MILITARY TECHNICAL COLLEGE
CAIRO-EGYPT

ICEENG 98

FIRST INTERNATIONAL CONF. ON
ELECTRICAL ENGINEERING

ADAPTIVE NEURAL NETWORK FOR REAL-TIME TRACKING CONTROL OF A DRIVE SYSTEM

R. S. Ahmed*, O. H. Abdalla**

ABSTRACT

Neural Networks are attractive alternative to the classical techniques for identification and control of complex physical systems, because of their ability to learn and approximate functions. This paper presents the development and implementation of adaptive Multilayer Neural Network (MNN) controller in real-time for a drive system. A MNN is first trained off-line to learn (identify) the inverse dynamics of the system, after the training is successfully completed, the MNN is used as a feedforward controller in the control scheme.

The advantage of the proposed controller is that the MNN is permanently training. On-line learning is applied while the system is under control to capture any system parameter variations or disturbances. Simulation results are presented to show the advantages of adaptive MNN controller compared to non-adaptive MNN controller. Also, experimental results show that the adaptive MNN controller is able to control the speed trajectory of the drive system with a high degree of accuracy, even in the presence of disturbances.

KEY WORDS

Neural networks, DC motor, Real-time control, Tracking control, Interface board.

1 INTRODUCTION

Design of conventional control systems usually involves the development of a mathematical model of the system to derive a control law. In many of the physical systems, it may be difficult to obtain an accurate mathematical model due to the presence of structured and unstructured uncertainties. Therefore application of classical or adaptive control schemes to these systems may give unsatisfactory response for trajectory tracking control.

Instead of driving a controller via the controlled process mathematically, neural network controller methodology tries to establish the controller directly from input-output data of the controlled process without the need for its mathematical model [1,2]. An advantage of using artificial neural networks (ANNs) is their ability to learn nonlinear functions whose analytic forms are difficult to derive and whose solutions are burdensome to compute. ANNs can be trained with observed data from the real plant, to reproduce the characteristics of the plant without significant prior knowledge of the plant dynamics.

* Assistant Professor, ** Professor, Dept. of Elect. Eng., Helwan University, Helwan, Egypt 11792.

At present, there is an extremely strong interest in employing ANN techniques in several control application areas such as process control, robotics, vision and pattern recognition. One of the most common used neural network in the area of system identification and control is the feedforward multilayer neural networks (MNNs). Several studies have found that a feedforward MNN with sufficient number of hidden layer neurons can approximate any nonlinear continuos function to any desired accuracy through a learning algorithm [3,4]. Back-propagation learning algorithm is the most common used for training the feedforward neural networks [5,6]. The objective of neural networks based adaptive control system for unknown nonlinear plants is to develop algorithms for identification and control using ANNs through a learning process. Some research workers [7] used the neural network for identification and control of a drive system (DC motor) using a model reference adaptive control. Although good simulation results were obtained, the MNN controller was non-adaptive (fixed weights). Also, the MNN controller was not tested for disturbances and not implemented in real-time.

In this paper, an adaptive MNN controller is developed and implemented in real-time of a drive system (DC motor, interface circuits, sensors and load). The proposed control scheme does not require the system dynamic model and its parameters. A MNN is used to identify the system inverse dynamics. The MNN is first trained off-line using the input-output data obtained from the system hardware setup. After successfully training, the MNN is used as a feedforward controller in the control scheme of the drive system. While the system is under control, on-line learning is used to update the weights of the MNN at each sampling instant to generate the appropriate control voltage for the drive system to follow a speed desired trajectory. Advantages of the adaptive MNN (using on-line learning) compared to non-adaptive MNN for controlling the drive system are presented through simulation results.

2 NEURAL NETWORKS

Artificial neural networks may be employed to represent the brain activities. The term artificial neural network is used to distinguish itself from the neural system in the human brain. There are many types of neural networks representing the brain structure and operation with various degrees of sophistication [5,6]. Artificial computing networks are far simpler than their biological counterparts because knowledge about actual brain functions is limited. No models have been successful in duplicating the performance of the human brain. Architecture of neural networks can be divided into two basic categories: feedforwrard, and feedback (recurrent) neural networks [6]. The feedforward neural network is commonly used for identification and control purpose.

