID control (Ogata, 1987) as follows:

$$\Delta u(k) = K_p[E(k) - E(k-1)] + K_i E(k) + K_d [E(k) - 2E(k-1) + E(k-2)] \qquad (11)$$

where;

K_p = Proportional gain

K_i = Integral gain

K_d = Derivative gain

$E(k)$ = Performance error between $y(k)$ and $y_r(k)$

$\Delta u(k)$ = Increment of the control input at the sampling instant k.

For the furnace, a three-term PID-controller is used and for the water bath a two-term controller is used. They are tuned using the method of Takahashi *et al.,* (1971), and their performance, with respect to the same set-point changes, are shown in Figures 10 and 11. For the furnace, the PID-controller parameters were tuned to 11.37, 1.88 and 17.28 for K_p, K_i and K_d, respectively. For the water bath, the PI controller parameters were tuned to 1.8 and 0.4 for K_p and K_i, respectively. Comparing the two algorithms on the two systems, it can be observed that the neural network-based control systems performed better. The neural network controllers operated

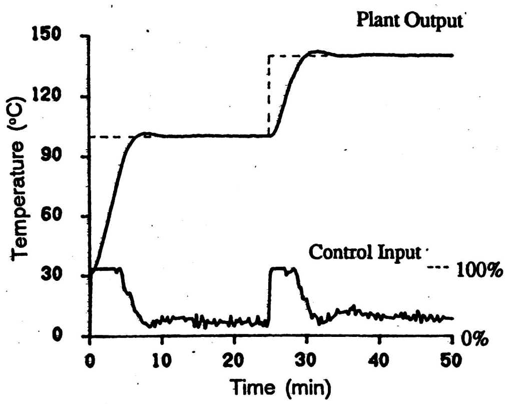

Figure 10. Performance of the PID-controller on the furnace temperature control system with respect to set-point changes

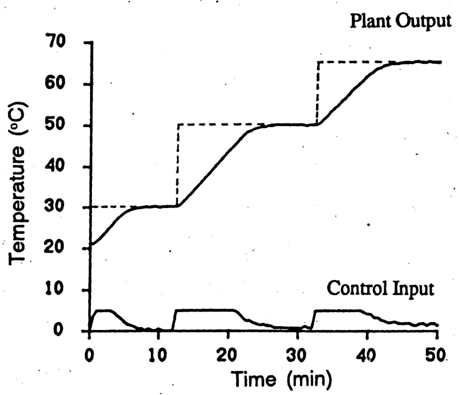

Figure 11. Performance of the PI-controller on the water bath temperature control system with respect to set-point changes

extremely fast in achieving the given set points showing good generalization capabilities. On the other hand, the performances of the conventional controllers are rather sluggish.

In another set of experiments, the ability of the neuro-controllers to overcome unknown load disturbances imposed on the processes are studied. The same set-points were used in the furnace experiments but in the water bath experiments, only one set-point at 50°C was used. In order to compare the algorithms under identical conditions, artificial load disturbances of 7°C and -5°C were added to the furnace output at the 65th and the 165th sampling instants, respectively. For the water bath, artificial load disturbances of 3°C and -3°C were added at the 53rd and the 75th sampling instants, respectively. The neural networks were not re-trained off-line and the feed-back controllers were not retuned. Figures 12 and 13 show the performances of the neuro-controllers due to load disturbances on the furnace and water bath, respectively while Figures 14 and 15 show the performances of conventional controllers. For clarity, the performance of the systems at the sampling periods of interest are shown in the Figures. The neural network controllers performed considerably well for both processes with the ability to recover very fast from the effects of these load disturbances. On the other hand, the feed-back controllers showed a poor rate of recovery which affected the systems badly. The neural networks are able to perform very well as they have the ability to adapt quickly to changes at their inputs. The feedback controllers have to be re-tuned in order to overcome the effects of the load disturbances.

