Artificial Intelligent Speed Control Strategies for Permanent Magnet DC Motor Drives

F. M. El-Khouly*, A. S. Abdel-Ghaffar, A.A. Mohammed**, and A.M. Sharaf*****
*, ** Department of Elec. Eng., Menoufia University, Shebin El-Kom, EGYPT.
*** Department of Elec. Engineering, University of New Brunswick,
Fredericton, CANADA, E3B 5A3, E.Mail sharaf@unb.ca
• currently with the University of New Brunswick, under Academic channel programme.

Abstract-The paper presents a prototype laboratory implementation and control system validation for two artificial intelligent (AI) based speed control strategies to be used with the permanent Magnet (PM) DC motor drives. These two AI strategies include a fuzzy logic based controller (FLC) and an on-line tunable artificial neural network (ANN) based controller. The use of these two AI based speed controllers is motivated by motor-drive system parameter uncertainties and the unknown nonlinear mechanical load characteristics over the extended range of operating conditions. The experimental results validate the good dynamic speed tracking performance of the two proposed speed controllers. Since the controlled motor-drive system precise parameter values are not required in the control system implementation, the controlled motor-drive system is robust and insensitive to the system parameters, load excursions, and operating condition changes

I. INTRODUCTION

High performance permanent magnet(PM) DC motor drives are used in multitude of industrial and process applications. In such applications, it is required that the motor speed closely follows a specified reference speed trajectory regardless of any load excursions, parameter variations and any model uncertainties. In case of DC motor drives the uncertainties come from the load side where the load usually has unknown nonlinear mechanical characteristics. This poses a challenge for classical PI controller with fixed parameters designs.

Several control techniques are evolving such as the sliding mode[1], variable structure[2], and model reference adaptive controls[3]. The sliding mode and variable structure control systems require a valid model and/or dynamics of the plant being controlled. Thus they are not robust enough to handle the structured model uncertainty and sensitive to large parameter variation and noise. Although adaptive controllers are effective in compensating for the influence of these structured uncertainties, it is not clear that the adaptive means can overcome any model unstructured uncertainties. In addition, conventional adaptive control schemes require the full information about the plant structure and may not guarantee the stability of the system in the presence of unmodeled dynamics [4]. Moreover, most of these adaptive control algorithms are rather complicated and, thus, need excessive computation effort for real-time implementation.

Facing this problem, investigators realized that incorporating artificial intelligence technologies into the automatic control systems would be more beneficial. The following three approaches can be classified as falling into that now become known as "intelligent control" :

-Knowledge-based Systems (KBSs), essentially based on expert systems (ESs). These use a symbolic reasoning approach, as commonly used in advisory systems, where the knowledge of experts in a small field of expertise is made available to the user.

-Artificial Neural Networks (ANNs), which are learning systems based on subsymbolic reasoning.

-Fuzzy Logic Control (FLC), which is very well suitable to handling heuristic knowledge in controlling a system.

This paper investigates ANN controller and FLC applications in speed control of PM DC motor drives.

II. SYSTEM MODEL

The permanent magnet DC motor is fed by type-B MOSFET chopper as shown in Fig. 1. The motor load comprises a fan load, as an example of the nonlinear load, and an additional DC generator feeding a resistive load to insure possible load excursion capability, (motor and load parameters values are given in appendix). A current feedback signal is used to ensure limited motor inrush current during starting and sudden load increasing conditions. The system, shown in figure 1, is modeled by the following equations:

$$V_a (t) = R_a \; i_a (t) + L_a \, [di_a(t) /dt] + K_V \, \omega (t) \qquad (1)$$

$$J \, [d\omega (t)/dt] + F \, \omega (t) = K_T \; ia (t) - T_L(t) \qquad (2)$$

The total load torque is given by:

$$T_L (t) = T_{LG} (t) + T_{LF} (t) \qquad (3)$$

with;

$$T_{LG} (t) = K^2_{eg} \; \omega (t) \, / \, (R_{ag} + R_L) \qquad (4)$$

and;

$$T_{LF} (t) = K_{L1} + K_{L2} \; \omega (t) + K_{L3} \; \omega^2 (t) \qquad (5)$$

where;

V_a,i_a -motor terminal voltage and armature current, respectively.
ω -motor speed.
R_a,L_a -motor armature resistance and reactance, respectively.
K_V,K_T -motor back emf and torque constants, respectively.
J,F -system moment of inertia and viscous friction, respectively.

