JPE 5-1-1

A Comparison of Different Intelligent Control Techniques For a PM dc Motor

S. I. Amer† and M. M. Salem*

†*Electronics Research Institute, Cairo, Egypt

ABSTRACT

This paper presents the application of a simple neuro-based speed control scheme of a permanent magnet (PM) dc motor. To validate its efficiency, the performance characteristics of the proposed simple neuro-based scheme are compared with those of a Neural Network controller and those of a Fuzzy Logic controller under different operating conditions. The comparative results show that the simple neuro-based speed control scheme is robust, accurate and insensitive to load disturbances.

Keywords: Permanent Magnet dc Motor- Fuzzy Controller- Neural Network – Single Neuron

1. Introduction

Recent developments in microprocessors, magnetic materials, semiconductor technology, and mechatronics provide a wide scope of applications of high–performance electric motors in various industrial processes. In high–performance motor drive applications involving mechatronics, such as robotics, rolling mills, machine tools, etc., accurate speed and position control is of critical importance. Although relatively expensive, dc motors are still widely used in such applications because of their reliability and ease of control. This is due to the decoupled nature of the field and armature mmf's of the two types of commonly used dc motors, separately excited and PM dc motors. The latter does not require an extra dc supply for the field, as the permanent magnet itself acts as the source of the flux. The PM dc motor is, thus, compact in size, robust, and highly efficient. Due to their high initial torque and efficiency, PM dc motors have increasing application areas, especially in load systems with power ratings ranging from a few watts to about 100 KW.

In high–performance drive applications, the control of a PM dc motor demands special attention. It has to have faster response, quicker recovery of speed from load impact and insensitivity to parameter variations. Conventional designs of robust control are often associated with constant gain controllers, such as PI or PID(proportional integral or proportional integral derivative), which stabilize a class of linear systems over a small range of system parameters. Moreover, these types of controller–based systems require accurate mathematical models to describe the system dynamics[1]. It is often quite difficult to obtain an accurate system model. Even if the model of the drive under control is obtained, unknown conditions, such as saturation, disturbances, parameter variations, noise, etc. are unavoidable and cannot be modeled accurately. Furthermore, the load to the motor is

Manuscript received June 8, 2004; revised September 21, 2004.
† Corresponding Author: amersanaa@yahoo.com
Tel: +202-3310551, Fax: +202-3351631 ,ERI
* Electronics Research Institute, Cairo, Egypt

uncertain and shows nonlinear mechanical characteristics, which may cause the drive system to become unstable. In recent years, many adaptive control techniques, such as model reference adaptive control(MRAC), sliding mode control(SMC), variable structure control, and use of a self-tuning regulator have become available to control systems that are less stable. These conventional adaptive control techniques are usually based on system model parameters.

Most of the adaptive control techniques for nonlinear systems are often associated with linearizing the model for a specific operating time interval and applying linear control theories. This introduces considerable errors because of the linearization of the nonlinear model. Real-time implementation is often difficult and sometimes not feasible because of the use of a large number of parameters in these adaptive schemes [2].

Recently, artificial neural networks(ANN) have proved extremely useful in pattern recognition, image processing and speech recognition. These networks are also receiving wide attention in control applications. When used as a motor controller in real time, an ANN can tune itself through on–line training and instruct the motor drive system to perform as desired.. The use of an ANN in high performance motor drives can make the systems robust, efficient, and immune to undesired operating conditions [3]. A novel a novel speed control strategy of a PM dc motor has been proposed [4], which incorporates an on-line weights and biases updating feature of the ANN. The ANN architecture was based on the inverse dynamic model of the nonlinear drive system. To enhance robustness, which is an important criterion of a high–performance drive, a unique feature of adaptive learning rates was also introduced. Stability over a wide operating range was obtained using an ANN structure with a local feedback provision [4]. A real-time implementation of ANN-based speed control of a dc motor drive has also been introduced [5]. Reference speeds have been arbitrarily fed to the ANN as inputs without using a stable reference model. Although the drive system stability has been improved by providing a feedback loop, the evaluated system responses have considerable amounts of speed overshoots under some operating conditions. The learning rate was fixed during the on-line weights and biases updating. So, there was a need to design an efficient and stable on-line self-tuning ANN–based dc motor drive system with an adaptive learning rate feature[5].

During the last decade, fuzzy logic controllers (FLC) have attracted great attention from both the academic and industrial communities. Recently, the use of FLC's has been suggested as an alternative approach to conventional control techniques for complex control systems such as nonlinear or time delayed system. That is, the design of a FLC does not require a mathematical description of the control system and a FLC can compensate for the environmental variation during operation. However, we cannot obtain a good control performance if the membership functions, fuzzy rules and scaling factors are incorrect. Recently, membership functions, fuzzy rules and scaling factors have been determined by evolutionary computations, which rely on a probabilistic search method based on genetics and evolutionary theory.