2.1 Multilayer Feedforward Neural Network

The multilayer feedforward neural network contains one input layer, one or two hidden layers and one output layer. A fully connected feedforward MNN with one hidden layer is shown in Fig. 1. The hidden layer is connected to the input layer through the synaptic weights v_{ij}, which form a matrix V, and to the output layer through the synaptic weights w_{jk}, which form a matrix W. The relation between the output value, O_k, of the k^{th} output neuron and the inputs, X, to the network is given by

$$O_k = f_k\left[\sum_{j=0}^{n_h} w_{jk}\, f_j \sum_{i=0}^{n_i} v_{ij} x_i\right] \qquad \text{for } n_o \ge k \ge 1 \qquad (1)$$

where v_{ij} is the weight between the j^{th} hidden neuron and i^{th} input neuron, w_{jk} is the weight between the k^{th} output neuron and j^{th} hidden neuron, f is the nonlinear activation function of the neurons, n_i, n_h and n_o are the number of input, hidden and output layer neurons, respectively.

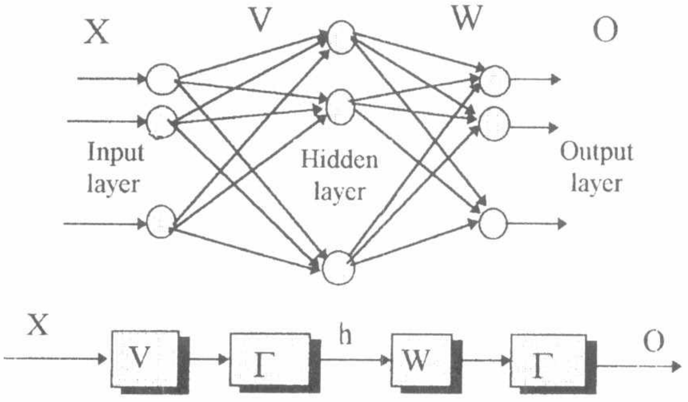


Fig. 1. A three layers feedforward neural network

The input-output mapping of a MNN can be represented by

$$O = \Gamma\left[W\ \Gamma\ [V\,X^{T}]\right].$$

where Γ is a nonnlinear matrix operator. The transfer functions of the hidden and output layer neurons are, in general, similar and non-linear. If the transfer functions of the hidden layer neurons are linear, then the multilayer neural network is equivalent to a single-layer neural network

2.2 MNN Learning

Neural network learning (training) is the process of modifying the strength of the connection weights between the network neurons. This modification leads to find an appropriate set of weights so that the error between the desired and actual output of the network is minimized for all training patterns. The network training starts by initially selecting small random weight values and presenting all training data repeatedly. The learning algorithm may be categorized as: *supervised* learning and *unsupervised* learning. Supervised learning is used for learning the feedforward neural networks. In supervised learning, a set of input and desired output patterns (X,Y) called a training data set is required. The training data should be normalized (rescaled) to lie between (0,1) or (±1) depending on the activation function of the neurons. At each instant of time when the input vector, X_i is applied, the desired response of the system is provided by a teacher (desired output), Y_i. The error signal between the desired and the actual response is used to modify neural network parameters (weights). An example of supervised learning is the backpropagation algorithm [5,6].
Also there are two different schemes for training neural networks: In the first approach which is called *global (batch)* learning, the weights are adapted after application of the whole input pattern vectors of the learning set. In the other approach referred to as *local (pattern)* learning, the weights are adapted after application of each pattern of the training set.

2.3 Adaptive and Non-Adaptive MNN

If the MNN is first trained (with a training sequence of finite length), and subsequently used with the fixed weights obtained from training, this mode of operation is referred to as "non-adaptive." Conversely, if the MNN is trained permanently while it is used (with a training sequence of infinite length), this mode of operation is referred to as "adaptive" [8].

3 SYSTEM DESCRIPTION

The simulation and experimental studies through this paper are done for a drive system composed of a permanent magnet DC motor, a power amplifier circuit to supply the power to the motor, a magnetic break for loading the motor and a tacho-meter to sense the motor speed.

