Australian Journal of Basic and Applied Sciences, 5(10): 1499-1504, 2011
ISSN 1991-8178

Control DC Motorspeed with Adaptive Neuro-Fuzzy control (ANFIS)

[1]Mohamad Reza Dastranj, [1]Elahe Ebrahimi, [1]Nemat Changizi and [2]EsmaeilSameni

[1]MSC Student, Islamic Azad University, Gonabad Branch, Iran
[2]Student Islamic Azad University Gonabad, Branch, Iran

Abstract: This paper presents an application of Adaptive Neuro-Fuzzy Inference System (ANFIS) control for DC motor speed. First, the controller is designed according to Fuzzy rules such that the systems are fundamentally robust. Secondly, an adaptive Neuro-Fuzzy controller of the DC motor speed is then designed and simulated; the ANFIS has the advantage of expert knowledge of the Fuzzy inference system and the learning capability of neural networks, a perfect speed tracking with no overshoot, give better performance and high robustness than those obtained by the fuzzy alone.

Key words: Nonlinear control; Fuzzy logic controller; DC Motor speed control; Neuro-Fuzzy; ANFIS.

INTRODUCTION

In spite of the development of power electronics resources, the direct current machine became more and more useful. Nowadays their uses isn't limited in the car applications (electrics vehicle), in applications of weak power using battery system (motor of toy) or for the electric traction in the multi-machine systems too.

The speed of DC motor can be adjusted to a great extent as to provide controllability easy and high performance (Hénao, 2002; Raghavan, 2005). The controllers of the speed that are conceived for goal to control the speed of DC motor to execute one variety of tasks, is of several conventional and numeric controller types, the controllers can be: PID Controller, Fuzzy Logic Controller; or the combination between them: Fuzzy-Neural Networks, Fuzzy-Genetic Algorithm.

The Adaptive Neuro-Fuzzy Inference System (ANFIS), developed in the early 90s by Jang (1993), combines the concepts of fuzzy logic and neural networks to form a hybrid intelligent system that enhances the ability to automatically learn and adapt. Hybrid systems have been used by researchers for modeling and predictions in various engineering systems. The basic idea behind these neuro-adaptive learning techniques is to provide a method for the fuzzy modeling procedure to learn information

About a data set, in order to automatically compute the membership function parameters that best allow the associated FIS to track the given input/output data. The membership function parameters are tuned using a combination of least squares estimation and back-propagation algorithm for membership function parameter estimation. These parameters associated with the membership functions will change through the learning process similar to that of a neural network. Their adjustment is facilitated by a gradient vector, which provides a measure of how well the FIS is modeling the input/output data for a given set of parameters. Once the gradient vector is obtained, any of several optimization routines could be applied in order to adjust the parameters so as to reduce error between the actual and desired outputs. This allows the fuzzy system to learn from the data it is modeling. The approach has the advantage over the pure fuzzy paradigm that the need for the human operator to tune the system by adjusting the bounds of the membership functions is removed.

This proposes an application of ANFIS. The best range and shapes of member ships functions obtained with ANFIS Simulation results are given to show the effectiveness of ANFIS controller.

Mmodel of DC Motor:

DC machines are characterized by their versatility. By means of various combinations of shunt-, series-, and separately-excited field windings they can be designed to display a wide variety of volt-ampere or speed-torque characteristics for both dynamic and steady-state operation. Because of the ease with which they can be controlled systems of DC machines have been frequently used in many applications requiring a wide range of motor speeds and a precise output motor control (Halila, 2001; Capolino, 2001).

In this paper, the separated excitation DC motor model is chosen according to his good electrical and mechanical performances more than other DC motor models. The DC motor is driven by applied voltage. Fig.1 show the equivalent circuit of DC motor with separate excitation. The Symbols, Designations and Units are publicized in Table 1.

The characteristic equations of the DC motor are represented as:

$$\frac{d}{dt} i_{ex} = \left(-\frac{R_{ex}}{L_{ex}}\right) . i_{ex} + \left(\frac{1}{L_{ex}}\right) . V_{ex} \tag{1}$$

Corresponding Author: Mohamad Reza Dastranj, MSC Student, Islamic Azad University, Gonabad, Branch, Iran.
E-mail: mohamadrezadastranj@gmail.com

$$\frac{d}{dt}i_{ind} = \left(-\frac{R_{ind}}{L_{ind}}\right).i_{ind} + \left(-\frac{L_{index}}{L_{ind}}\right).W_r.I_{ex} + \left(\frac{1}{L_{ind}}\right).V_{ind} \tag{2}$$

$$\frac{d}{dt}W_r = \left(-\frac{L_{index}}{J}\right).i_{ex}.i_{ind} + \left(-\frac{Cr}{J}\right).W_r.I_{ex} + \left(\frac{-fc}{J}\right).W_r \tag{3}$$

Table 1: Symbols, Designations and Units.

