FUZZY NEURAL NETWORK CONTROLLER AS A REAL TIME CONTROLLER USING PSO

[1]Sıtkı Öztürk, [2]Cihan Karakuzu, *[3]Melih Kuncan and [4]Ahmet Erdil
[1]Faculty of Engineering, Department of Electronics and Telecommunications Engr. Kocaeli University, Turkey
[2] Faculty of Engineering, Department of Computer Engineering Bilecik Şeyh Edebali University, Turkey
[3, 4] Faculty of Engineering, Department of Mechatronics Engineering Kocaeli University, Turkey
Geliş Tarihi: 2016-11-09 Kabul Tarihi: 2016-12-14

Abstract

Direct current (DC) motors are commonly used to control position or speed in many applications. The speed of the DC motors is adjustable in a wide range with advantages such as easy control theorems and high performances. DC motors are used in industrial branches like transportation, electrical train, vehicle, crane, printer, drivers, paper industry in which adjustable speed and sensitive position handling are necessarily. In recent years, these applications are commonly used for household appliance in which low power and low cost are required with adjustable speed and sensitive position handling as well. In this study, permanent magnet direct current motor actuator is implemented by using fuzzy neural network structure. Particle Swarm Optimization (PSO) algorithm is used as training algorithm of fuzzy neural network controller. Learning and control in real time is executed in Matlab. Dynamic performance of the system is observed for constant and variable reference trajectory of speed.

Keywords: DC Motor Speed Control, ANFIS, Optimization, Particle Swarm Optimization (PSO)

1. Introduction

In this article, ANFIS (Adaptive Network Based Fuzzy Inference System) is used as fuzzy neural network which is equivalent to Sugeno fuzzy logic model. Such these network structures are known as “fuzzy inference system based applied artificial neural network” or “adaptive fuzzy-neural inference system” [1]. The learning and control block used is shown below, Figure 1. In Fig. 2 is shown the block that ANFIS network controller is trained and used.

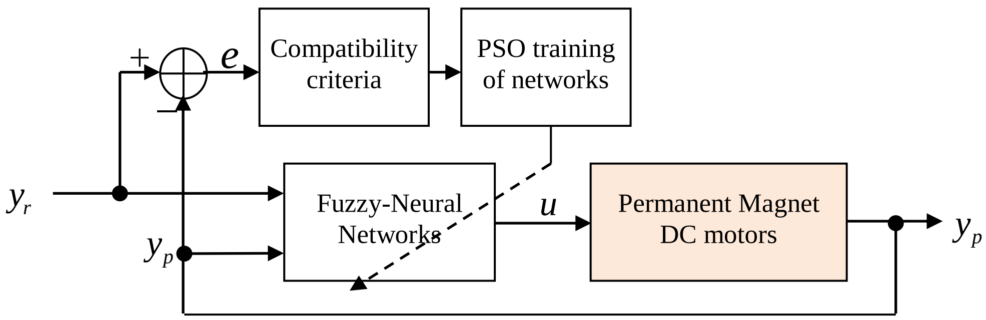


Figure 1. Teaching of Fuzzy Neural Network

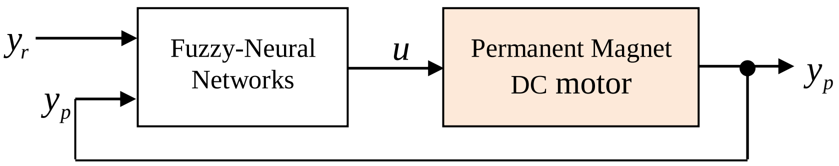


Figure 2. Control of the System by using Fuzzy Neural Network

2. Adaptive Network Based Fuzzy Inference System, ANFIS

*Corresponding author: Address: Faculty of Engineering, Department of Mechatronics Engineering Kocaeli University, 41380, Kocaeli TURKEY. E-mail: melih.kuncan@kocaeli.edu.tr, Phone: +902623033098

Doi: 10.21541/apjes.79471

ANFIS (Adaptive Network Based Fuzzy Inference System) is used as fuzzy neural network which is equivalent to Sugeno fuzzy logic model. First order Sugeno fuzzy model which has two input x, y and an output z are analyzed to figure out fuzzy inference system in ANFIS. Two rules which are commonly used in rule sets are shown below for first order Sugeno fuzzy model.