3.1 System Model

The DC motor dynamics are given by the following equations

$$\frac{di_a(t)}{dt} = -\frac{R_a}{L_a} i_a(t) - \frac{K}{L_a}\omega(t) + \frac{1}{L_a} v_t(t) \qquad (2)$$

$$\frac{d\omega(t)}{dt} = \frac{K}{J_a} i_a(t) - \frac{D}{J_a}\omega(t) - \frac{1}{J_a} T_l(t) \qquad (3)$$

where ω, V_t, i_a, R_a, L_a, J_a, K, D and T_l are the rotor speed, terminal voltage, armature current, armature resistance, armature inductance, rotor inertia, torque (back emf) constant, damping constant and load torque, respectively. The armature voltage, V_t of the DC motor is supplied from the power amplifier circuit and is proportional to its control voltage V_c. If the relationship between V_t and V_c is considered a linear, and from equations (2) and (3), the relationship between the control voltage and the motor speed can be written as a difference equation as

$$V_c(k) = \beta\omega(k) + \gamma\omega(k-1) + \zeta\omega(k-2) \qquad (4)$$

where β, γ and ζ are constants and their values depend on the motor parameters and the sampling interval T. Equation (4) can also be written in the following form

$$V_c(k) = h\,(\omega(k),\ \omega(k-1),\ \omega(k-2)) \qquad (5)$$

The object of the system model is to find a relation between the system output and its input. This is useful in the next section's to determine the number of input and output layer neurons of the neural network controller.

3.2 System Hardware Setup

The hardware setup of the drive system consists of a permanent magnet DC motor, power amplifier, magnetic brake for load conditions, tacho-generator for speed sensing and personal computer (PC). An interface board (A/D and D/A converters, and digital I/O lines) is used to offer a data acquisition solution. A PC 486 AT is used for implementation of the software neural network controller. The computer processor executes the software controller program which is written in C language. Data transfer from and to the drive system is accomplished via the PC memory. The digital inputs from the system interface to the software controller are taken through the A/D converter, while the analog outputs from the software controller to the system interface are taken through D/A converter of the interface board.

4 MNN IDENTIFICATION OF THE SYSTEM INVERSE DYNAMICS

A MNN is used to identify the unknown system dynamics (DC motor, amplifier circuit, sensor and load) that maps the control voltage V_c to the motor speed ω. Because the MNN is used to identify the inverse dynamics of the system, the inputs to the MNN is the system output (motor speed), and the output from the MNN is the control voltage. The MNN is trained to emulate the unknown function, $h(\bullet)$ in equation (5), so the inputs to the MNN are $\omega(k)$, $\omega(k-1)$ and $\omega(k-2)$ and the MNN output, $V_{nn}(k)$ is the estimate of the control voltage $V_c(k)$. The MNN consists of three layers, 3 neurons in the input layer, 7 neurons in the hidden layer and one neuron in the output layer. The normalized training data obtained from the hardware setup is used to train the MNN off-line. The weights of the MNN were randomly initialized with a small values (± 0.2). The error signal, e(k), between the desired output, $V_c(k)$, and the network output, $V_{nn}(k)$, is used to update the network weights during the learning process through the back-propagation learning algorithm [5,6]. Figure 2 shows the configuration of the system identification using a MNN. The estimate of the system inverse dynamics is contained in the MNN. After the training of the MNN is completed, it is tested with a test data (not contained in the training data). The output of the MNN was found to be very close to the desired output.

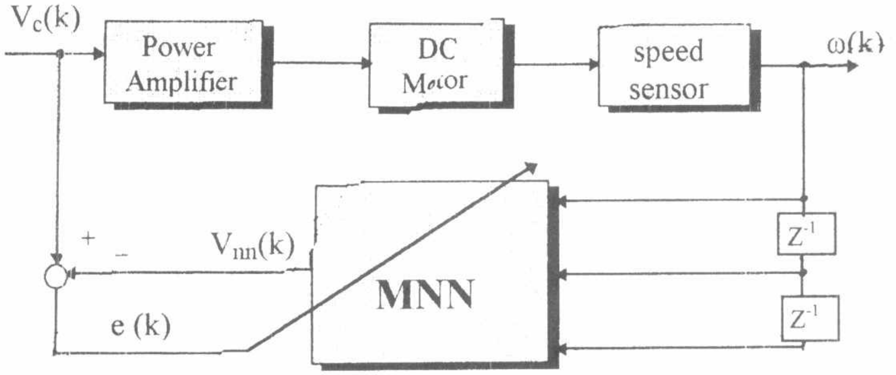


Fig. 2. System identification using a MNN

5 ADAPTIVE MNN CONTROLLER

The objective of the motor control system is to drive the motor so that its speed follow a prespecified desired trajectory. The basic control scheme consists of the trained MNN which is used as a feedforward controller and a fixed gain feedback controller as shown in Fig. 3. A feedback controller is used in the feedback loop to serve the on-line learning and to achieve low errors during transient. On-line learning is used to tune the weights of the MNN to capture any variations of system parameters and disturbances. The system control voltage, V_c, is composed of the output of the feedforward controller, V_{nn}, and the output of the feedback controller, V_p.