It is known that the torque levels of loads such as rollers, carrying bands, crains, lifts, and conveyors vary continuously. Therefore, the torque–speed characteristics of these types of loads also vary depending on the variations in torque. However, in many systems, the speed is required to be constant. The speed and load estimation are carried out by measuring voltage and current instead of using speed and torque sensors, which is detailed in [6]. The current, voltage and speed data obtained previously for different load levels including the no-load case, is stored in a look-up table for fuzzy control. It then is used to find the type and operating point of an unknown constant type load. This operating point gives information about the current, voltage and speed levels of the PM dc motor. T Then applying this data about the speed and voltage, the motor is operated at desired speed and voltage level by using fuzzy logic control rules.

In this paper, the performance of a PM dc motor is studied under different operating conditions using a Neural Network controller, a Fuzzy logic controller and a Single Neuro controller. A comparison between the three control algorithms is carried out and discussed. The permanent magnet dc motor drive system dynamics are described in section 2. In section 3, the fuzzy logic controller is described. The structure of the Neural Network controller is introduced in section 4. The single neuro controller is presented in section 5. The simulation results and comparison between the three control methods are shown in section 6.

2. PM dc motor Drive System dynamics

The dynamics of the PM dc motor drive system can be described by the following equations:

$$v_a(t) = R_a\, i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \tag{1}$$

$$E_b(t) = K_e\, \omega_r(t) \tag{2}$$

$$T_e(t) = K_t\, i_a(t) \tag{3}$$

$$T_e(t) = J \frac{d\omega r(t)}{dt} + B\,\omega_r(t) + T_L(t) + T_f \tag{4}$$

where: $v_a(t), E_b(t), and\, i_a(t)$ are the time–varying motor terminal voltage, back EMF, and armature current, respectively, $\omega_r(t)$ is the motor speed, R_a and L_a are the armature resistance and inductance, $K_e\, and\, K_t$ are the motor back emf and torque constants respectively, $T_e(t), T_L(t), and\, T_f$ are the developed torque, load torque, and frictional torque respectively, and J and B are the inertia and viscous constants respectively. The motor parameters are given in Appendix A.

Since the motor field is permanent, the armature voltage is the only parameter which can be varied to control the drive system.

3. Fuzzy logic control of the PM dc motor

Recently, fuzzy logic control(FLC) has become a popular research area in the control engineering. This is due to its ability to be used with systems that are complex or when it is difficult to develop a mathematical model. Fuzzy logic control has found a wide range of application areas such as photovoltaic energy conversion, home heating systems, process control, system identification, expert systems, robotics, pattern recognition and man–machine interface systems. FLC can be regarded as a set of heuristic decision rules that include the experience of a human operator.

The main advantages of FLC are;

1- No mathematical formulation of the system is needed.
2- Linguistic variables and approximate reasoning are used to describe inexact objects and achieve multi objective control.

The fuzzy controller relates significant and observable variables to the control actions, and consists of a fuzzy relationship or algorithm. The input error time sequences such as error and change in error are converted to fuzzy variables. These variables are evaluated by the control rules using the compositional rule of inference, and the approximately computed control action is then reconverted to the crisp value required to regulate the process. So the essential elements in designing fuzzy controllers include:

1- Defining input and output variables.
2- Converting the input variables to fuzzy sets.
3- Determining the fuzzy control rules.
4- Reconverting the fuzzy control actions into crisp control actions.

Fig. 1 shows a block diagram for a fuzzy logic control system. The error signal e(k) and its rate of change de(k) are selected as inputs to the fuzzy logic controller. These are normalized into a common universe of discourse and their linguistic fuzzy subsets along with their membership grades are then defined using the functions. The fuzzy membership grades of the control input change in the fuzzy subsets are obtained based on the rules given by Table 1. Then this normalized value of control input change is reconverted back to its actual level.

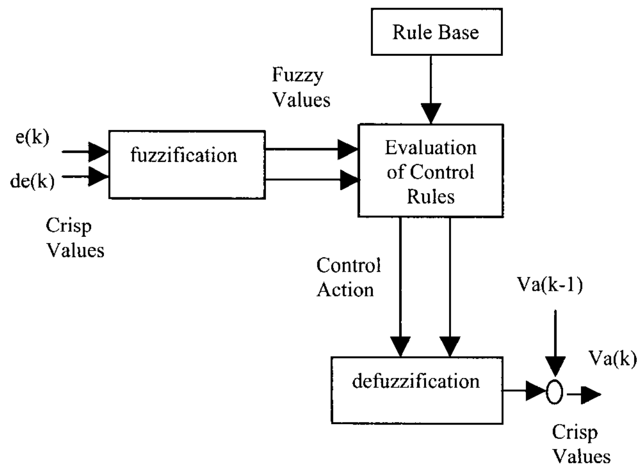


Fig. 1 FLC Block diagram

Table 1 Fuzzy Control Rule Decision Table

de(k) e(k)	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PB	PB	PM	PS	ZE
NM	PB	PB	PB	PM	PS	ZE	NS
NS	PB	PB	PM	PS	ZE	NS	NM
ZE	PB	PM	PM	ZE	NS	NM	NB
PS	PM	PS	ZE	NS	NM	NB	NB
PM	PS	ZE	NS	NM	NB	NB	NB
B	ZE	NS	NM	NB	NB	NB	NB