Rule 1: If $x\ is\ A_1$ and $y\ is\ B_1$ then $f_1 = p_1x + q_1y + r_1$
Rule 2: If $x\ is\ A_2$ and $y\ is\ B_2$ then $f_2 = p_2x + q_2y + r_2$

Basic inference principle of Sugeno fuzzy model is in Fig. 3. The equivalent model in ANFIS of Sugeno fuzzy model is shown below as well, Fig. 4.

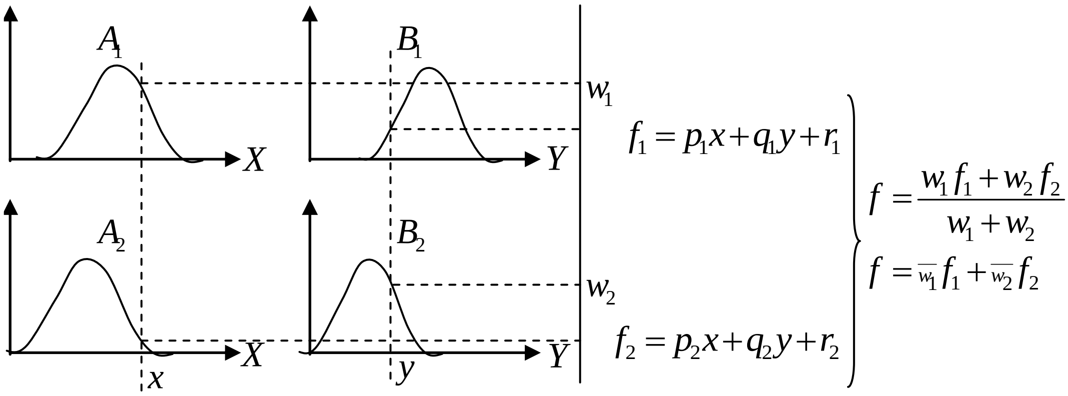


Figure 3. Sugeno fuzzy model with 2 inputs and 2 rules [1]

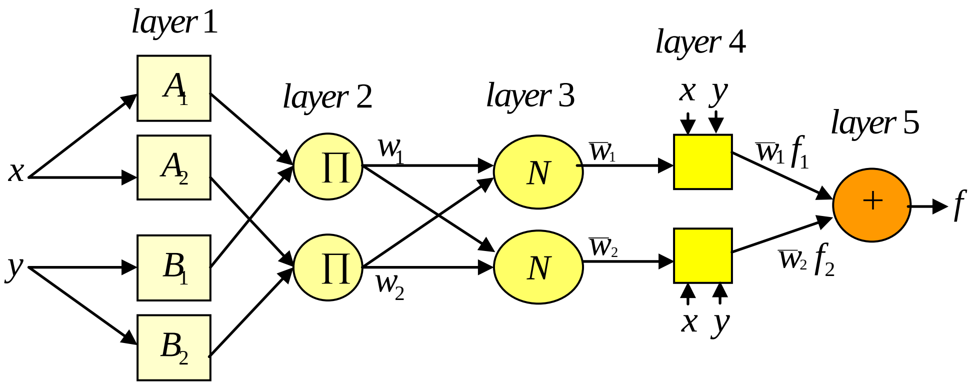


Figure 4.The equivalent model in ANFIS of Sugeno fuzzy model in Fig.3 [1, 2]

PSO parameters which are shown below and used to teach are necessary to control the speed of DC motor. In steady state conditions, system is run with sample as 0.3 sec. Training procedure is repeated 40 times with random ANFIS parameters for a default value which is continuously changing sinusoidal.

• Number of particle or swarm size = 60
• Moment of Inertia, linear decreasing values which are depends on iteration in the range of $w =$ [1,2 0,8].
• Coefficients of learning $c_1 = c_2 = 2$

In ANFIS network structure, there are 8 parameters for 4 membership functions which is Gaussian as input, 12 parameters for 4 rules and total 20 parameters. Every particle in set (swarm) has 20 parameters and is used so.