Symbols	Designations	Units
iexandiend	Excitation current and Induced current.	[A]
wr	Rotational speed of the DC Motor.	[Rad/Sec]
VexandVind	Excitation voltage and Induced voltage	[Volt]
RexandRind	Excitation Resistance and Induced Resistance.	[Ω]
Lex,Lindand Lindex	Excitation Inductance Induced Inductance and Mutual Inductance.	[mH]
J	Moment of Inertia.	[Kg.m2]
Cr	Couple resisting.	[N.m]
fc	Coefficient of Friction.	[N.m.Sec/Rad]

The equivalent circuit of DC motor with separate excitation illustrated in Fig. 1.

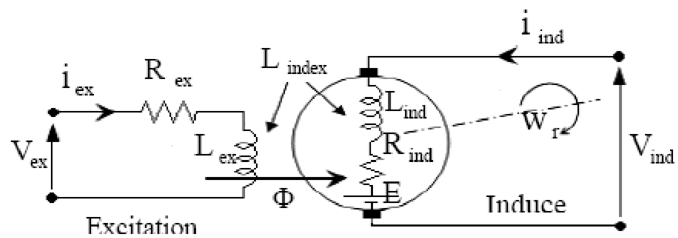

Fig. 1: Equivalent circuit of DC motor with Separate Excitation.

From the state equations (1), (2), (3) previous, can construct the model with the environment MATLAB. The model of the DC motor in Simulink is shown in Fig. 2. The various parameters of the DC motor are shown in Table 2.

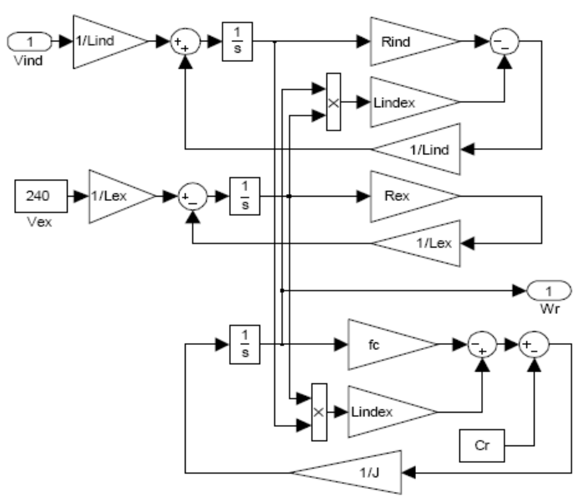

Fig. 1: Model of the DC Motor in Simulink.

Table 2: Parameters of the DC Motor.

Vex=240[V]	Lind=0.012[mH]
Vind=240[V]	Lindex=1.8[mH]
Rex=240[Ω]	J=1[Kg.m2]
Rind=0.6[Ω]	Cr=29.2[N.m]
Lex=120[mH]	fc=0.0005[N.m.Sec/Rad]

Into the expressions for N and P. Rather than continuing with algebra here, we will simply represent these equations in Simulink.

Simulink can work directly with nonlinear equations,

Adaptive Neuro-Fuzzy controller:

The ANFIS controller generates change in the reference voltage Vref, based on speed error e and derivate in the speed error de defined as:

$$e = \omega_{ref} - \omega \tag{4}$$

$$de = [d(\omega_{ref} - \omega)]/dt \tag{5}$$

Whereωref and ω are the reference and the actual speeds, respectively.

In this study first order Sugeno type fuzzy inference was used for ANFIS and the typical fuzzy rule is:

$$if\ e\ is\ Ai\ and\ de\ is\ Bi\ then\ z = f(e, de) \tag{6}$$

Where Ai and Bi are fuzzy sets in the antecedent and

z = f (e, de) is a crisp function in the consequent.

The significances of ANFIS structure are:

Layer 1:

Each adaptive node in this layer generates the membership grades for the input vectors Ai, i =1… 5. In this paper, the node function is a triangular membership function:

$$O_i^1 = \mu_{Ai}(e) = \begin{cases} 0 & e \le a_i \\ \frac{e-a_i}{b_i-a_i} & a_i \le e \le b_i \\ \frac{c_i-e}{c_i-b_i} & b_i \le e \le c_i \\ 0 & c_i \le e \end{cases} \tag{7}$$

Layer 2:

The total number of rule is 25 in this layer. Each node output represents the activation level of a rule:

$$O_i^1 = w_1 = min(\mu_{Ai}(e), \mu_{Bi}(e)), i = 1, \ldots, 5 \tag{8}$$

Layer 3:

Fixed node i in this layer calculate the ratio of the i-th rule's activation level to the total of all activation level:

$$O_i^3 = \bar{w}_i = \frac{W}{\sum_{j=1}^{n} W_j} \tag{9}$$

Layer 4:

Adaptive node i in this layer calculate the contribution of i-th rule towards the overall output, with the following node function:

$$O_i^4 = \bar{w}_i Z_i = \bar{W}_i(p_i e + q_i de + r_i) \tag{10}$$

Layer 5:

The single fixed node in this layer computes the overall output as the summation of contribution from each rule:

$$O_i^5 = \sum_{i=1}^{2} \bar{W}_i Z_i = \frac{W_1 Z_1 + W_2 Z_2}{W_1 + W_2} \tag{11}$$

The parameters to be trained are ai, bi and ci of the premise parameters and pi, qi, and ri of the consequent parameters. Training algorithm requires a training set defined between inputs and output (Jang, 1993). Although, the input and output pattern set have 150 rows. Figure 3 shows optimized membership function for e and de after trained. Figure 4 shows Surface plot showing relationship between input and output parameters after trained. Figure 5 shows The ANFIS model structure.

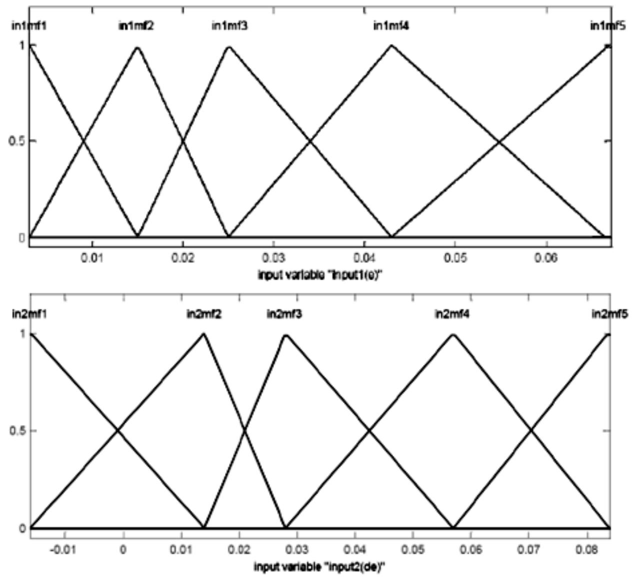

Fig. 3: Membeship functions for e and de after trained.

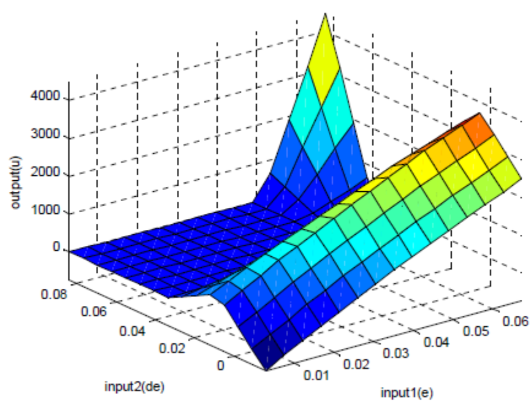

Fig. 4: Surface plot showing relationship between input and output parameters.

The number of epochs was 100 for training. The number of MFs for the input variables e and de is 5 and 5, respectively. The number of rules is then 25 (5×5 = 25). The triangular MF is used for two input variables. It is clear from (14) that the triangular MF is specified by two parameters.

Therefore, the ANFIS used here contains a total of 95 fitting parameters, of which 20 (5×2 + 5×2 = 20) are the premise parameters and 75 (3×25 = 75) are the consequent parameters.

The training and testing root mean square (RMS) errors obtained from the ANFIS are 4.7×10-6 and 5.3×10-6 respectively

Table 3: Performances of two controllers

Results	Fuzzy Logic ontroller (FLC)	FLC using neural networks(ANFIS)
Rising time [Sec]	0.00385	0.00301
Overtaking [%]	0	0
Steady state error[%]	9×10-4	3.2×10-4

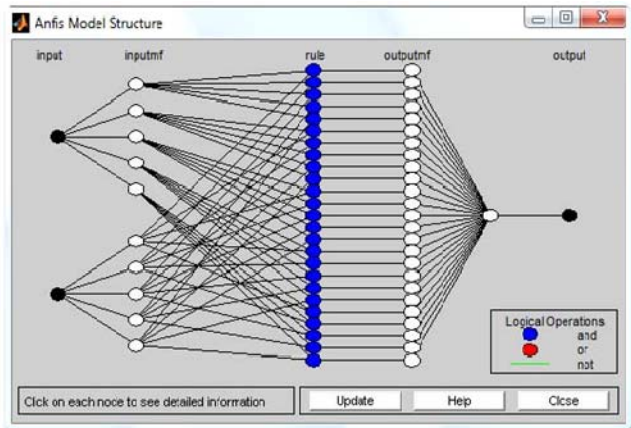

Fig. 5: The ANFIS model structure.