4. Neural Network Controller

It is well known that ANN needs to be trained. High convergence accuracy and high convergence rates are desirable for ANN's training. This becomes even more serious for dynamic modeling of an ANN. As in ANN's dynamic modeling, errors in the outputs from each sample's training will be fed back to the ANN as inputs for the next sample's training. As a result, errors in each training will not only affect the training result of the current sample's training but also affect the following sample's training. Accumulation and propagation of the errors will greatly degenerate the performance of the ANN. Training of an ANN is basically a process of finding the global minimum of a predefined objective function. The most popular training algorithm is the back–error propagation (BP) algorithm. Many efforts have been made to improve the convergence properties. Some even suggested using the second–order partial derivatives of the objective function. However, the computation needed would be considerably increased. Fortunately, an algorithm has been developed that can greatly improve the performance of the conventional BP algorithm by simply introducing two variables to modify the step sizes of a conventional BP [8]. This improved BP algorithm offers high convergence rate and accuracy.

There are two learning models for an ANN as a neuro–controller, off–line training and on–line training. In the off–line training method, the learning process involves the minimization of the overall error between the desired ANN output and the actual ANN output. After the learning process is completed, the connection weights between the neurons are fixed and the ANN is used as a controller. This method required some means of determining the desired neuro–controller output, such as a reference controller. Also, tremendous amounts of training data covering the entire possible range of system operation must be used in training. Conversely, by using the alternative on-line approach, the NN learns during feed-forward control. This approach does not require a reference controller or large amounts of training data. The BP training algorithm is an iterative gradient algorithm. It is normally designed to minimize the mean square error between the actual output of an ANN and the desired output. It uses a recursive algorithm starting at the output units and working back through the hidden layer to adjust the neural weights according to the following equations:

$$S_{pj} = \sum w_{ji} O_{pi} + \theta_j \tag{5}$$

$$Opj = F(S_{pj}) \tag{6}$$

where: S_{pj} : input of neuron j for pattern p,

O_{pj} : output of neuron j for pattern p

θ_j : neuron bias, w_{ji} : weight from unit i to unit j,

$F(S_{pj})$: activation function

$$w_{ji}(t+1) = w_{ji}(t) + \Delta\ w_{ji}(t) \tag{7}$$

$$\Delta w_{ji} = \xi\ \delta_{pi} O_{pj} \tag{8}$$

$$\delta_{pj} = -\partial E_p / \partial O_{pj}\ \ F'(S_{pj}) \tag{9}$$

where $F'(S_{pj})$: differentiation of $F(S_{pj})$

E_p : error function , ξ : learning rate

δ_{pj} : error term for unit j

The error function normally used in the standard BP algorithm is :

$$E_p = 0.5\,(t_{pj} - O_{pj})^2 \tag{10}$$

where t_{pj} is the target output of the neuron j in the ANN. When neuron j is in an output layer,

$$-\partial E_p / jO_{pj} = (t_{pj} - O_{pj}) \tag{11}$$

and when j is in a hidden layer

$$-\partial E_p / \partial O_{pj} = \sum \delta_{pk} w_{kj} \tag{12}$$

Fig. 2 represents the multi-layer neural network controller for the PM dc motor. The inputs of the forward neural network are the reference speed, the actual speed, the previous values of the actual speed, the neural network output and the previous value of the neural network output.

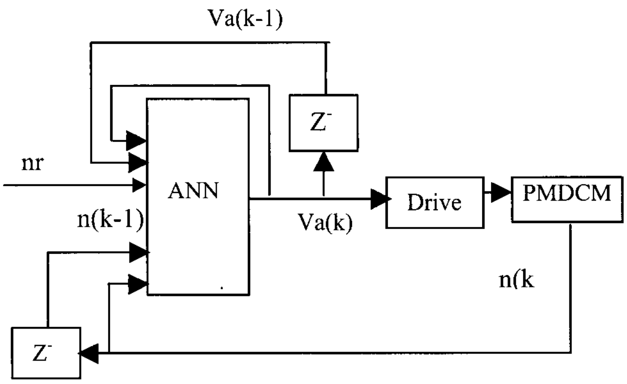


Fig. 2 Block diagram of Neural Network controller

5. Single-Neuron Neuro-Controller

This section presents the design of the proposed single-neuron neuro-controller(SNNC) as a speed controller for the PM dc motor. The SNNC consists of only one weight and one neuron with a linear hard limit activation function as shown in Fig.3.