3. Particle Swarm Optimization Algorithm

Most of the physical problems have a nonlinear relation. Having the high degree of difficulty of the variables in the problem reduce the performance of the classical solution methods. In addition to the usage of fuzzy logic in nonlinear problems with such as genetic algorithm and artificial neural network in addition to the effective use of methods such as genetic algorithms and neural networks methods, many methods are used to solve these types of problems by inspiring from the system of nature revealed. One of these methods is heuristic search based on swarm [3].

Particle swarm optimization (PSO) is random search algorithm which based on population and is improved by inspiring bird flocks. It is used for the solution of nonlinear problems. PSO algorithm starts to work with an initial population and refreshes its individuals (particle to find final and most appropriate solution. Potential solutions called particles in the algorithm of PSO scan all workspace by using current optimal solution. The most important difference of PSO from classical one is that derivation is methods not required. This advantage of PSO provides less complex processing load which is necessary for solutions of many problems. The usage of PSO is quite simple because there is less number of parameters to set. PSO can be applied successfully in such areas function optimization, fuzzy system control and learning of artificial neural networks [4-8].

PSO tries to find final and most appropriate solution by repeating of each steps. According to the two best particles, positions of particles are refreshed in each step. The first is currently the best coordinates which are obtained until that time for each particle. This particle is called "P_{best}" and should be recorded. The second is the best coordinates obtained so in swarm and provide best solution. This is the best global value and called "G_{best}" [9]. For instance, it is shown below that the positions X and the speeds V of S particle which are the member of a space that has N parameter and S particle which means is N-dimensional space.

$$\boldsymbol{X} = \begin{bmatrix} x_{11}x_{12} & \ldots. & x_{1N} \\ x_{21}x_{22} & \ldots. & x_{2N} \\ \ldots \quad \ldots & \ldots. & \ldots \\ x_{S1}x_{S2} & \ldots. & x_{SN} \end{bmatrix}_{SxN} \tag{1}$$

$$\boldsymbol{V} = \begin{bmatrix} v_{11}v_{12} & \ldots. & v_{1N} \\ v_{21}v_{22} & \ldots. & v_{2N} \\ \ldots \quad \ldots & \ldots. & \ldots \\ v_{S1}v_{S2} & \ldots. & v_{SN} \end{bmatrix}_{SxN} \tag{2}$$

Local best (P_{best}) matrix is represented just like below.

$$\boldsymbol{P_{best}} = \begin{bmatrix} p_{11}p_{12} & \ldots. & p_{1N} \\ p_{21}p_{22} & \ldots. & p_{2N} \\ \ldots \quad \ldots & \ldots. & \ldots \\ p_{S1}p_{S2} & \ldots. & p_{SN} \end{bmatrix}_{SxN} \tag{3}$$

The series of particle position, particle speed, P_{best} and G_{best} of i which is index of particle are shown below respectively;

$$\boldsymbol{X}_i = [x_{i1}\, x_{i2} \;\; \ldots. \;\; x_{iN}] \tag{4}$$

$$\boldsymbol{V}_i = [v_{i1}\, v_{i2} \;\; \ldots. \;\; v_{iN}] \tag{5}$$

$$\boldsymbol{P_{best}}_i = [p_{i1}\, p_{i2} \;\; \ldots. \;\; p_{iN}] \tag{6}$$

$$\boldsymbol{G_{best}} = [p_1\, p_2 \;\; \ldots. \;\; p_N] \tag{7}$$

After the two best are determined, positions and speeds of particle are updated by the equation below [10].

$$v_{i,n}^{k+1} = w.v_{i,n}^{k} + c_1.rnd_1^k \left(P_{best}{}_{i,n}^{k} - x_{i,n}^{k}\right) + c_2.rnd_2^k\left(G_{best}{}^{k} - x_{i,n}^{k}\right) \tag{8}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \qquad i = 1,2,\ldots,S; \;\; n = 1,2,\ldots,N \tag{9}$$

In Eq. (8), w which is moment of inertia should be selected in certain range and decreases linear in the each step of iteration. c_1 and c_2 are learning coefficients and acceleration parameters which lead every particle to the P_{best} and G_{best} position i.e. c_1 represents parameter which regulates the movement cause of its own experience and c_2 represents parameter which regulates the movement cause of other particle's experience. These parameters are often selected in range from 0.2 to 2 and are positive as well [11]. rnd in equation is the numbers which are uniformly distributed in range [0,1]. k represents number of iteration and i represents the particle index. Thus, each particle in swarm uses not only its own experience but also every other particle's experience [12]. Linear decreasing of w can be calculated by Eq. 10 [13].

$$w = w_{max} - iter.\frac{w_{max}-w_{min}}{iter_{max}} \tag{10}$$

Changes of PSO in search space are shown in Fig. 5 as vector [14]. Flowchart of PSO is given in Fig. 6.