0-7803-1993-1/94 $4.00 © 1994 IEEE

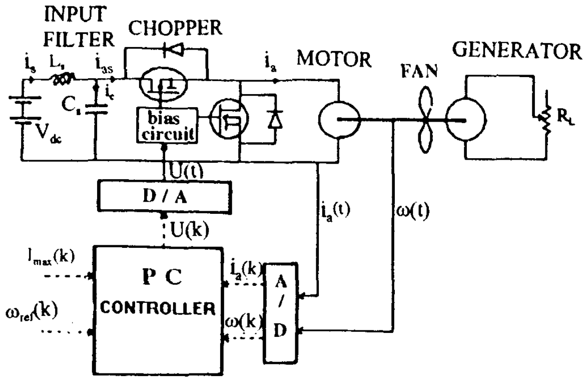


Fig. 1. System configuration.

K_{eg}, R_{ag} -DC generator back emf constant and armature resistance, respectively.
R_L -DC generator load resistance.
$K_{L1\ 2,3}$ -fan load torque constants.
T_{LG},T_{LF} -generator and fan load torques, respectively.

The discrete-time equivalent model is derived by first combining equations (1) to (5) and then replacing all continuous differentials with finite differences. The resulting state space equation can be approximated as a nonlinear function $f(.)$ as following:

$$\omega(\kappa) = f(\omega(\kappa\text{-}1), \omega(\kappa\text{-}2), V_a(\kappa)) \qquad (6)$$

Some parameters of this discrete function are unknown or varying, and so the need for intelligent control arise.

III. ANN CONTROLLER

Several schemes have been proposed to design a controller using ANN's. One of them is training an ANN to learn the system's inverse, and then the desired system output is achieved using the control input produced by the system's inverse[5]. If the system is a non-minimum phase system, then the resulting design is not internally stable and the inverse control scheme is not acceptable. Also, the system is not robust enough under the controlled plant parametric variations and any load excursion. Sharkawi[6] used the reference model with the ANN inverse model controller. Other examples of ANN controllers are references[7,8], which used reference models to train the ANN. Narendra[9] proposed a scheme of an indirect adaptive control. This paper proposes a simple training algorithm based on the back-propagation technique. The proposed ANN controller is trained on-line by using the actual system's output error directly, with little a priori knowledge of the controlled plant.

The object is to design an on-line tunable ANN based controller which will replace a conventional speed controller. In other words, the ANN controller is cascaded with the controlled plant as shown in Fig. 2. The inputs to the ANN are the reference speed at present time instant κ ($\omega_{ref}(\kappa)$), and the actual motor speeds at instants κ-1 and κ-2 (namely ω(κ-1), ω(κ-2), respectively). There are only one hidden layer with five neurons and one neuron in the output layer.

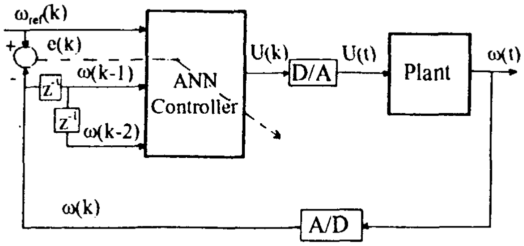


Fig. 2. System block diagram with ANN controller.

The neurons in hidden layer have a tansigmoid activation functions while it is a logsigmoid in the output neuron layer.

The ANN controller works as an inverse model and it is necessary to prepare its initial value of the weights, which is acquired by prior off-line training in order to avoid instability [10]. The on-line training insures robust controller against system parameter changes and the nonlinear load characteristics over a wide range of operating points.

Computation of the ANN output are summarized in the following steps:

Step 1- Compute the output of the jth neuron in the hidden layer ($O_j(\kappa)$) as:

$$Oj(\kappa) = \{2.0 /[\{1.0+\exp(-2\, h_j(\kappa) - 2\, b_j(\kappa)] \} - 1.0 \qquad (7)$$

$$\text{with } h_j = \Sigma_{i=1}^{N} W_{ij}(\kappa)\, X_i(\kappa) \qquad (8)$$

where;
$W_{ij}(\kappa)$ -the weights between the ith input ($X_i(\kappa)$) and jth neuron in the hidden layer at instant κ.
$b_j(\kappa)$ -the bias of the jth neuron in the hidden layer at instant κ.
N -the number of the inputs to the ANN (N=3 in this case).