If the MNN is trained well, then the estimate and actual system inverse dynamics are very close, and the MNN controller alone provides all the necessary voltage for the motor to track the desired speed trajectory and the output of the feedback controller will be almost zero. If there is a mismatch between the estimate of the system inverse dynamics and the actual system inverse dynamics, on-line learning is used to adjust the weights of the MNN to generate the appropriate voltage required for a desired trajectory.

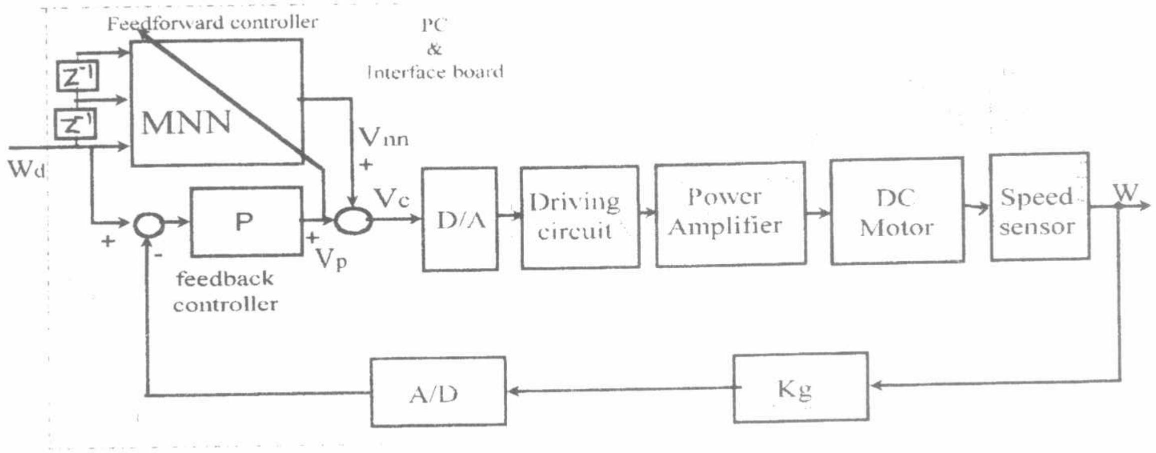


Fig. 3. Adaptive MNN controller for a drive system

6 SIMULATION RESULTS

Before implementing a real time controller, it is preferred, if possible, to employ a simulator rather than the actual process to be controlled. Dynamic simulations facilitate better understanding of the controlled dynamic process and provide insights into the nature of the interactions between the inputs and outputs. In addition, performance of the proposed controller can be investigated by performing tests on dynamic simulators. So simulation results of the proposed adaptive MNN controller for the drive system is performed first. Also, the advantages of adaptive MNN controller compared to non-adaptive is presented through simulation results. Several motor speed trajectories were tested for adaptive and non-adaptive MNN controllers. For brevity, only the following trajectory is presented. Figure 4(a) shows the desired and simulated speed trajectories with non-adaptive (fixed weights) MNN controller when a sudden disturbance is applied at time 3.5 sec and removed at 10.5 sec. The system control voltages are shown in Fig. 4(b). It can be seen that the actual system output does not follow the desired output very close. This is due to the fixed weights of the MNN controller. To show the effectiveness of the on-line learning, the same previous trajectory with the same disturbance is repeated with adaptive MNN controller. Figure 5(a) shows the desired and simulated motor speed trajectories. The system control voltages in this case are shown in Fig. 5(b). It can be seen from this figure that the simulated and desired output trajectories are very close even in the case of disturbances.

7 EXPERIMENTAL RESULTS

The capability of the MNN controller for tracking control of the DC motor was tested in the laboratory by applying different speed trajectories. To ensure a robust performance, the control scheme was also tested for sudden disturbances. The motor system was first tested without controller (open-loop). Figure 6(a) shows the desired and actual motor speeds when a desired constant speed of 1200 rpm and a sudden disturbance is applied to the motor system at time 5.2 sec and removed at 9.2 sec. It is shown from this figure that the speed is dropped from 1200 rpm to 780 rpm during the period of disturbance. A photograph of the actual motor speed and the control voltage is shown in Fig. 6(b).

