ADAPTIVE NEURAL NETWORK CONTROL OF A DC MOTOR

Mehmet Karadeniz — mekara@gazi.edu.tr
İres İskender — iresis@gazi.edu.tr
Selma Yüncü — syuncu@gazi.edu.tr

Gazi University, Faculty of Engineering & Architecture, Department of Electrical & Electronics Engineering, 06570 Maltepe, Ankara TURKİYE

Keywords : Neural Networks, dc motor, control

ABSTRACT

This paper presents a Multilayer Neural Network controller for real time control applications. A model reference structure is developed and a neural network is used as a compensator in the closed loop system. This scheme can be used in the control of nonlinear systems and/or as an adaptive controller if desired.

I. INTRODUCTION

In the industrial processes there are many systems having nonlinear properties. Moreover, these properties are often unknown and time varying. The commonly used Proportional-Integral-Derivative (PID) controllers are simple to be realized, but they suffer from poor performance if there are uncertainties and nonlinearities. The neural network controllers have emerged as a tool for difficult control problems of unknown nonlinear systems. Neural networks (NN) are used for modeling and control of complex physical systems because of their ability to handle complex input-output mapping without detailed analytical models of the systems. Since Multilayer Neural Networks (MNN) can approximate arbitrary nonlinear mapping through a learning mechanism, they can compensate the nonlinearities.

There are several control strategies for neural networks which some of them are as: 1) Feedforward control, 2) Direct inverse control (extracting inverse dynamics), 3) Indirect adaptive control method based on NN identification, 4) direct adaptive control with guarantied stability, 5) Feedback linearization, 6) Predictive control [1,2]. In the direct inverse control method, a MNN is trained by specialized back-propagation algorithm [2]. This method has attracted much attention in recent years because it is intuitive, and simple to be implemented [3,4,5]. However, the plant to be controlled may not have a unique or stable inverse, which is the drawback of direct inverse dynamics method. Besides, if all the poles and zeros of the system to be controlled are negative, direct inverse controller performs well. But the control will not be successful if any pole or zero of the system is positive.

In this study, a control strategy is proposed for the real time control of single-input single-output linear and nonlinear systems. At first glance, the proposed method resembles previous two methods mentioned before: model reference direct inverse control and indirect adaptive control. The proposed method uses specialized back-propagation training algorithm and is desired to track a reference model. The method can be considered as a direct controller design. Although it does not require any knowledge of the system dynamics, it requires the sensitivity of the controlled system. The proposed method can control systems which has nonnegative poles. However it cannot control systems having nonnegative zeros. In addition, it needs faster computing for stabilization comparing to direct inverse method.

II. THE NEURAL NETWORK CONTROLLER STRUCTURE AND SYSTEM IDENTIFICATION USING NEURAL NETWORK

In using neural networks for system identification, training data can be obtained by observing the input-output behavior of a plant. If previous values of the

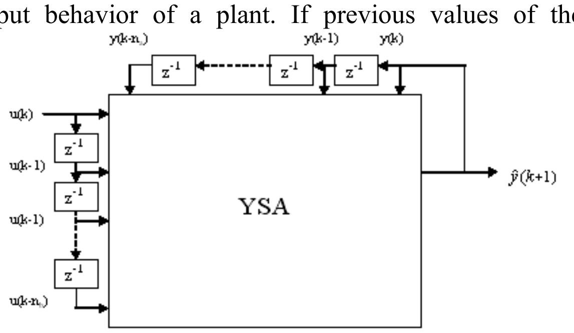

Figure 1. System Identification by TDNN

input and output are present, the future value can be predicted as

$$\hat{y}(k+1) = f[\vec{y}(k), \vec{y}(k-1), .. \vec{y}(k-n_a), \vec{u}(k), \vec{u}(k-1), .. \vec{u}(k-n_b)] \quad (1)$$

which is called as “one step ahead prediction”. Equation (1) can be easily implemented as shown in figure 1 [6,7]. Previous values of the input and output held by time delay elements give dynamic behavior to the network. This structure is called as time delayed neural network (TDNN). In this paper, all controller and identifier NNs in the systems are time delayed neural networks. Therefore, the proposed NN controller has the same structure as shown in figure 1.