Step 2- Compute the output of the output layer ($O_{out}(\kappa)$) as:

$$O_{out}(\kappa) = 1.0 / \{1.0 + \exp(h`(\kappa) - b`(\kappa) \} \qquad (9)$$

$$\text{and, } h`(\kappa) = \Sigma_{i=1}^{N1} W`_j(\kappa)\, O_j(\kappa) \qquad (10)$$

where;
$W`_j(\kappa)$ -the weights between the jth neuron in the hidden layer and the output layer.
$b`(\kappa)$ -the bias of the output layer neuron at time instant κ.
N1 -the number of neuron in the hidden layer (N1=5 in this case).

Step 3- Compute the actual control signal to the controlled plant, U(κ). Since the logsigmoid function used in the neurons forces the ANN output to be within the range (0, 1), although the control input is limited by the range of actuator (U_{min}=0 , U_{max} =10). Therefor the actual control signal is computed by:

$$U(\kappa) = O_{out}(\kappa)\, .\, (U_{max} - U_{min}) \qquad (11)$$

A. On-Line Training Algorithm

The back-propagation algorithm is used to train the ANN by minimizing the cost function, in terms of the system output error, defined as:

$$E(\kappa) = (1/2)\ \{e(\kappa)\}^2 = (1/2)\ \{\omega ref\ (\kappa) - \omega\ (\kappa)\}^2 \quad (12)$$

Using the gradient algorithm, the weights from the hidden to the output layer are modified by:

$$W'_j(\kappa+1) = W'_j(\kappa) + \Delta W'_j(\kappa) \quad (13)$$

where, $\Delta W'_j(\kappa) \propto - \{\partial E(\kappa) / \partial W'_j(\kappa)\}$ (14)

$$(\partial E(\kappa) / \partial W'_j(\kappa)) = -\{e(\kappa)\}\{\partial \omega\ (\kappa) / \partial O_{out}(\kappa)\}\ \{\partial O_{out}(\kappa) / \partial h'(\kappa)\}\ \{\partial h'(\kappa) / \partial W'_j(\kappa)\} \quad (15)$$

Because; $\{\partial O_{out}(\kappa) / \partial h'(\kappa)\} = h'(\kappa)\{1.0 - h'(\kappa)\}$ (16)

and, $\partial h'(\kappa) / \partial W'_j(\kappa) = O_j(\kappa)$ (17)

Eqn (12) becomes:

$$\{\partial E(\kappa) / \partial W'_j(\kappa)\} = - \{e(\kappa)\}\ \{\partial \omega\ (\kappa) / \partial O_{out}(\kappa)\}\ O_j(\kappa)\, h'(\kappa)\ \{1.0 - h'(\kappa)\} \quad (18)$$

Substituting Eqn (15) into Eqn (11) gives :

$$\Delta W'_j(\kappa) = \eta_1\ \delta\ O_j(\kappa) \quad (19)$$

where ,$\delta(\kappa) = \{e(\kappa)\}\ \{\partial \omega\ (\kappa) / \partial O_{out}(\kappa)\ )\ h(\kappa)\ \{1.0 - h'(\kappa)\}$ (20)

and $\eta_1 > 0.0$ is a gain factor. The only unknown in Eqn (16) is $\{\partial \omega(\kappa) / \partial O_{out}(\kappa)\}$. As η_1 is an adjustable scaling factor, then the absolute value of $\{\partial \omega(\kappa) / \partial O_{out}(\kappa)\}$ is not important and another gain factor η_2 can be used as:

$$\eta_2 = \eta_1\ \{\partial \omega(\kappa) / \partial O_{out}(\kappa)\} \quad (21)$$

In the gradient algorithm, the solution converges to minimum of the cost function if and only if the search is made along the negative direction of the gradient of cost function. So, it is necessary and sufficient condition for the convergence of the training algorithm is:

$$\text{sign}\ (\eta_2) = \text{sign}\ \{\eta_1\ (\partial \omega(\kappa) / \partial O_{out}(\kappa))\} \quad (22)$$

If the system output monotonically increases (decreases) as the control input to the controlled plant increases, then the sign of $\{\partial \omega(\kappa) / \partial O_{out}(\kappa)\}$ is defined by D =1.0 (D = -1.0),for the system under study D=1.0. Therefor the corresponding algorithm for updating the weights from hidden to output layer is:

$$W'_j(\kappa+1) = W'_j(\kappa) + \eta_2\ \delta(\kappa)\ O_j(\kappa) \quad (23)$$

where, $\delta(\kappa) = e(\kappa)\ D\ h'(\kappa)\ \{1.0 - h'(\kappa)\}$ (24)