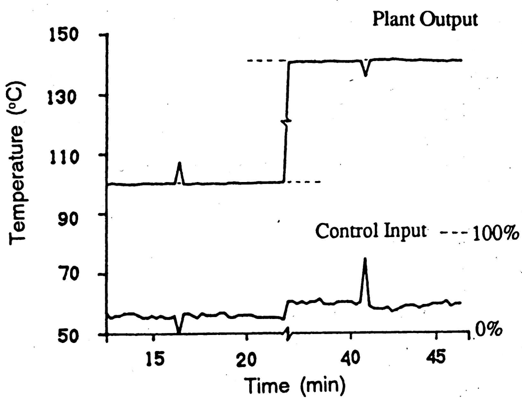

Figure 12. Performance of the neuro-controller on the furnace temperature control system under the effect of load disturbances

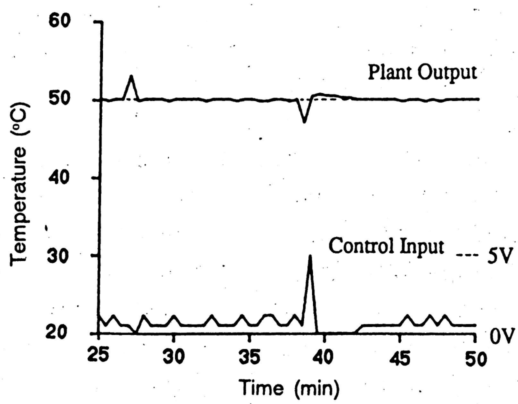

Figure 13. Performance of the neuro-controller on the water bath temperature control system under the effect of load disturbances

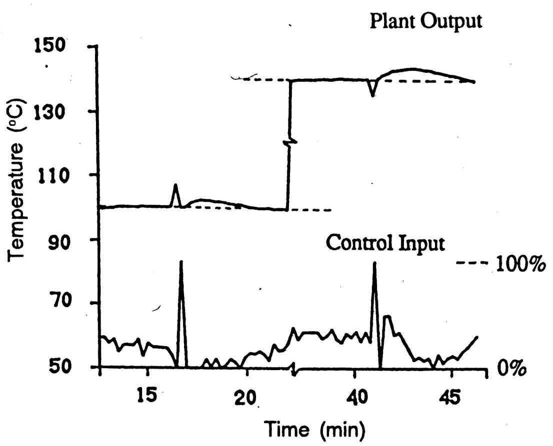

Figure 14. *Performance of the PID-controller on the furnace temperature control system under the effect of load disturbances*

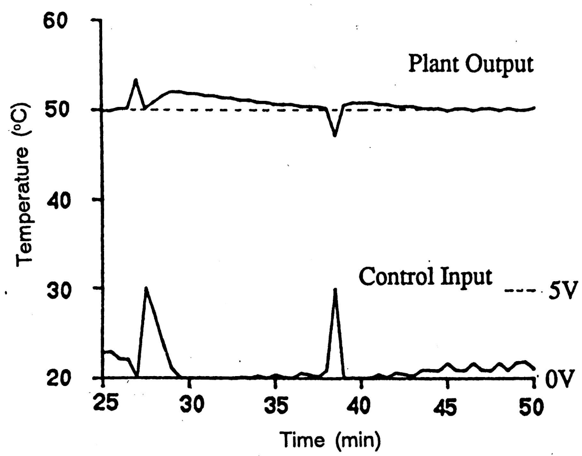

Figure 15. *Performance of the PI-controller on the water bath temperature control system under the effect of load disturbances*

From all the experiments, it can also be observed that the outputs of the neuro-controllers are much smoother in nature. On the other hand, the outputs of the conventional feedback controllers fluctuated adversely which could shorten the lifetime of the actuators. The neural network controllers show very little fluctuations at their outputs even with the occurrence of large load disturbances. The nonlinear mapping ability of the neural networks helps to smooth their outputs which could prolong actuators' life.

CONCLUSIONS

The application of neural networks in control is still in its early stage. Many control theorists and engineers are still skeptical whether neural networks can be used to provide better control solutions. In this paper, backpropagation neural networks have been developed as direct controllers for two kinds of real-time temperature control systems. The method carried out shows that the neural networks learned from the plants input-output behaviour without the use of any conventional controller or having any prior knowledge regarding dynamics. The method carried out can also be easily implemented on a wide variety of stable control systems.