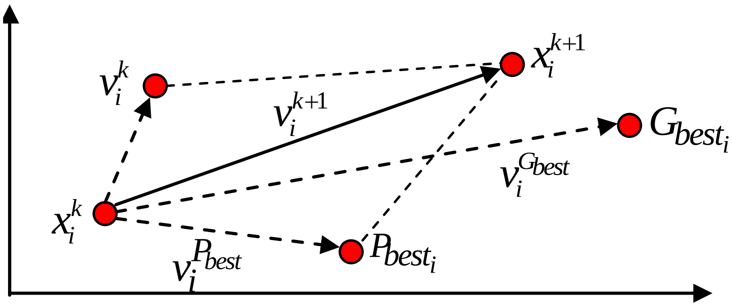


Figure 5. Representation of PSO parameters as vector x^k, current position; x^{k+1}, next position; v^k, current speed; v^{k+1}, next speed; $v^{P_{best}}$, speed based P_{best}, $v^{G_{best}}$, speed based G_{best}. P_{best_i}

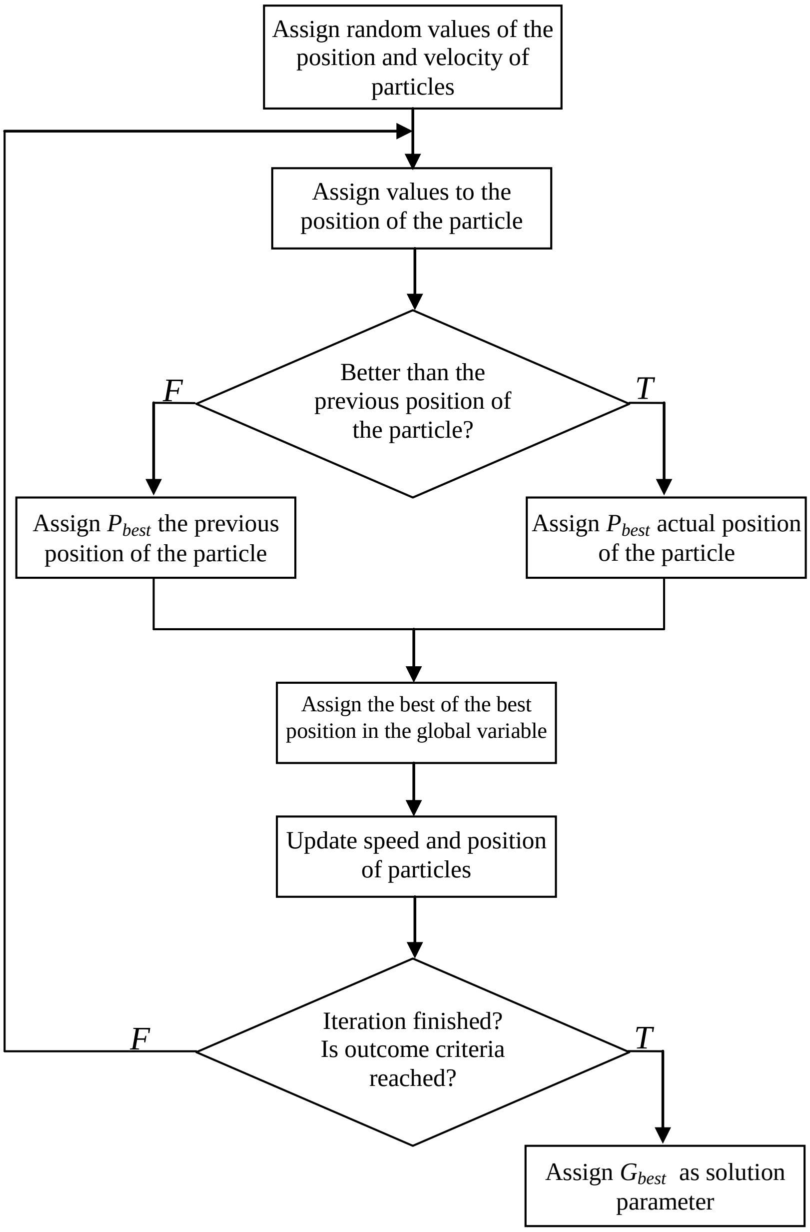


Figure 6. Flowchart for the algorithm of PSO

4. Determination of Global Minimum and Maximum Points

To test the success of the improved PSO algorithm, the global minimum and maximum points of the solution was carried out with the mathematical expression on the surface and equations including many local maximum and minimum points in Eq. 11. Particles is defined as $P = [x_1, x_2]$ since this function depends on two variables.

$$y = 3(1-x_1)^2 e^{-(x_1{}^2+(1+x_2)^2)} - \left(2x_1 - 10(x_1{}^3 + x_2{}^5)\right) e^{-(x_1{}^4+x_2{}^4)} - \frac{1}{3} e^{-((1+x_1)^2+x_2{}^2)} \quad (11)$$

In order to find the global minimum and maximum points for surface defined by the function, 10 particles and 100 iteration were used for PSO training. Position value of the piece in the first and the hundredth operations are given in the Table 1.