III. DIRECT MODEL REFERENCE ADAPTIVE NN CONTROLLER

The proposed method can be used as adaptive or non-adaptive controller. If learning process continues, the controller will be an adaptive controller. In non-adaptive case, learning process is executed as offline or for a certain period of time. Figure 2 shows the usage of NN controller after completing the training for the non-adaptive case.
This structure is different from the direct inverse dynamics control structure in which error is directly the input of structure. The objective is to train the neural network in such a way that to obtain a controller to control the plant (figure 2).

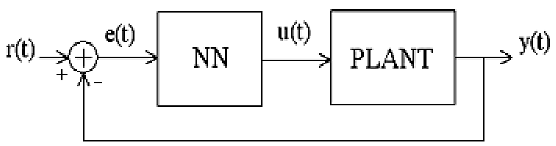

Figure 2. MNN used as a controller (Plant : controlled system).

To achieve this, the neural network should be trained in such a way that for the input of error, e(t) produces proper control parameter, u(t) to be applied to the plant to produce y(t). In standard back-propagation algorithm, both u(t) and y(t) are required for training the network. Therefore, a special learning algorithm is required.

Let there is a reference model having input of $r(t)$ and desired output signal of $d(t)$. The error between the output of the controlled system, y(t) and the reference model output can be defined as

$$e_c(t) = d(t) - y(t) \tag{2}$$

The objective is to obtain an output signal from the system shown in figure 2 as close as possible to the output of the reference model by reducing the square of the error. Therefore the cost function is defined as

$$E = \frac{1}{2} e_c(t)^2 \tag{3}$$

For the output layer of the network, which has one neuron, the training rule for specialized back-propagation can be written as

$$\Delta w_i = -\eta \frac{\partial}{\partial w_i} E \tag{4}$$

where

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial u(t)} \frac{\partial u(t)}{\partial net} \frac{\partial net}{\partial w_i} \tag{5}$$

In this equation $net = \sum_k x_k w_k$ is the sum of weighted input signal of the activation function, x_k is the kth input, w_k is the kth weight of the output neuron. The right side terms of Equation (5) can be shown as the following equations.

$$\frac{\partial u(t)}{\partial net} = a'(net)$$

$$\frac{\partial net}{\partial w_i} = x_i(t)$$

$$\frac{\partial E}{\partial u(t)} = \frac{\partial}{\partial u(t)} \frac{1}{2} \left[d(t) - y(t) \right]^2$$

$$= -\left[d(t) - y(t) \right] \frac{\partial y(t)}{\partial u(t)} \tag{6}$$

$a'()$ is the derivative of the activation function of the output neuron. Thus,

$$\Delta w_{ij} = \eta a'(net) x_i(t) \left[d(t) - y(t) \right] \frac{\partial y(t)}{\partial u(t)} \tag{7}$$

The updating rule for the output layer is obtained. In equation (7), $d(t) - y(t)$ is the error between the controlled system and reference system. Error term between the output of the system and the reference signal $e(t)$ is implicit to this equation, because $e(t)$ is the input of NN and calculations performed on it at the hidden layer.

Hidden neurons have the same updating rule as in standard back-propagation [8]. The only difference between equation (7) and standard back-propagation is the term $\partial y(t) / \partial u(t)$, which is the sensitivity of the controlled system. This term may not be obtained directly from the controlled system itself accurately since it is time independent. One way to obtain the sensitivity term is to identify the controlled system by a neural network, because the identified model provides the stationary properties of the system. If the controlled system is linear, its sensitivity is a constant value. Since learning rate of

NN included in equation (7) is an arbitrary or adjustable constant, two constant parameters can be considered as one constant parameter, therefore identification is not required for linear systems.

The proposed training structure is shown in figure 3. Note that the error signal $e(t)$ is the only input of the neural network controller. This feature makes the controller different from the direct inverse dynamics control and also renders to control nonzero pole systems successfully.