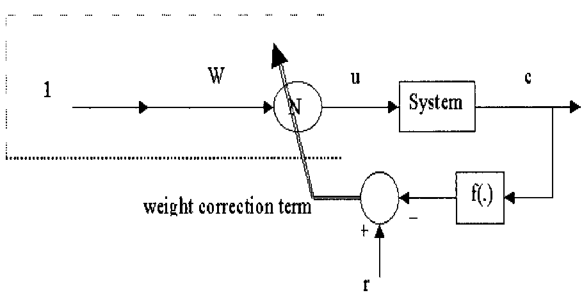


Fig. 3 Single neuron configuration

The SNNC output can be derived as:

$$u(t) = W(t) \tag{13}$$

$$W(t) = W(t-1) + \eta * WCT \tag{14}$$

where "WCT" is the neuron Weight Correction Term.

η : Learning rate

W (t) : neuron weight

u(t) : neuron output

The back propagation training algorithm is used to train the SNNC. The weight correction term is given by [9]:

$$WCT = \left[(r(t) - c(t)) - k \frac{d(c(t))}{dt} \right] \tag{15}$$

Where:

c(t): the controlled variable

r(t): the controller input (reference input)

Based on the back propagation algorithm, the weight change of the SNNC is given by:

$$\Delta W = \eta * WCT \tag{16}$$

Based on equations (15) and (16), the SNNC weight update depends on two parameters named k and η. The two parameters are selected by trial and error based on the author experiences. In future work, it is proposed that the two parameters be obtained analytically. The SNNC speed controller has ω_r^* (reference speed) and ω_r (a ctual speed) as the controller input and the controlled variable respectively. The WCT of the SNNC is given by the following equation:

$$WCT = \omega_r^* - \omega_r - k_\omega \frac{d}{dt} \omega_r \tag{17}$$

The controller output is the armature voltage.

The selected SNNC controller parameters are as follows: $\eta = .02$ and $k = .02$

6. Simulation Results and Discussions

Several tests were performed in this study to evaluate the performances of both the FLC, ANN controller and simple neuro controller based PM dc motor drives. The speed and current responses under various operating conditions, such as changes in reference speed and changes in load, were observed. Some of the sample results are documented in the following section. Fig.4 represents the block diagram of the general control system.

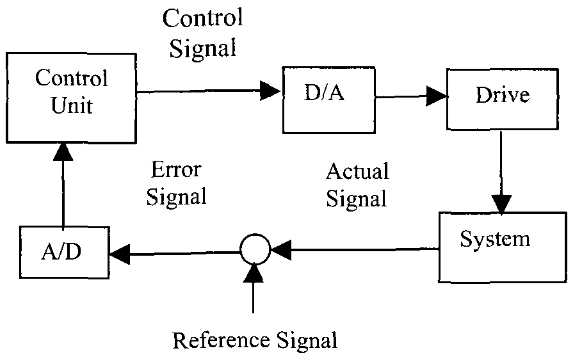


Fig. 4 Block diagram of general controller

Simulation was performed to obtain the speed responses for varying reference speeds with a constant load (full load 100%). Fig. 5 shows the speed and corresponding armature current responses using the fuzzy logic controller. Fig. 6 presents the responses of the ANN-controller under the same operating conditions. Fig. 7 shows the responses of the simple neuro controller under the same operating conditions. It is observed from Figs. 5, 6 and 7 that the performances of the simple neuro-based PM dc motor drive system are much better than those of the fuzzy or the ANN controller–based systems. The FLC and ANN controller–based drive system have the problem of over-shooting around 50 rad/sec. It is reported in [5] that the response of the ANN controller can be improved by using an adaptive learning rate.

The speed response of the simple neuro–based system is more robust against the step change in reference speed. The current response to the step change of speed is shown in Fig. 5, Fig. 6 and Fig. 7. It is observed from Fig. 5, Fig. 6 and Fig. 7 that the current control performances of the simple neuro–based PM dc motor drive system are much better than those of the fuzzy or the Neural Network controller based systems. The FLC and NN controller–based drive systems have the problem of oscillations represented by positive and negative current which causes damage to the motor and increases the settling time when there is a change in reference speed.