At the end of PSO training, the solution of minimum points are found as in Eq. 12 and the solution of minimum points are found as in Eq. 13 It can be seen in Fig. 7 and Fig. 8 that the results of function is very close to min/max points.

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.4870 \\ -1.0618 \end{pmatrix} \Rightarrow Min_x y = -2.9224 \quad (12)$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -0.2169 \\ 1.0486 \end{pmatrix} \Rightarrow Max_x y = 3.8783 \quad (13)$$

Table 1. The First and Last Position Value for Global Minimum Points

1. Iteration Position		*100. Iteration Position*	
x_1	x_2	x_1	x_2
-1.0928	0.0243	0.4869	-1.0618
-0.5286	-0.2248	0.4870	-1.0617
-1.6138	-0.1029	0.4870	-1.0617
-0.7880	0.1565	0.4870	-1.0617
-0.1533	0.1728	0.4870	-1.0617
-0.5657	-0.5811	0.4870	-1.0618
-2.1950	1.0851	0.4870	-1.0617
-0.2389	1.9996	0.4870	-1.0618
-1.1736	-1.2964	0.4870	-1.0618
-1.0244	-1.2227	0.4908	-1.0662

Table 2. The First and Last Position Value for Global Maximum Points

1. Iteration Position		*100. Iteration Position*	
x_1	x_2	x_1	x_2
0.4818	1.6187	-0.2169	1.0486
0.6383	0.4572	-0.2168	1.0486
0.0504	1.8080	-0.2169	1.0486
-0.9331	0.1063	-0.2168	1.0485
0.4724	-0.2848	-0.2169	1.0487
0.1852	-0.5866	-0.2170	1.0486
-0.2825	-0.5760	-0.2169	1.0486
-1.5402	-0.3962	-0.2169	1.0486
0.0238	1.3984	-0.2169	1.0486
-0.8552	-1.1535	-0.2169	1.0486