Figure 7(a) shows the desired and actual motor speeds of the previous case with the adaptive MNN controller. It can be seen from these figures that the actual motor speed follows the desired speed very

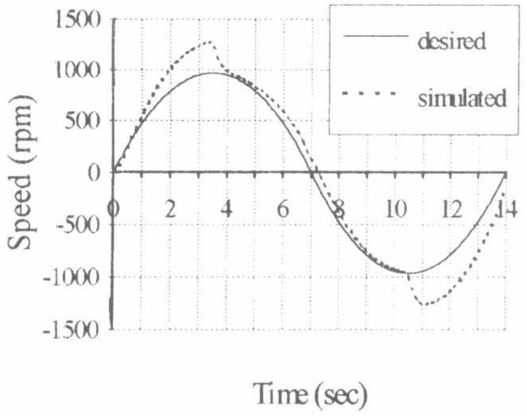


Fig. 4(a). Desired and simulated speeds (non-adaptive MNN controller)

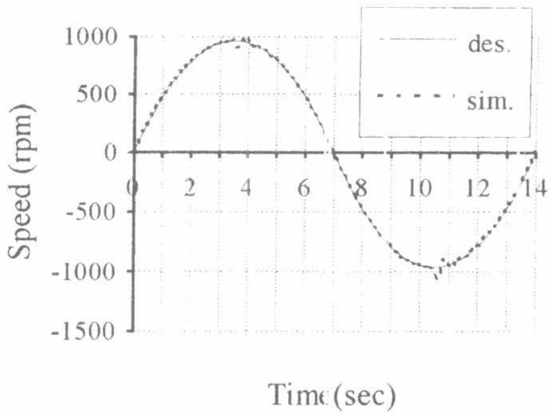


Fig. 5(a). Desired and simulated speeds (with adaptive MNN controller)

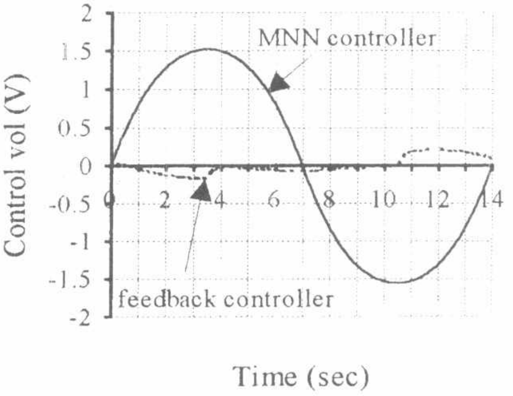


Fig. 4(b). System control voltages

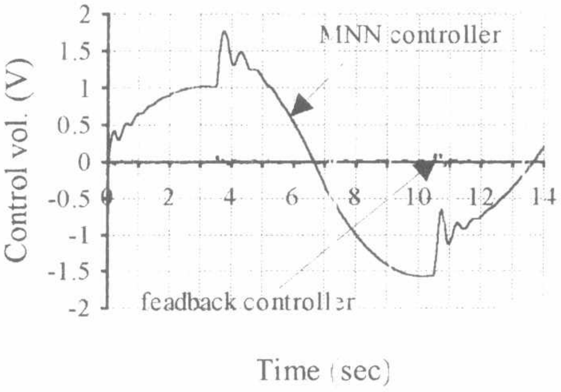


Fig. 5(b). System control voltages

closely, and the control system is able to handle the sudden disturbance applied to the motor system. The output of the MNN controller changes in response to the disturbance input whereas the output of the feedback controller is approximately zero except at the starting and at the instant of applying the disturbance. The MNN controller provides almost all the voltage to the motor system to track the desired trajectory as shown in Fig. 7(b). A photograph of the actual motor speed and the control voltage is shown in Fig. 7(c).

The system response to a parabolic speed trajectory with disturbance applied at time 3.3 sec and removed at time 10.3 sec is shown in Fig. 8. It can be seen from Fig. 8(a) that the actual motor speed follows the desired speed trajectory very closely. Figure 8(b) shows the system control voltages. It can be seen from this figure that the MNN controller provides almost all the control input voltage to the motor. The control input voltage provided by the feedback controller is very small compared to the MNN controller and changes at the instants of applying and removing the disturbance. A photograph of the actual motor speed, position and the control voltage is shown in Fig. 8(c). It can be seen that the actual motor speed trajectory follows the desired speed trajectory very well. It is shown from these results that the adaptive MNN controller is efficient for trajectory control of the drive system.