Fig. 8 shows the speed and current responses of the fuzzy logic based system for a step change of load at constant speed(300 rad/sec). The motor was running at full load and after some time, a step down change of load occurs (from 100% to 25%). Fig. 9 shows responses of the

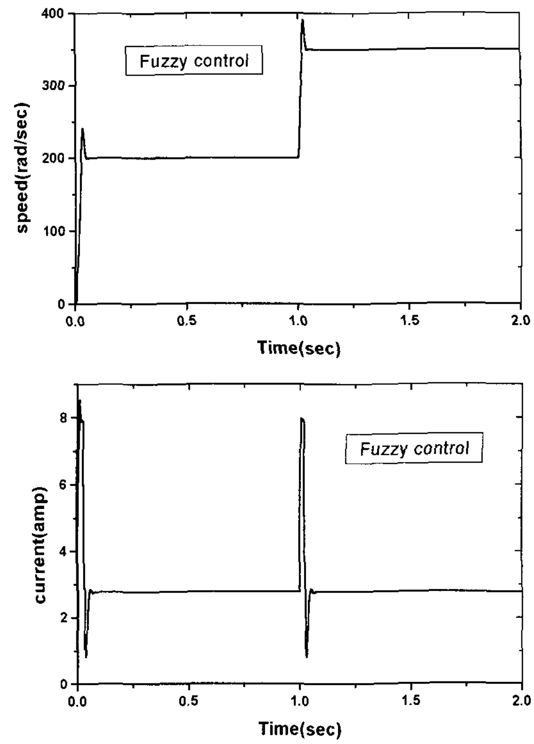


Fig. 5 Fuzzy logic controller response to the change in reference speed

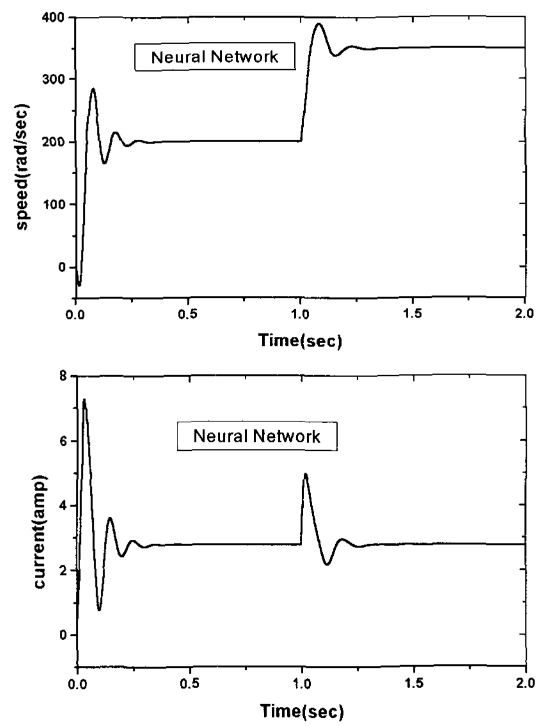


Fig. 6 ANN-controller response to the change in reference speed

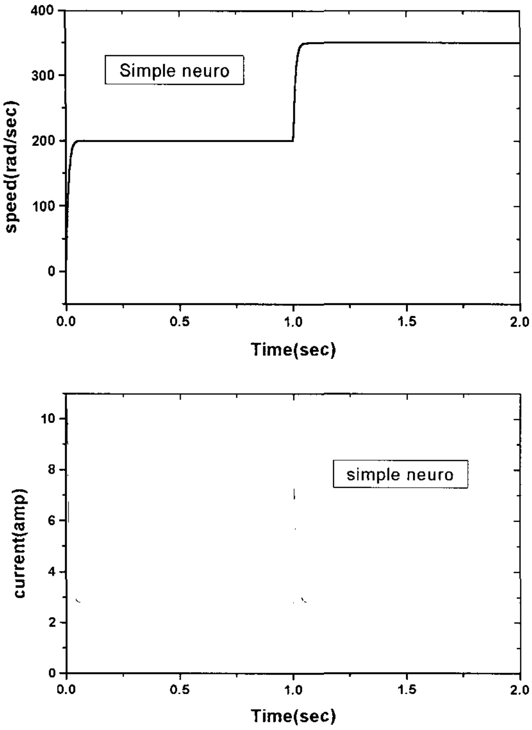


Fig. 7 Simple neuro controller response to the change in reference speed

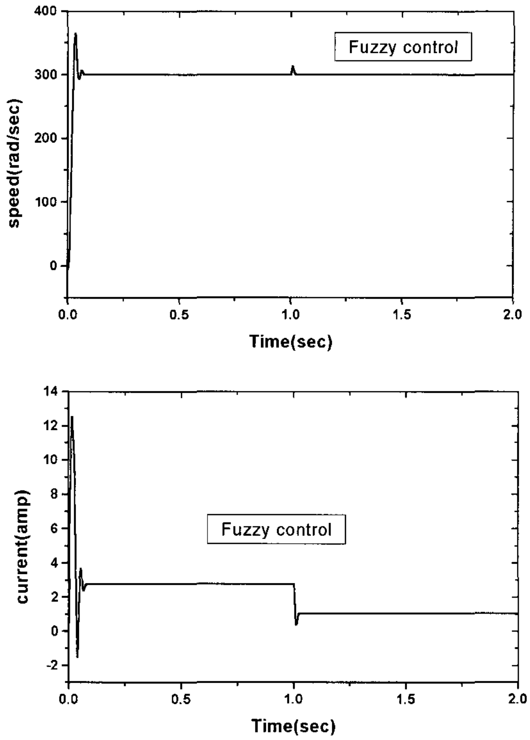


Fig. 8 Fuzzy logic controller response to the step down change

ANN ccontroller–based system under this operating condition. The ANN adjusts its weights and biases to this changing condition of sudden load impact and provides appropriate control voltage, so that the drive system responds according to the reference speed. Fig. 10 shows responses