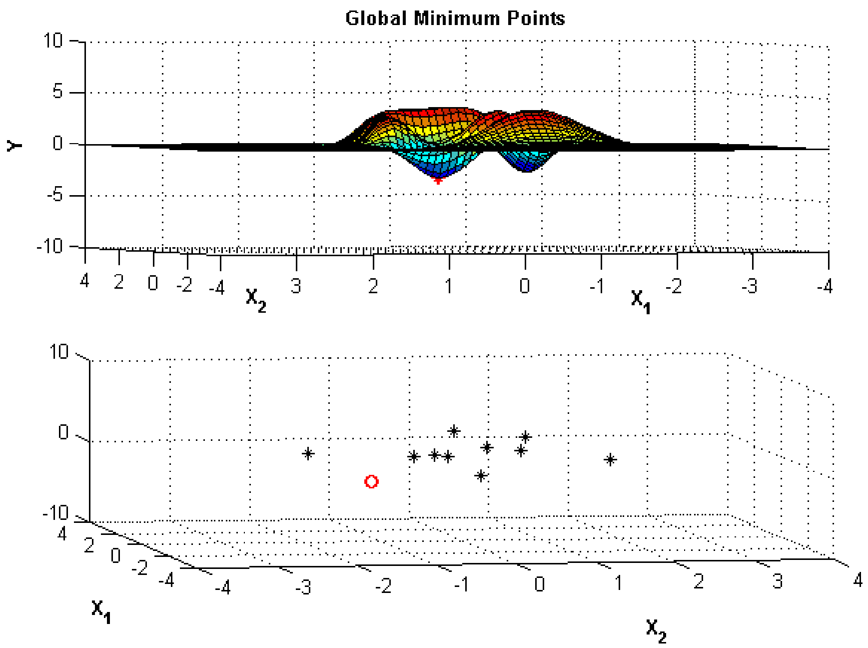


Figure 7. Global Minimum Points

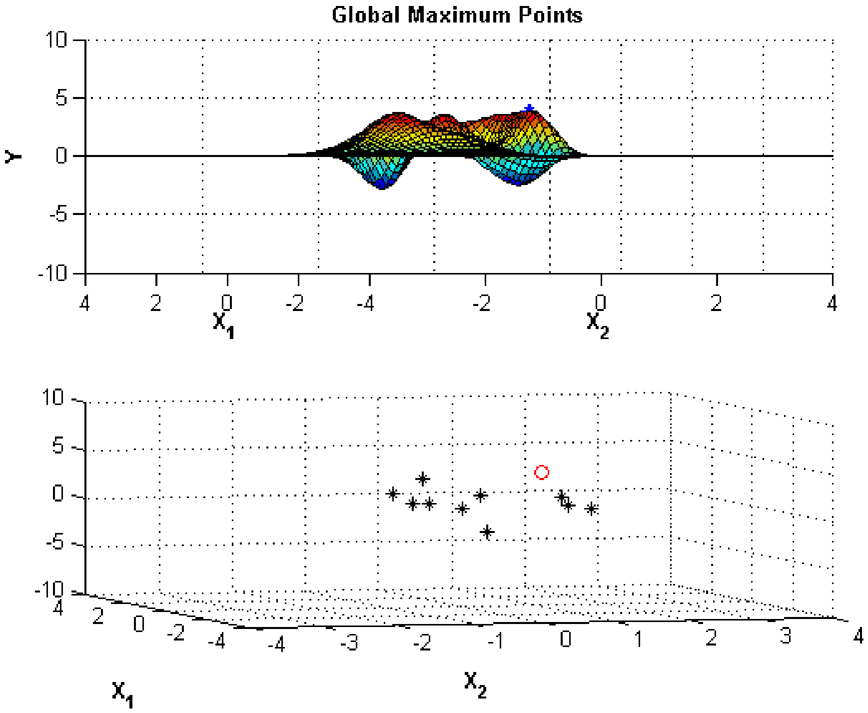


Figure 8. Global Maximum Points

During this study, real-time data were collected using NI USB 6218 DAQ card. According to the reference motor speed, motor speed was controlled by ANFIS controller. In Matlab, optimization of the parameters of the ANFIS controller carried out using closed loop control block in Fig. 9. After training phase, PMDC motor speed is controlled with ANFIS which is operated with these parameters

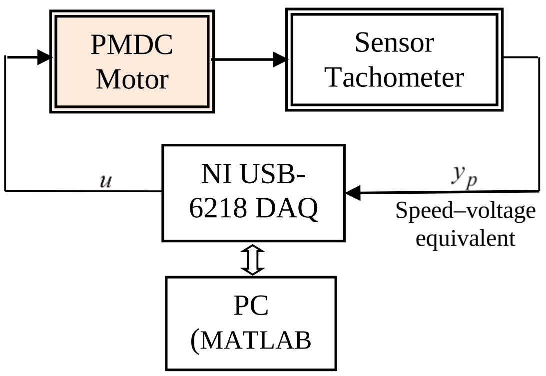
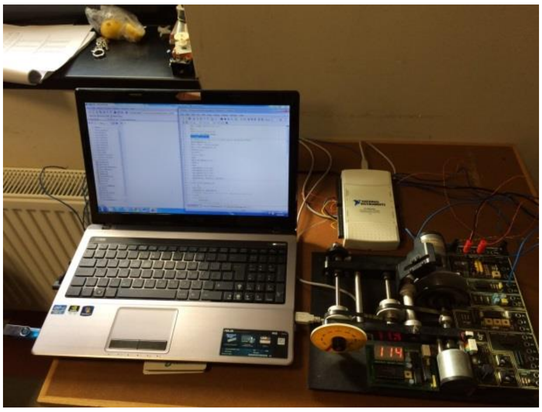