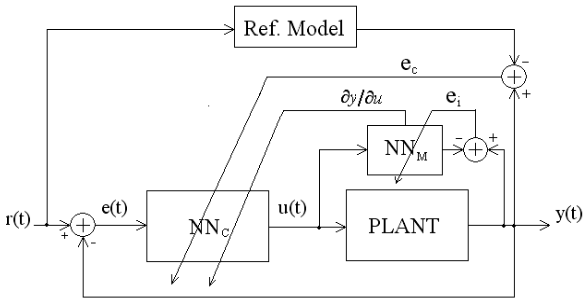

Figure 3. Model reference adaptive neural network controller

The structure given above can be used to train the controller network. In addition, if the training process continues, the proposed structure becomes a model reference adaptive controller.

IV. SPEED CONTROL OF A DC MOTOR BY PWM TECHNIQUE

A separately excited motor is required to be controlled to track the speed command in one direction by pulse width modulation technique. The controlled system is shown in figure (4).

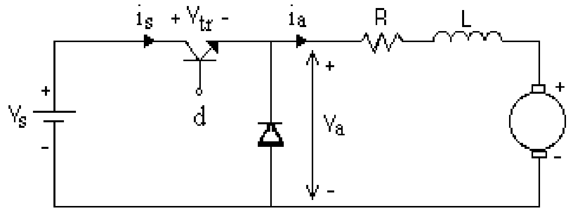

Figure 4. PWM Controlled DC Motor

Speed of the motor is controlled by controlling power delivered to the motor, while keeping the field voltage and field current of the motor constant. The chopper is used to control the armature voltage of the motor. Power is delivered to the system from a constant voltage source V_s. The circuit has a power transistor and a power diode called as "freewheeling diode". During the on state of transistor, power is delivered to motor. When the transistor is off, armature current of the motor continues to flow on freewheeling diode. To vary the power applied to the motor a square wave signal is applied to the transistor gate. The applied signal has constant frequency but variable on-off durations in each period. The speed of the motor is controlled by varying the duty cycle d of the transistor. The corresponding waveforms are given in figure 5.

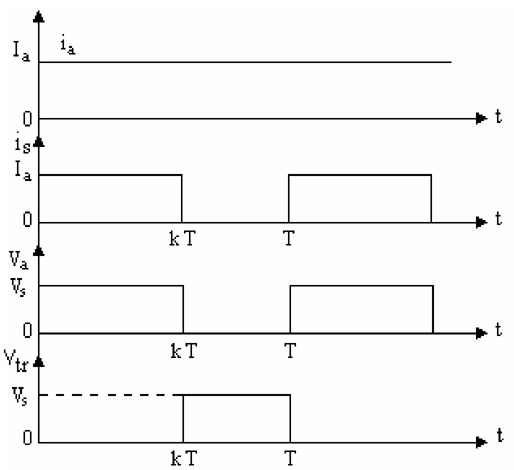

Figure 5. Voltage and current waveforms of the system

For a lossless converter the power delivered to the system is proportional with the duty cycle as

$$P = dV_a I_a \tag{8}$$

where V_a and I_a are armature voltage and current of the motor. The average value of the armature current is as follows:

$$I_s = dI_a \tag{9}$$

The equivalent input resistance of the chopper converter seen from the source side is as:

$$R_{eq} = \frac{V_s}{I_s} = \frac{V_a}{I_a}\frac{1}{d} \tag{10}$$

The maximum peak-to-peak current ripple is

$$\Delta I_{\max} = \frac{V_s}{R}\tanh\frac{R}{4fL} \tag{11}$$

where f is the sampling frequency [9].

The aim is to find a proper value for duty cycle to make motor tracks the desired speed. Neural Network Controller extracts the desired duty cycle signal during its learning.

V. SIMULATION RESULTS

Simulation programs are written in C++ language. In this study, 110 V, 2.5 hp, 1800 rpm separately excited DC motor having the following parameters are used: R_a = 1 Ω, L_a = 46 mH, J = 0.093 kgm^2, B = 0.008 Nt-m/rad/s,

K_v = 0.55 V/rad/s. The other parameters used in the simulation are as follows: Amplitude of PWM armature voltage 200V (i.e., V=200V), reference speed during training process w_r = 100 rad/s, load torque $T_L = 10Nm$. The chopping frequency of the transistor is 1kHz.