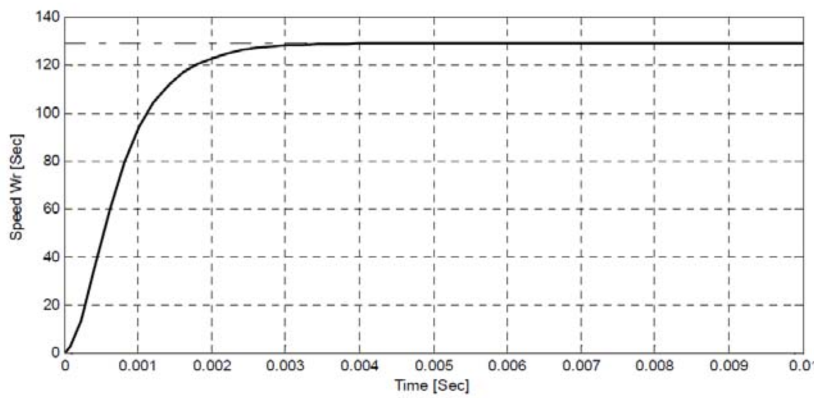

Fig. 6: The speed response of Fuzzy Logic Controller(FLC).

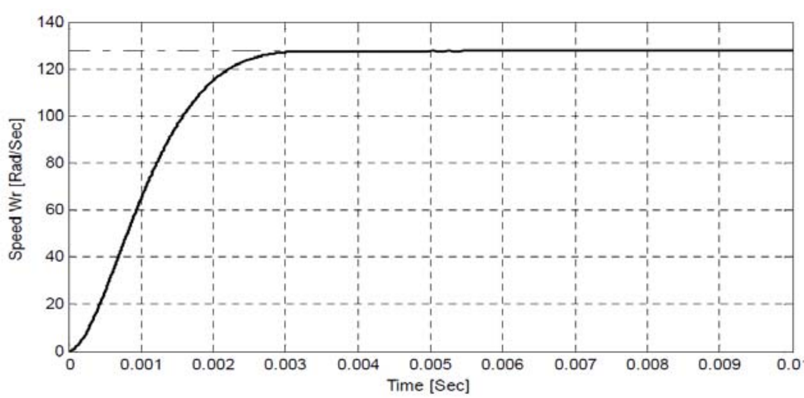

Fig. 7: The speed response of FLC using neural networks (ANFIS).

Conclusion:

In this paper, The speed of a DC Motor drive is controlled by means of three different controllers. According to the results of the computer simulation, the Adaptive Neuro-Fuzzy (ANFIS) controller efficiently is better than the traditional FLC. The ANFIS is the best controller which presented satisfactory performances and possesses good robustness (no overshoot, minimal rise time, Steady state error = 0). The major drawback of the fuzzy controller presents an insufficient analytical technique design (choice of the rules, the membership functions and the scaling factors). That we chose with the use of the Neural Networks for the optimization of this controller in order to control DC motor speed. Finally, the proposed controller (ANFIS) gives a very good results and possesses good robustness.

REFERENCES

Capolino, G.A., G. Cirrincione, M. Cirrincione, H. Henao, R. Grisel, 2001. Digital signal processing for electrical machines. Invited paper, Proceedings of ACEMP'01 (Aegan International Conference on Electrical Machines and Power Electronics), Kusadasi (Turkey), pp: 211-219.

Clerc, M., J. Kennedy, 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evolutionary Computation, 6: 58-73.

Constantin, V.A., 1995. Fuzzy logic and neuro-fuzzy applications explained. Englewood Cliffs, Prentice-Hall.

Halila, A., 2001. Étude des machines à courant continu. MS Thesis, University of LAVAL, (Text in French).

Hénao, H., G.A. Capolino, 2002. Méthodologie et application du diagnostic pour les systèmes électriques. Article invité dans Revue de l'Electricité et de l'Electronique (REE), (Text in French), 6: 79-86.

Jang, J.S.R., 1993. Adaptive network based fuzzy inference systems. IEEE Transactions on systems man and cybernetics, pp: 665-685.

Kennedy, J., R. Eberhart, 1995. Particle swarm optimization. Proc. IEEE Int. Conf. on Neural Network, 4: 1942-1948.

Kim, J., N. Kasabov, Hy FIS, 1999. Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Networks.

Lin, C.T., C.S.G. Lee, 1996. Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems. Upper Saddle River, Prentice-Hall.

Raghavan, S., 2005. Digital control for speed and position of a DC motor. MS Thesis, Texas A&M University, Kingsville.

Shi, Y., R. Eberhart, 1998. A modified particle swarm optimizer. Proc. 1998 Int. Conf. on Evolutionary Computation–The IEEE World Congress on Computational Intelligence, Anchorage, pp: 69-73.