A novel feature of the neuro-control schemes is their simple implementation. The same backpropagation algorithm can be used to solve two different control problems using only the plants open-loop input-output characteristics, whereas, in many traditional adaptive control approaches, prior mathematical modelling has to be done separately for each process. In addition, their inherent self-learning abilities allow the neuro-controllers to be further fine-trained on-line at the desired output values by specialized learning. In comparison to conventional feedback controllers, the neuro-controllers give improved performance and, once trained, have no requirement for tuning even when confronted with moderate load disturbances.

One drawback of the neuro-controllers is the requirement for prior training before they can be used on-line. In addition, a judicious selection of neural network models, learning rate, and the number of training patterns are essential for their success. In future when fast parallel and general purpose neural hardware, coupled with systematic procedures of selecting the neural network models, are available such problems may be overcomed. Future research efforts should test the neuro-control techniques on highly nonlinear, multivariable, and unstable real-time control systems.

ACKNOWLEDGEMENTS

The author wishes to thank the reviewers for their comments and helpful suggestions.

REFERENCES

Cybenko, G. (1988). Approximation by superpositions of a sigmoidal function. *Tech. Rep.*, **856,** University of Illinois, Dept. of Elect. and Computer Eng.

Hecht-Nielsen, R. (1989). Theory of back-propagation neural networks. *Proc. of Int. Joint Conf. on Neural Networks,* San Diego, pp 593-608.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks,* **2 (5),** pp 359-366.

Jordan, M. and Rumelhart, D.E. (1991). Internal world models and supervised learning. *Machine Learning: Proc. of the 8th International Workshop, California.*

Khalid, M. and Omatu, S. (1992). A neural network controller for a temperature control system. *IEEE Control Systems Magazine,* **12 (3),** 58-64.

Kosko, B. (1992). *In: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Prentice-Hall, New Jersey.

Miller, III, W.T., Sutton, R.S., and Werbos, P.J., Eds. (1990). *Neural Networks for Control, MIT Press, USA.*

Nagata, S., Sekiguchi, M., and Asakawa, K. (1990). Mobile robot control by a structured hierarchical neural network. *IEEE Control Systems Magazine,* **10 (3),** 69-76.

Narendra, K.S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks,* **1 (l),** 4-27.

Nguyen, D.H. and Widrow, B. (1990). Neural for self learning control systems. *IEEE Control Systems Magazine,* **10 (3),** 18-23.

Ogata, K. (1987). *Discrete-Time Control Systems,* Prentice Hall.

Pao, Y.H. (1989). Learning discriminants: The generalized perceptron. In: *Adaptive Pattern Recognition and Neural Networks,* Addison-Wesley, USA, pp 113-140.

Psaltis, D., Sideris, A., and Yamamura, A. (1988). A multilayered neural network controller. *IEEE Control Systems Magazine,* ***10 (3),*** 44-48.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning internal representation by error propagation. *In Parallel Distributed*

Processing: Explorations in the Microstructure of Cognition Vol. I, Cambridge, MA : MIT Press.

Saerens, M. and Soquet, A. (1989). A neural controller based on backpropagation algorithm. *Proc. of the First IEE Int. Conf. on Artificial Neural Networks,* London, pp 211-215.

Takahashi, Y., Chan, C.S. and Auslander, D.M. (1971). Parametereinstellung bei linearen DDC-algorithmen. *Regelungstechnik und ProzeB-Datenverarbeitung,* **19,** 237-244.

Werbos, P.J. (1991). Artificial neural networks: General capabilities and control applications. In: *IECON'90 Conference Tutorial,* Kobe, Japan.

Yabuta, T. and Yamada, T. (1991). Learning control using neural networks. In: *Proceedings of the 1991 IEEE Int. Conference on Robotics and Automation,* Sacramento, USA, pp 740-745.