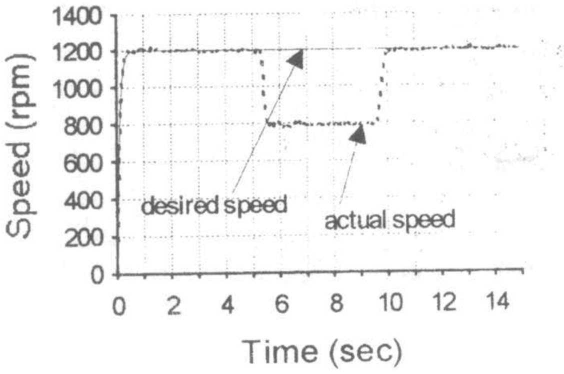


Fig. 6(a). Desired and actual motor speeds without controller.

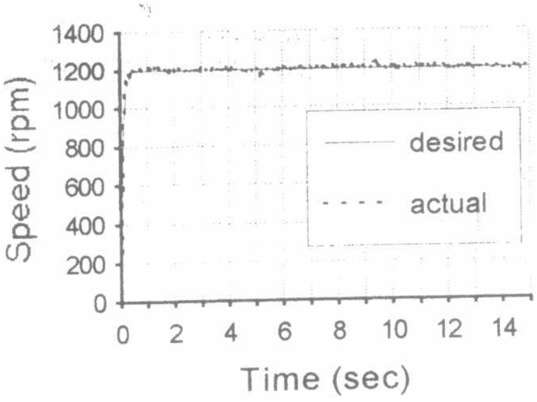
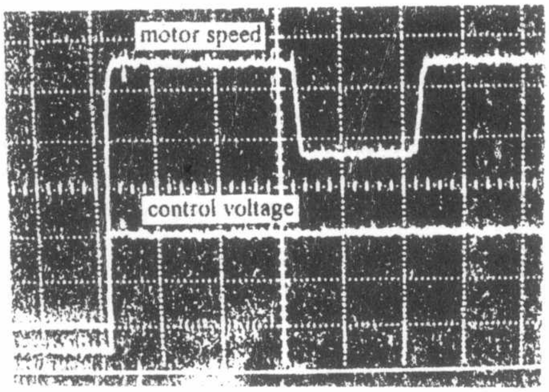


Fig. 7(a). Desired and actual motor speeds with MNN controller.