Using the same procedure the weights from the ith input to the jth neuron in the hidden layer, $W_{ij}(\kappa)$, are updated as:

$$W_{ij}(\kappa+1) = W_{ij}(\kappa) + \eta_2\ \delta_j(\kappa)\ X_i(\kappa) \quad (25)$$

where, $\delta_j(\kappa) = \delta(\kappa)\ W'_j(\kappa)\ \{1.0 - h_j^2(\kappa)\}$ (26)

and the bias of the output layer is updated by:

$$b'(\kappa+1) = b'(\kappa) + \eta_2\ \delta(\kappa) \quad (27)$$

Also, the bias of the jth neuron in hidden layer is updated as:

$$b_j(\kappa+1) = b_j(\kappa) + \eta_2\ \delta_j(\kappa) \quad (28)$$

A current limiter is used to prevent the inrush current under starting and load changing as:

$$\begin{array}{lll} U(\kappa) = U(\kappa) & \text{if} & i_a < I_{max} \\ U(\kappa) = U(\kappa - 1) & \text{if} & i_a \geq I_{max} \end{array} \quad (29)$$

IV. FUZZY LOGIC BASED CONTROLLER

Fuzzy logic controllers have emerged as one of the most active and fruitful areas for research in the application of fuzzy set theory. Different approaches and applications for the FLC have been developed [11-15]. The fuzzy algorithm emulates the operator behaviour more than simulating the exact mathematical model of the plant itself. This takes the form of rules namely production rules which are deduced according to heuristics and human expertise. In this paper, a real-time FLC implementation using the fuzzy set theory is presented. The following steps describe the FLC algorithm:

1- Calculate the normalized speed error at the κth time instant.

$$e\ (\kappa) = [\ \omega_{ref}(\kappa) - \omega\ (\kappa)\ ]\ /\ \omega_{base} \quad (30)$$

where ω_{base} is the base speed.

2- Calculate the normalized speed error change.

$$\Delta e\ (\kappa) = G_1\ \{e\ (\kappa) - e\ (\kappa - 1)\} \quad (31)$$

where;

e(κ-1) -speed error at the previous instant.

G_1 -gain to normalize the error change.

3- A membership function is used to assign the membership grades to each linguistic class. These classes are namely positive big, positive small, zero, negative small, negative medium, and negative big {PB, PM, PS, ZE, NS, NM, NB}, respectively. For simplicity, a normalized nonlinear exponential membership function, shown in Fig. 2, is used and given by:

$$\mu\ (x) = \exp\{-(\ x - a\ )^2 / 2\ b^2\} \quad (32)$$

where a, b are defined in table I, and x is either the fuzzy input variables (e or Δe) or the controller output change (ΔU).

4- Using the law of intersection of two fuzzy set, the membership

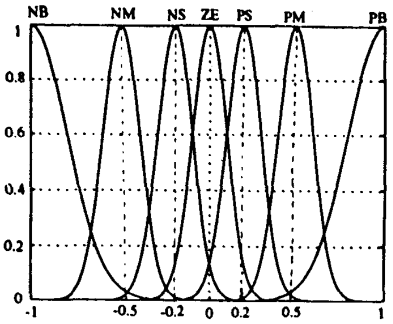


Fig. 3. Grads of membership functions.

TABLE I
FUNCTION DEFINITION OF PRIMARY FUZZY SETS.

primary fuzzy set	a	b
NB	- 1.0	0.4
NM	- 0.5	0.2
NS	- 0.2	0.2
ZE	0.0	0.2
PS	0.2	0.2
PM	0.5	0.2
PB	0.1	0.4

grades of the condition part are computed as:

$$\mu\ (i, j) = \min\ \{\ \mu_e\ (i)\ ,\ \mu_{\Delta e}\ (j)\ \} \qquad (33)$$

where i, j = 1, 2,......, 7 (number of labels)

5- The proposed fuzzy control rules matrix is shown in table II. As shown, there are 49 assignment rules on the basis on which the fuzzy controller's output is selected. For example, a typical rule is :

$R_{1,2}$ IF e is NM and Δe is PB THEN the controller output change (ΔU) is PS.