Figure 6 shows some of the uncontrolled dc motor variables operated at duty-cycle of 0.5.

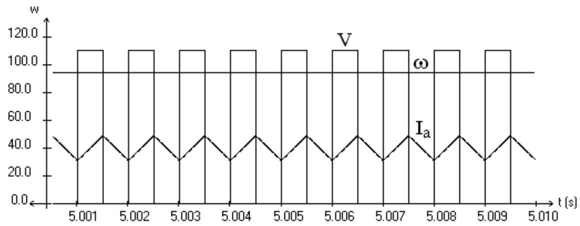

Figure 6. Input voltage, speed and armature current of Motor. Armature current has been multiplied by 30

The diagram of neural network used in the control of DC motor is shown in figure 7. The time delay neural network is a multilayer neural network and has two inputs. The MNN has two hidden layers, which consist of 10 neurons. The output layer of the NN has only one neuron. Since the motor model is linear, identification is not required.

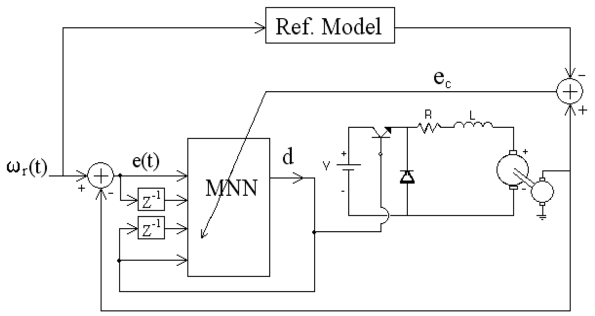

Figure 7. Control Schema of DC Motor

The network is trained for 250 second applying a 0.1 Hz square wave to the reference model. The load was 10 Nm during training. First the training was completed then off line performance of the controller was tested for different waveforms of reference signal and different load conditions. The step pulse response at 10 Nm load is given in figure 8. The responses of the motor to a sinusoidal input signal are given in figures 9 and 10 for two different load torques which are zero and 15 Nm. The responses of the motor to a triangular input signal are given in figures 11 and 12 for the same load conditions as figures 9 and 10. To represent on the same scale, the controller outputs –duty cycles- for each condition shown in figure 8-12 are multiplied by 20.