of the simple neuro–based system under this operating condition. From Fig.8, 9 and 10 it can be shown that the performance of the simple neuro-based system is much better than that of the fuzzy or Neural Network systems without an adaptive learning rate.

The motor speed and current responses at the step down change of load torque are shown in figs. 8, 9 and 10. These figures show the simple neuro controller is much better because the transient time is very small at the step down change of load torque. Also table 2 summarizes the simulation results.

Figs. 11&12 show comparisons between fuzzy logic control and simple neuro-control. Because the neural network needs the adaptive learning rate with the controller and the simple neuro controller can be considered to be a modification of a neural network, we

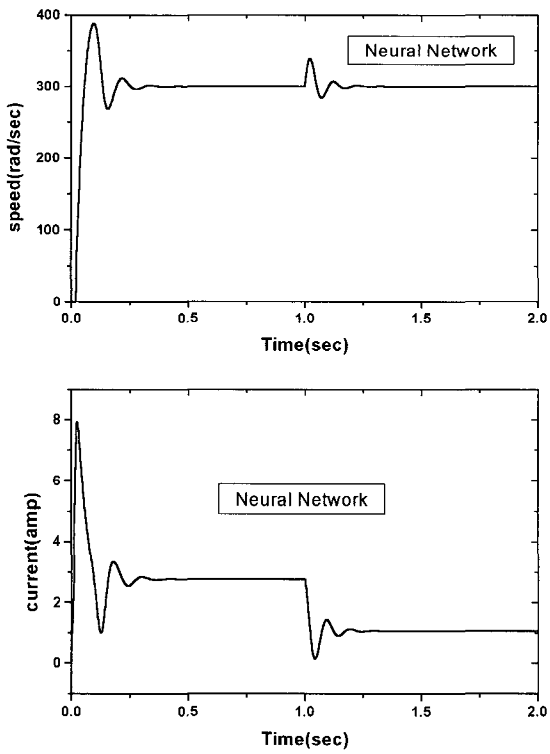


Fig.9 ANN-controller response to the step down change

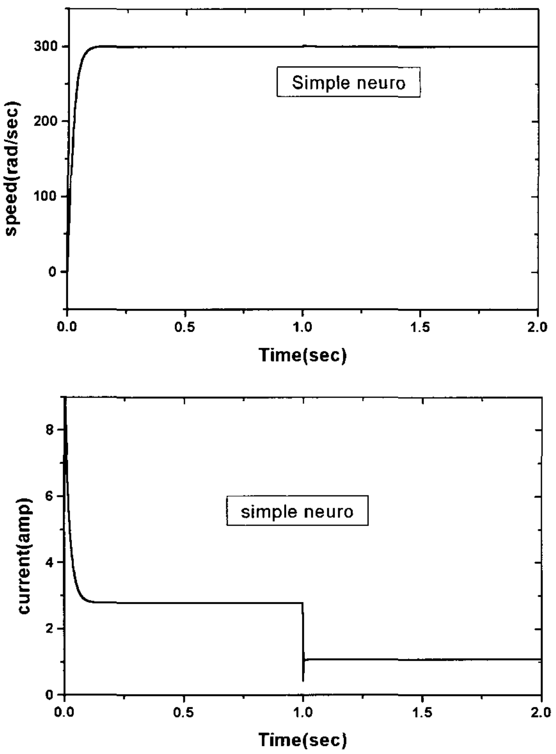
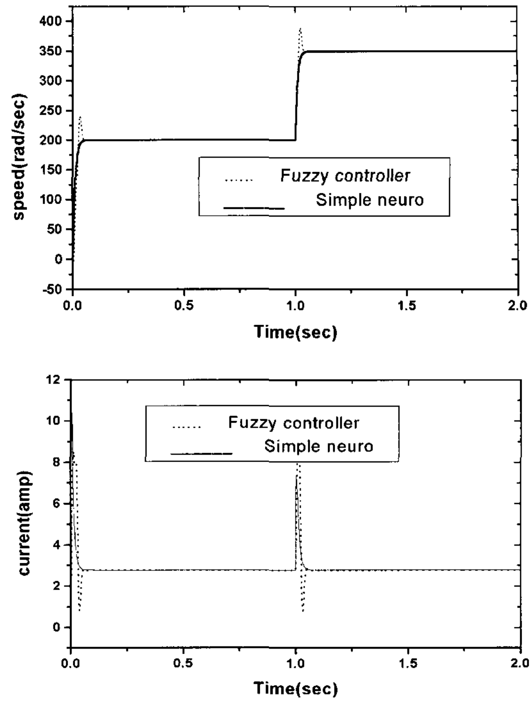