Knowing the membership grades for the condition parts and the fuzzy output matrix, the membership grades for the controller output are computed, by scaling the output membership function, as follows:

$$\mu_R\ (i, j) = \ \mu\ (i\ , j)\ .\ \mu_U\ (i\ , j) \qquad (34)$$

where;

μ_U (i, j) -membership function of the output linguistic label defined by the fuzzy output rules matrix.

i, j -elements of the rows and columns of the output linguistic control matrix, respectively.

6- The final grades of membership are determined using the composition rule of fuzzy set theory. For example, the controller output characterised by the linguistic label "PB" can be evaluated as follows:

$$\mu_R\ (PB) = \max\ \{\ \mu_R\ (1,7),\ \mu_R\ (2,6)),\ \mu_R\ (2,7),\ \mu_R\ (3,7) \qquad (35)$$

TABLE II
LINGUISTIC OUTPUT CONTROL RULES MATRIX

		Speed error (e)						
		NB	NM	NS	ZE	PS	PM	PB
Speed error change (Δe)	PB	ZE	PS	PS	PM	PM	PB	PB
	PM	NS	ZE	PS	PS	PM	PM	PB
	PS	NS	NS	ZE	PS	PS	PM	PM
	ZE	NM	NS	NS	ZE	PS	PS	PM
	NS	NM	NM	NS	NS	ZE	PS	PS
	NM	NB	NM	NM	NS	NS	ZE	PS
	NB	NB	NB	NM	NM	NS	NS	ZE

7- The change in the controller output is defined using the centre of area method, which yields:

$$\Delta U\ (\kappa) = \Sigma_i\ \{\ \mu\ (\Delta U_i\ )\ .\ \Delta U_i\ \}\ /\ \Sigma_i\ \ \mu\ (\Delta U_i\ ) \qquad (36)$$

with; i=1, 2, ..., n

where;

ΔU_i -sample value along the output universe of discourse.

$\mu(\Delta U_i)$ -the corresponding membership grade.

n -the number of samples along the universe of discourse.

8- Since the output controller universe of discourse is normalized while the actual output control signal to the chopper has the analog range from 0.0 to 5.0 volts, then an output scaling factor is used to obtain the actual output control voltage. This scaling factor is given by:

$$K_{out} = 5.0 \qquad (37)$$

9- Motor current is fedback in order to avoid any overloading and to prevent starting inrush current. So, a current limiter factor is added to the controller output. This factor is defined as:

$$K_I = 1.0 \qquad \text{if } i_a < I_{max}$$

$$K_I = \text{sign}\{(I_{max} - i_a)/\ I_{max}\} \qquad \text{if } i_a \geq I_{max} \qquad (38)$$

where I_{max} is the maximum allowable motor current (I_{max}=2 I_{rated}). The 'sign' is defined in such a way to keep a negative action of the controller under condition $i_a \geq I_{max}$.

10-The actual output control voltage is calculated as:

$$U(\kappa) = U(\kappa\text{-}1)\ +\ \Delta U_{act.}(\kappa) \qquad (39)$$

with;

$$\Delta U_{act.}(\kappa) = \Delta U\ (\kappa)\ \ K_{out}\ \ K_I\ \ G_O \qquad (40)$$

where;

U(κ),U(κ-1) -the control voltage at instants (κ), (κ-1), respectively.

$\Delta U_{act.}(\kappa)$ - the actual change of the controller output at instant κ.

ΔU(κ),K_{out},K_I -values defined by equations (36), (37), and (38), respectively.

G_O -an output gain

System output response is highly affected by the two gains, G_I

System output response is highly affected by the two gains, G_I and G_O. These gains are adjusted by trail and error until the best response is obtained (values are G_I=2.0 and Go= 0.4).

V. EXPERIMENTAL RESULTS

The proposed controllers are implemented using a digital computer and data translation card DT2821. The initial values of the weights and biases are given in appendix and the learning factor (η_2) is adjusted by trail and error (its value is chosen equal 0.01). To validate the controller robustness and effectiveness, the system performance is tested laboratory over a wide range of operating conditions.

Experimental results under a wide range of speed reference step changes are shown in figs.4 (a and b) with ANN controller, and in figs. 5 (a and b) with the FLC.