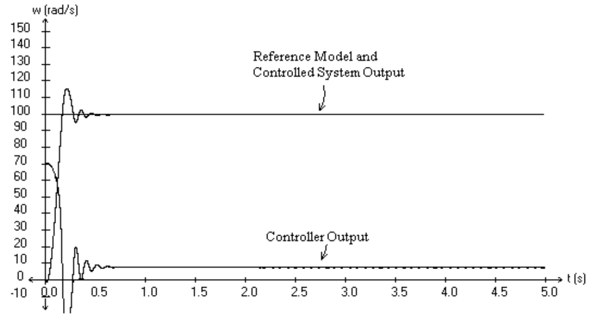

Figure 8. 100 rad/s step response of DC Motor at Load 10 Nm

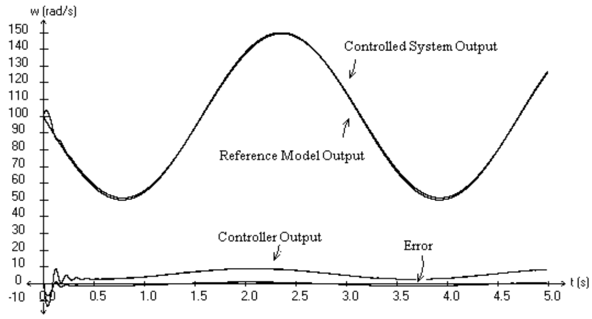

Figure 9. Sinusoidal waveform response of DC Motor at no load

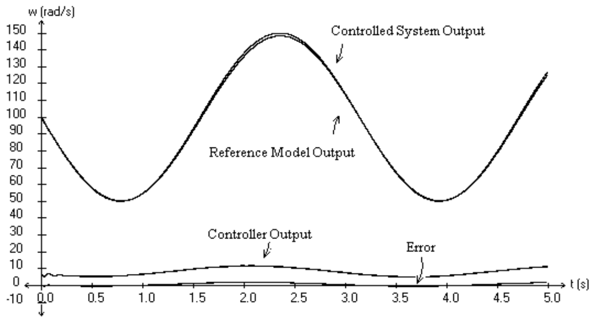

Figure 10. Sinusoidal waveform response of DC Motor at load 15 Nm

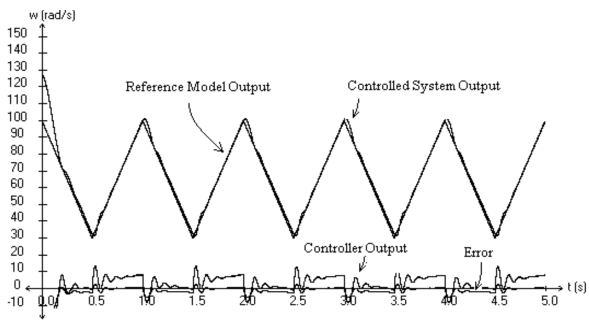

Figure 11. Triangular waveform response of DC Motor at no load

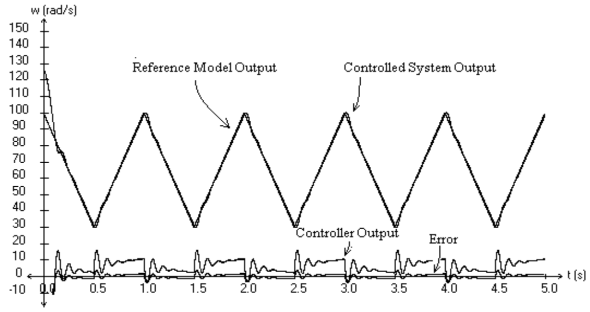

Fig 12. Triangular waveform response of dc motor at load 15 Nm

VI. CONCLUSIONS

This paper presents an NN compensator to make motor tracks the desired speed. The controller consists of a multilayer perceptron and time delay elements. The specialized back-propagation algorithm was used for the NN controller. Simulation results show how well the control method performs.

In addition, this control method can be used as an adaptive controller if training mode of the neural network continues. The proposed controller will perform well if sampling rate of training (or weight updating frequency of) is high enough and the zeros of the system are nonnegative.

Thanks to Scientific Research Unit of Gazi University for its support to prepare this study and the implementation of the system that is under processing.

REFERENCES

1. Choi Y. K., Lee S. K., Kay Y.C., Design and Implementation of an Adaptive Neural Network Compensator for Control Systems. IEEE Transactions on Industrial Electronics, vol 48, no 2, April 2001

2. Magnus N., Neural networks for modelling and control of dynamic systems. Springer, Berlin. New Yor 2000

3. Psaltis D., Sideris A., Yamamura A.A. Multilayer Neural Network Controller. IEEE Control System Magazine. 8(2) 17-21, 1988.

4. Saerens M., Soquet A., Neural Controller Based on Back-Propagation Algorithm. IEE Proceedings-F, Vol. 138, No 1, February 1991.

5. Ahmed R.S., Kuldip S.R., Khalifa H.K., Real-Time Tracking Control of a DC Motor Using a Neural Network. IEEE 1995.

6. Cichocki A., Unbehauen R., 1996, Neural Networks for Optimization and Signal Processing John Willey & Sons

7. K. S. Narendra and K. Parthasarathy, Identification and Control of Dynamical Systems using Neural Networks, IEEE Transaction on Neural Networks, Vol.1, pp 4-27, March 1990.

8. Haykin S., 1994, Neural Networks, Macmillan Collage Publishing Company

9. Rashid M. H., Power Electronics, Prentice-Hall 1988.