1 div. = 2 sec., 200 rpm, 0.5 volt

Fig. 6(b). Photograph of the actual motor speed and control voltage

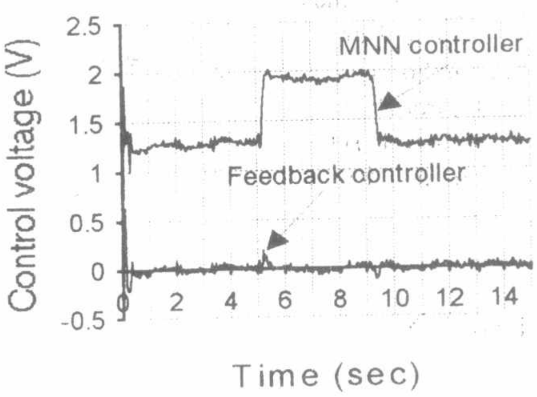


Fig. 7(b). Motor control voltages

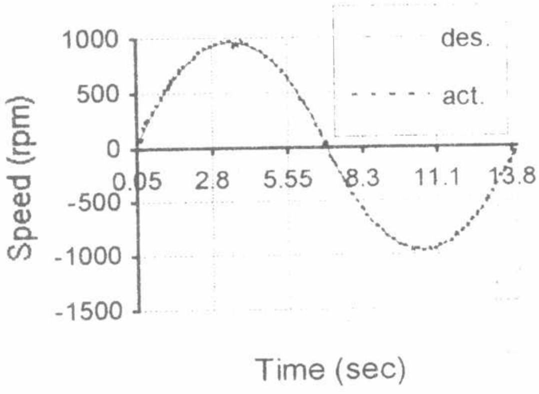
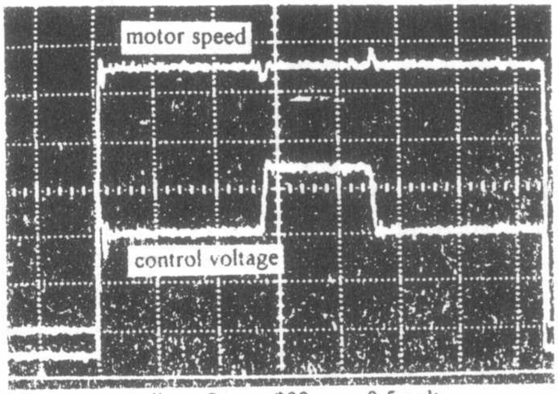
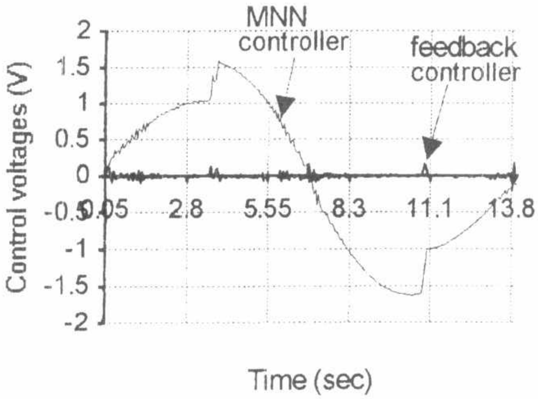


Fig. 8(a). Desired and actual motor speeds

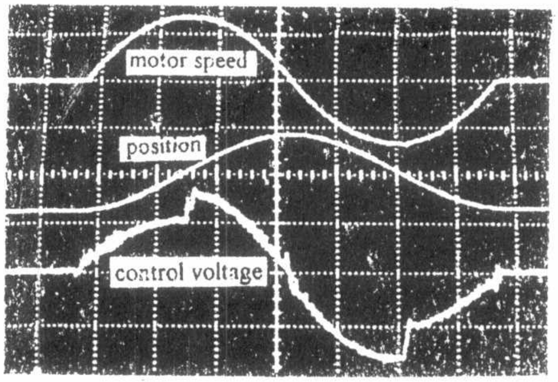


1 div. = 2 sec., 200 rpm, 0.5 volt

Fig. 7(c). Photograph of the actual motor speed and control voltage



8(b). Motor control voltages



div. = 2 sec., 740 rpm, 1 volt

Fig. 8(c). Photograph of the actual motor speed, position and control voltage

8 CONCLUSIONS

This paper presents the development and implementation of a real-time adaptive MNN controller for a drive system (DC motor). The advantages of this scheme is that it does not require the system dynamic model and its parameters and therefore treats the system as a black box. While the system is under control, on-line learning is used to adjust the weights of the MNN to capture any system parameters variation and/or external disturbances. The advantages of the adaptive MNN over non-adaptive is presented through simulation results. It is shown by simulation and experimental results that the proposed control scheme with the adaptive MNN is efficient for trajectory tracking control of the drive system with high degree of accuracy even in the case of sudden disturbance.

REFERENCES

[1] Narendra K. S. and Parthasarathy K., "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Transactions on Neural Networks, vol. 1, pp. 4-27, March 1990.

[2] Ahmed R. S., Rattan K. S. and Abdallah O. H., "Adaptive Neural Network For Identification And Tracking Control Of A Robotic Manipulator," IEEE National Aerospace & Electronics Conference, Dayton, OH. pp. 601-609, May 1995.

[3] Bulm E. K. and Li L.K., "Approximation Theory and Feedforward Networks," Neural Networks, vol 4, pp. 511-515, 1991.

[4] Hornik W. K. and Stinchombe M., "Multilayer Feedforward Networks Are Universal Approximators," Neural Networks, Vol. 2, pp. 359-366, 1989.

[5] Zurada J. M., Introduction to Artificial Neural Systems, West Publishing Company, 1992.

[6] Haykin S., Neural Networks: A Comprehensive Foundation, Macmillan College, 1994.

[7] Weerasooriya S. and El-Sharkawi M. A., "Identification and Control of a DC Motor Using Back-Propagation Neural networks," IEEE Trans. On Energy Conversion, Winter Meeting, pp. 1-7, 1991.

[8] Nerrand O., et. al. "Training Recurrent Networks: Why and How? An Illustration in Dynamical Process Modeling," IEEE Trans. On Neural Networks, Vol. 5, No. 2, March 1994.