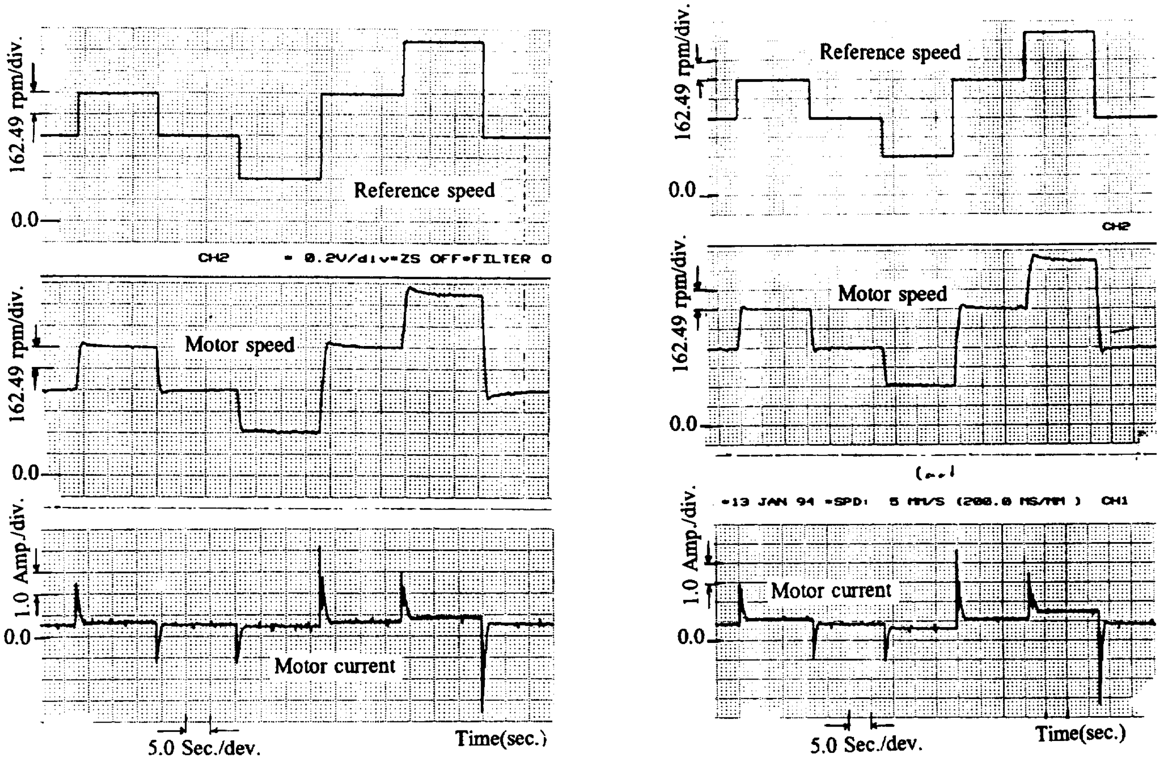
Figs. 6 and 7 depict the experimental test results for speed and current dynamic responses during starting and load excursions for both ANN and FLC proposed controllers, respectively.

The results show a good speed reference tracking for both proposed FLC and ANN controllers.

VI. CONCLUSION

The paper presents two artificial intelligent (AI) speed reference tracking control systems, using Fuzzy Logic based controller and an Artificial Neural Network controller. Since the exact system parameters are not required in the implementation of the proposed controllers, and due to the inherent future of the controllers adaptive capability, the performance of the controlled motor-drive system is robust, stable, and insensitive to both drive system parameters and operating condition changes. The robustness of the controller is validated under load excursions for different speed reference trajectories.