Figure 9. ANFIS Controller and PMDC Motor Control Block

5. Experimental Results

Fig. 10 shows ANFIS control surface. ANFIS implements the control surface after 100 training iterations, the controller (fuzzy-neural network) seems to bring about a completely different position in space. When default value in the positive great value, if the output of the system is a negative value, the controller output the system will be applied to a big positive control signal to bring the system to the desired positive value. Similarly, if the default value is negative and the system output is positive, this time the controller system will generate a negative output control signal to bring the desired value. System output and the default value of the response to the controller for a specific range of values is the soft transition surface shape as shown in the figure.



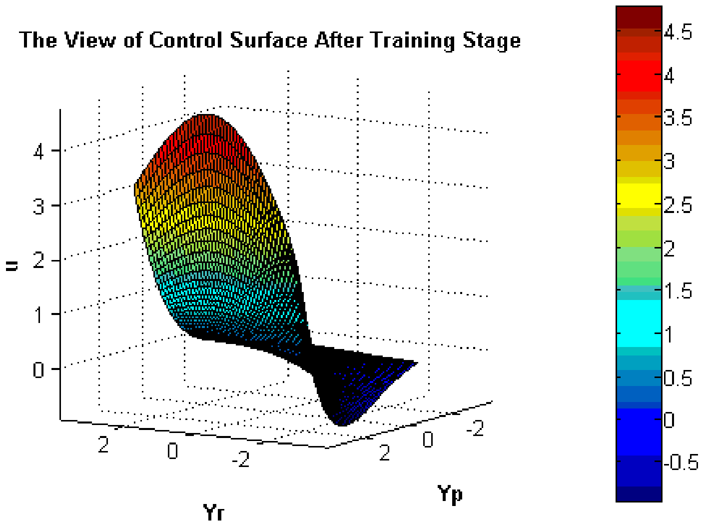


Figure 10. ANFIS control surface

Fig. 11 shows comparison in engine-speed-control output graph, cost function-iteration graph, error between the reference speed and engine speed graph. It is observed that the motor speed followed the reference speed with error rate as shown in figure.

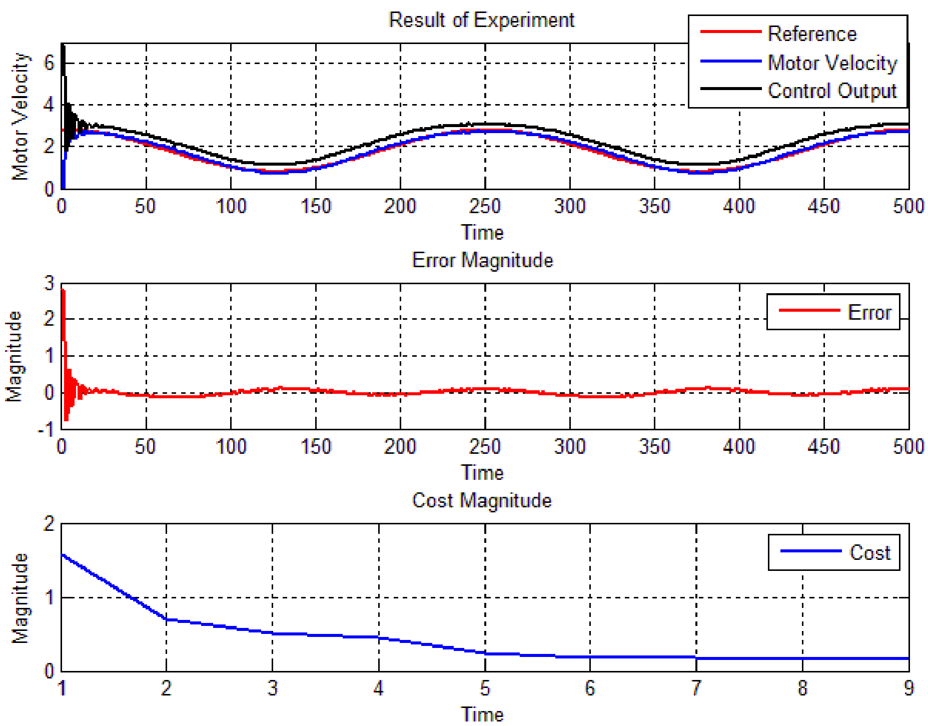


Figure 11. Motor-speed-control output graph, Error-time graph, cost function-time graph