Fig. 10 Simple neuro controller response to the step down change

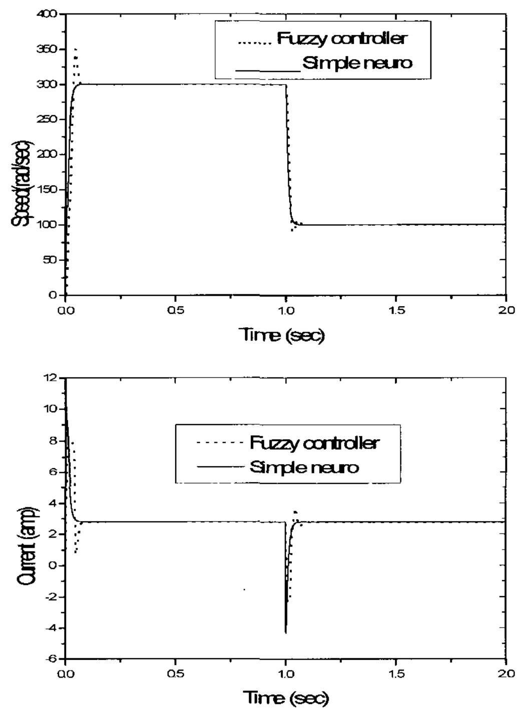
Table 2 summary results

Item	Neural Network	Fuzzy Controller	Simple Neuro
Over shooting	high	lower	Lowest
Transient current	lowest	lower	high
Settling time	long	shorter	shortest
Oscillation	high	lower	lowest

compare only the results between the fuzzy logic control and simple neuro control. Fig. 11 represents the speed and current responses at a step up(a) change in reference speed (from 200 rad/sec to 350 rad/sec) and a step down(b) change in reference speed (from 300 rad/sec to 100 rad/sec) at 100% load. Fig. 12 represents the speed and current responses to a step up(a) change of load torque (from 0% to 50%) and a step-down(b) change of load torque (from 100% to 25%) when the reference speed is 300 rad/sec.