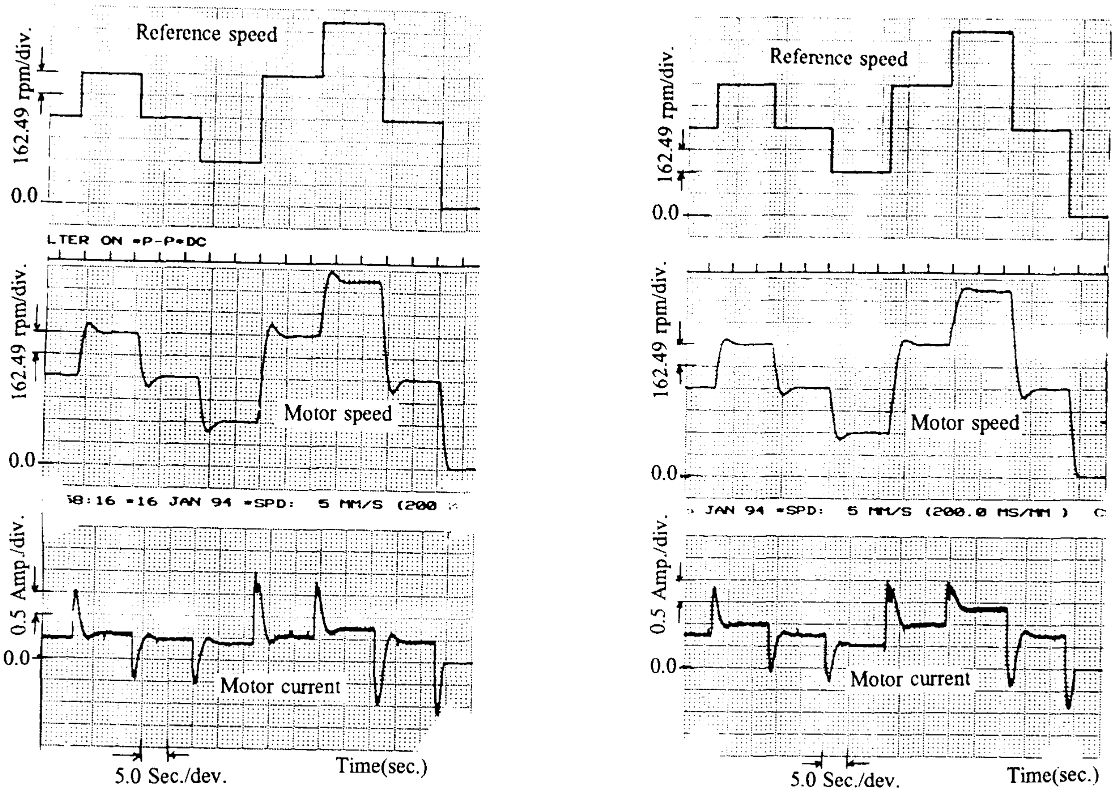
A new direct on-line tunable adaptive controller using ANN has been developed. The training algorithm is derived based on back-propagation, enabling the ANN to be trained with the actual controlled plant output error rather than the ANN output error.



(4 - a) At no load (R_L= 0.0 Ω).

(4 - b) At load (R_L= 40.0 Ω).

Fig. 4. Performance under step change in reference speed with ANN controller.



(5 - a) At no load (R_L= 0.0 Ω).

(5 - b) At load (R_L= 40.0 Ω).

Fig. 5. Performance under step change in reference speed with FLC.

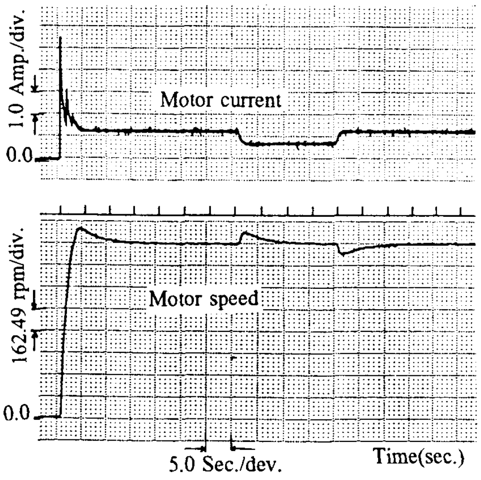


Fig. 6. Starting and load changing with ANN controller.

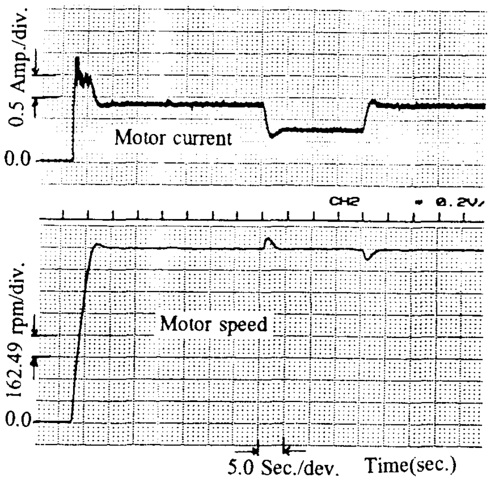


Fig. 7. Starting and load changing with FLC.