6. Conclusion

Mainly in this study training of fuzzy-neural networks with particle swarm optimization algorithm and interpretation of the results was performed. There are 8 parameters for 4 Gaussian shaped membership function in the input of the network of ANFIS structure and the output side has 12 parameters for 4 rules. As a result of repeated 100 times with 0.3 sec sampling period, successful control results shown in Fig. 11 are obtained. When looking at response speed of the DC motor for variable default value, the system seems to sit on the desired value in a short time with less suspended and steady-state error. Reference interval is selected as $0 - 5$ V range. Values above or below the reference value from the controller are rounded to the value in the specified reference range. This study used two input membership function ANFIS structure, better results by increasing the input membership functions are expected to be obtained subsequent studies. Except sinusoidal reference signals as input can be made for experimenting performance comparisons.

Acknowledgment

This work has been realized at Process Control Laboratory at Kocaeli University. The authors thanks to Process Control Laboratory Research and Development group for their contributions.

References

[1] Jyh-Shing Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *Transactions on Systems, Man, and Cybernetics*, Vol. 23, No: 3, May/June 1993
[2] http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.htm (ziyaret tarihi: 01/06/07)
[3]Karaboga, D., "An Idea Based On Honey Bee Swarm for Numerical Optimization. Technical Report-TR06", Erciyes University Engineering Faculty Computer Engineering Department, Kayseri, 2005
[4] Fuqing Zhao, Zongyi Ren, Dongmei Yu, Yahong Yang, "Application of AnImproved Particle Swarm Optimization Algorithm for Neural Network Training",0-7803-9422-4/05/2005 IEEE
[5] C. F. Juang, Chao-Hsin Hsu, "Temperature Control by Chip-ImplementedAdaptive Recurrent Fuzzy Controller Designed by Evolutionary Algorithm", IEEETransactions on Circuits ans Systems-1: Regular Papers, Vol.52, No.11, November, 2005
[6] C. F. Juang, C. F. Lu,, "Load-frequency control by hybrid evolutionary fuzzyPI controller", IEE Proc.-Gener. Transm. Distrib, Vol. 153, No: 2, March 2006
[7] S. P. Ghoshal, "Optimizations of PID gains by particle swarm optimizationsin fuzzy based automatic generation control", Electric Power Systems Research,Volume 72, Issue 3, s: 203-212, 15 December 2004
[8] Jih-Gau Juang, Bo-Shian Lin, Kuo-Chih Chin, "Automatic Landing ControlUsing Particle Swarm Optimisation", Proceedings of the IEEE InternationalConference on Mechatronics, July 10-12 2005, Taipei, Taiwan
[9] [Çura T., "Modern sezgisel teknikler ve uygulamaları, Papatya Yayıncılık Eğitim", 2008
[10] Shi, Y. ve Eberhart, R. C. "A modified particle swarm optimizer", Proceedings of the IEEE International Conference on Evolutionary Computation s: 69-73. IEEE Press, Piscataway, NJ, 1998.
[11] Jalilvand A., Kimiyaghalam A., Ashouri A., Mahdavi M., "Advanced Particle Swarm Optimization-Based PID Controller Parameters Tuning", *12th IEEE International Multitopic Conference*, Karachi, Pakistan. 2008
[12] Tamer S, Karakuzu C, "Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri", *ELECO 2006 Elektrik-Elektronik-Bilgisayar Sempozyumu*, Elektronik Bildirileri Kitabı, (302-306), Bursa, Türkiye. 2006
[13] Kennedy, J., Eberhart, R., "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks*, (pp. 1942-1948), WA, USA. 1995
[14] Allaoua B., Gasbaoui B., Mebarki B., "Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy", *Leonardo Electronic Journal of Practices and Technologies,* ISSN 1583-1078, Issue 14, (p.19-32), 2009