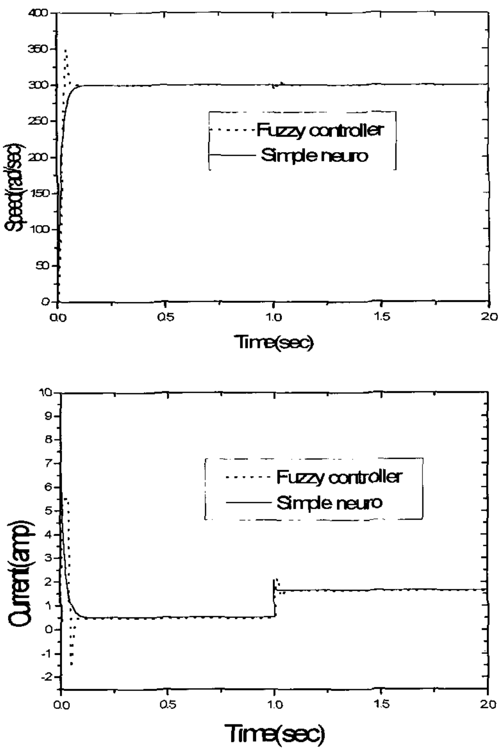


(a) step-up in reference speed from 200 rad/sec to 350 rad/sec

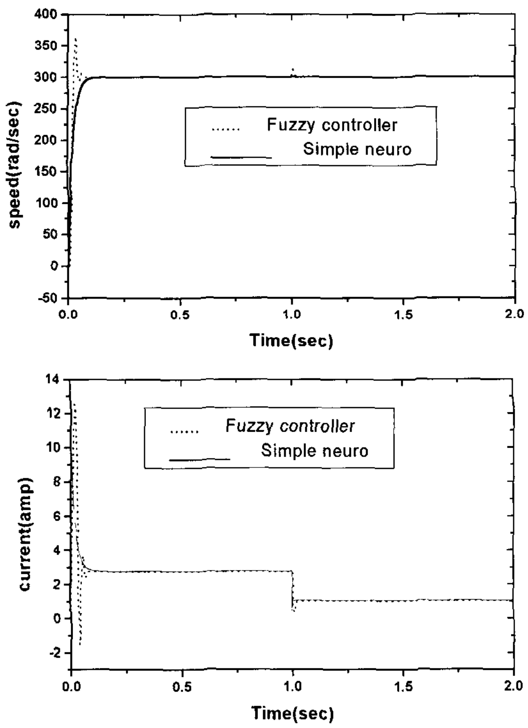


(b) step-down in reference speed from 300 rad/sec to 100 rad/sec

Fig. 11 Comparison of responses to a step-up & step-down changes in reference speed at full load



(a) step-up of load torque from 0% to 50%



(b) step-down of load torque from 100% to 25%

Fig. 12 Comparison of responses to a step-up & step-down load torque at 300 rad/sec

7. Conclusions

This paper presents an application of simple neuro control, Neural Network control and fuzzy logic control to a permanent magnet dc motor. The performance characteristics of all methods were compared. The comparative results indicate that the performance of the Simple Neuro control is clearly superior in the case of speed change (up or down) and load disturbances. When using fuzzy logic control to improve the system stability, one needs to decrease the overshoot ratio and settling time by adjusting either the rules, the input and output scaling factors, or some other parameters of the fuzzy controller. Also, the use of a Neural Network must incorporate the adaptive learning rate which changes according to the operating point in the proposed system, thus reducing the possibilities of overshooting during the transient conditions.

The proposed simple neuro-based speed control system of the PM dc motor is found to be robust, efficient, and easy to implement.

Appendix

Motor data:
Ra = 2.8 Ω,
B=.002 N.m/K r/min,
Kt=.0438 N.m/A,
La=1.17 mH,
J=.00002288 kg.m^2
Tf=.0212 N.m,
Ke= .0439 V.s/rad

References

[1] P.C. Sen "Thyristor Dc drives", New York wiley 1981,(Book).

[2] T. Fukuda, T. Shibata, "Theory and applications of neural Networks for industrial control system", IEEE Trans. On Ind. Electron., Vol.39, pp.472-489, Dec, 1992.

[3] S.Wee, M.El Sharkawi, "Identification and control of a dc motor using back-propagation neural networks", IEEE Trans. Energy. Conv. Vol.6, No.4, pp.663-669, Dec. 1991.

[4] M.A. Rahman, M.A. Hoque, "On-line Self-tuning ANN-Based Speed Control of a PMdc motor", IEEE Trans. on mechatronics, Vol.2, No.3, pp.169-178, Sept. 1997,.

[5] A.M.Zaki, H.M. Mashaly, A. Abdel-Sattar, S. Wahsh and S.I Amer, "Implementation of adaptive Neural Network Real-time speed controller for dc-motor", Electronic Conference, in proc. EPE'99 – Lausanne.

[6] S.S.Gurleyuk, S.Akpinar "Fuzzy control of a permanent magnet dc Motor and its lead estimation without speed sensor", in proc. MEPCON'97, Alex, Egypt, jun 4-6, 1997, pp. 142-145.

[7] Gi-Hyun, J.H. Park, Dong-Hee lee, Donf-Wan Kim, Hyun-Ju Park, J.I.Woo, "Implementation of FIC for speed control of a dc sorvomo using Real – Type Tabu search", in proc. ISIE 2001, pp.932-936, Pusan, Korea.

[8] L.Guan, S.Cheng and R.Zhou, "Artificial neural network power system stabiliser trained with an improved Bp algorithm", IEE proc-Gener. Transm. Distrib., Vol.143, No.2, pp.135-141, March 1996.

[9] M.M. Salem, A.M. Zaki, O.A. Mahgoub, E. Abu El-Zahab, and O.P.Malik, "Simple neuro-controller with a modified error function for synchronous generator", Elect. Power and Energy systems 25, pp.759-771, 2003.

S.I.Amer was born in Cairo, Egypt . She received the B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from Ain Shams University, Cairo, Egypt. Since 1998, she has been working as a researcher at the Power Electronics & Energy Conversion Department at the Electronics Research Institute in Cairo, Egypt. Her research interests include artificial intelligence and programmable logic controllers.

M. M. Salem received the B.Sc. degree in electrical engineering in 1988 from Minofia University in Egypt. He received the M.Sc. and the Ph.D degrees in electrical engineering from Cairo University in Egypt in 1994 and 2000, respectively. He is currently working as a researcher at the Electronics Research Institute in Cairo, Egypt. His research interests include artificial intelligence, neural networks, control systems, electrical drives, and digital controller hardware.