APPENDIX

A permanent magnet DC motor with ratings of 36 (W), 24 (V), and 1400 rpm is used in results. Following parameter values are associated with it,

System moment of inertia = 0.001525 Kg m^2,
Armature inductance = 0.00929 H,
Back emf constant = 0.1987465 Nm A^{-1},
Armature resistance = 4.0 Ω ,
System vicious fraction= 0.0008015 Nm/rad/s.

The values of both fan load torque constants and, the additional load, PM DC generator parameters are:

Fan load torque=$K_{L1} + K_{L2}\omega + K_{L3}\,\omega^2$
where, K_{L1} = 0.0486, K_{L2} = 55.47 E-5, K_{L3} = 19.799 E-5

Generator armature resistance = 4.0 Ω ,
Generator back emf constant = 0.1809 Nm A^{-1}.

The initial values of weights and biases for the ANN controller, obtained from off-line training of the inverse model, are:

W_{11}=-1.106	W_{21}=-0.462	W_{31}=0.1106
W_{12}=-1.167	W_{22}=-0.0008	W_{32}=0.3738
W_{13}=0.3583	W_{23}=0.652	W_{33}=-0.938
W_{14}=0.3883	W_{24}=-1.118	W_{34}=-0.3032
W_{15}=0.709	W_{25}=-0.6804	W_{35}=-0.65926
b_1=-0.2126	b_2=0.5602	b_3=0.1719
b_4=1.0369	b_5=0.80025	
W'_1=0.1097	W'_2=-0.921	W'_3=0.2817
W'_4=-0.34788	W'_5=0.26313	
b'=0.6768		

REFERENCES

[1] H. Hashimoto, K. Maruyama, and F. Harashima, "A Microprocessor Based robot manipulator Control with Sliding Mode," *IEEE Trans. Ind. Electron.*, Vol. IE-34, pp. 11-18, 1987.

[2] M. El-Sharkawi, and C. Huang, "Variable Structure Tracking of DC Motor for High Performance Applications," *IEEE Trans. on Energy Conversion*,Vol. 4, pp. 643-650, 1989.

[3] H. Naitoh, and S. Tadakuma, "Microprocessor Based Adjustable Speed DC Motor Drives Using Model Reference Adaptive Control," *IEEE Trans. Industry Applications*, Vol. IA-23, pp. 313-318, 1987.

[4] C.J. Harris, and S. A. Billings, "Self-Tuning and Adaptive control: Theory and Applications," *London: Peter Peregrinus Ltd.*, 1985.

[5] D. Psaltis, A. Sideris, and A.A. Yamamura, "A Multilayered Neural Network Controller." *IEEE Control System Mag.*, pp. 17-21, 1988.

[6] S. Weerasooriya, and M. El-Sharkawi, "Laboratory Implementation of a Neural Network Trajectory Controller for a DC Motor." *IEEE Trans. on Energy Conversion*, Vol. 8, No. 1, March 1993.

[7] A. Guez and J. Selinsky, "A neuromorphic controller with a Human Teacher." *Proc. of 1988 Int. Conf. on Neural Networks 2*, II595-II602, 1988.

[8] I. Bar-Kana, and A. Guez, "Neuromorphic Adaptive Control." *Proc. of 1988 IEEE Int. Conf. on Decision and Control 2*, 1739-1743.

[9] K.S. Narendra, and K. Parthasarathy, "Identification and control of Dynamical Systems using Neural networks." *IEEE Trans. on Neural Networks 1*, pp. 4- 27, 1990.

[10] T. Fukuda, and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Trans. on Industrial Electronics*, Vol. 39, No. 6, pp. 472-489, December 1992.

[11] E.H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *Int. Journal Man-Machine Stud.*, Vol 7, pp. 1-13, 1975.

[12] W.J.M. Kickert, and H.R. VAN nauta Lemke, "Application of a fuzzy controller in a warm water plant", *Automatica*, Vol. 23, pp. 301-308, 1976.

[13] P.J. King, and E.H. Mamdani, "The application of fuzzy control systems to industrial processes", *Automatica*, Vol. 13, pp. 235-242, 1977.

[14] K.H. Abdul-Rahman and S.M. Shaidehpour, "A fuzzy - based optimal reactive power control", *IEEE Trans. on Power Systems*, Vol. 8, No. 2, pp. 662-670, May 1991.

[15] M.A.M. Hassan, O.P. Malik,and G.S. Hope, "A fuzzy logic based stabilizer for a synchronous machine", *IEEE Trans. on Energy Conversion*, Vol. 6, No. 3, pp. 407-413, September 1991.