Research Journal of Applied Sciences, Engineering and Technology 10(5): 570-580, 2015
DOI:10.19026/rjaset.10.2465
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 8, 2015 Accepted: February 5, 2015 Published: June 15, 2015

Research Article

Speed Control of a DC Motor for the Orientation of a Heliostat in a Solar Tower Power Plant using Artificial Intelligence Systems (FLC and NC)

Abdelfettah Zeghoudi and Ali Chermitti
URMER Research Unit, Faculty of Sciences, University of Abou-bekr Belkaid, Tlemcen, Algeria

Abstract: Recent research in the field of motor controls is becoming more interesting especially with the developments of new control methods. This study shows a comparative study between two artificial intelligence methods namely neural networks and fuzzy logic for the speed control of a DC motor for the orientation of a heliostat in a solar tower power plant. The speed controller of DC motor is performed using two Fuzzy Logic Configuration (FLC1 and FLC2) and neural controller in MATLAB environment. The simulation results are used to make sure the real possibility of using artificial intelligence systems to identify and control this type of installation. The performance of the Fuzzy Logic Controller and Neural control are compared using different errors metrics. The results show that Fuzzy logic method is more efficient compared with neural controller method.

Keywords: Artificial intelligence, DC motor, fuzzy logic, heliostat, neural controller, solar tower power plant, speed control

INTRODUCTION

DC motors have been popular in the control industry area for a long time, because they have many good characteristics. In addition, high reliability, flexibility and low cost of DC motors lead to use them in different applications such as robot manipulators, tracking systems and home appliances… etc. However, all control systems of DC motors suffer from problems related to undesirable overshoot, longer settling times, vibrations and stability when switching from state to another. In addition, control of DC motors is known as big challenge because of the nonlinearity presented in the concept of these motors. Hence, several studies have been carried out in this problem where they proposed new methods to control the speed and the rotation of DC motors based on artificial intelligence methods such as fuzzy logic, neural network and hybrid fuzzy-neural methods. Rigatos (2009), an adaptive fuzzy control of DC motors using state and output feedback is showed. Michele and Bartolomeo (2014), a PID fuzzy logic controller is applied for the control of a distillation column. However, single models are limited. Hence, hybrid models are proposed in literature. Premkumar and Manikandan (2014), it presented a fuzzy neural network IP controller for robust position control of induction motor drive. Besir (2009), Speed control of permanent magnet excitation transverse flux linear motor by using adaptive neuro-fuzzy controller is shown. Hasanien *et al*. (2010), adaptive Neuro-Fuzzy Inference System based speed controller for brushless DC motor is studied.

In addition, knowing the non-linearity in the motor helps to use an adaptive tracking control methods with the technique of input-output linearization (Ibbini and Zakaria, 1996; Kim *et al*., 1997). However, when these nonlinearities or disturbances are unknown, neural or fuzzy control is more suitable for succeeding satisfactory performance of the closed-loop system (Rahma and Hoque, 1998; Rubaai *et al*., 2002). Moreover, no one of the previous mentioned papers shows the best and robust artificial method is the best for such problems. For that, we proposed in this study, a comparative study of a controller of a separately excited DC motor for the orientation of a heliostat in a solar tower power plant modeling developed by employing it in MATLAB software. First, two Fuzzy Logic Controllers (FLC1 and FLC2) are tested. FLC1 are differ from FLC2 in the number of rules and inputs. In FLC1, we used two inputs (error and derivate of the error) and five rules. Where, in FLC2, we only used one input (the error) with three rules. Second, different neural network architectures and configuration is tested in the speed control. Finally a comparative study between neural, FLC1 and FLC2 controllers is discussed to choose the best technique for controlling the speed of DC motor for the orientation of a heliostat.

SYSTEM DISCRETION

The solar field consisting of a large number of tracking mirrors called heliostats. One heliostat comprises a set of mirrors, a tracking system, a frame, a

Corresponding Author: Abdelfettah Zeghoudi, URMER Research Unit, Faculty of Sciences, University of Abou-bekr Belkaid, Tlemcen, Algeria, Tel.: +213 552 99 88 20

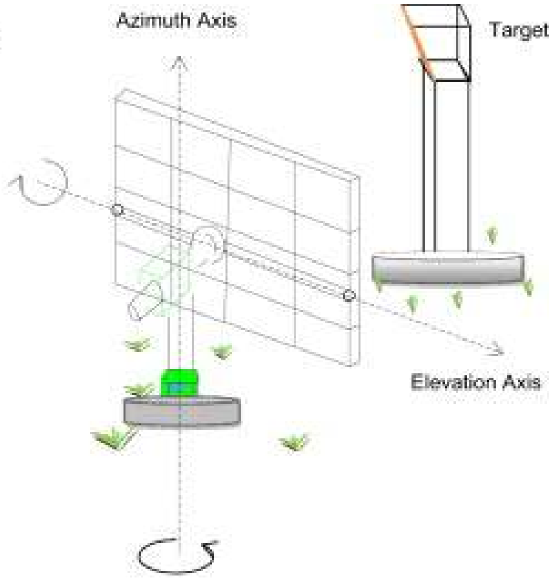


Fig. 1: Basic concept of heliostat

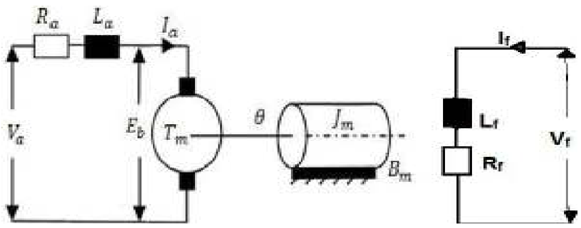


Fig. 2: Separately excited DC motor model
V_a, V_f: The armature and field are voltage (V); I_a, I_f: The armature and field are current (A); L_a, L_f: The armature and field are inductance (H); R_a, R_f: The armature and field are resistance (Ω); T_m: The mechanical is torque developed (Nm); ϴ: The rotor is speed (rad/sec); J_m: Is moment of inertia (Kg.m^2); B_m: Is friction coefficient of the motor (Nm/ (rad/sec))

base of the structure and the control system. The single heliostat basic design is illustrated in Fig. 1.

The heliostat return on the field is a function of the optical efficiency. Cosine effect, shading effect, blocking effect, the reflectivity of the mirror, atmospheric attenuation and receiver spill are the main factors affecting optical performance of heliostats (Solangi *et al*., 2011).

It is well known that half of the total investment cost and 40% of total energy losses are attributed to the heliostat field. It is essential to optimize the design to reduce the cost of capital and improve the overall efficiency of the plant (Kadir *et al*., 2010; Solangi *et al*., 2011).

The heliostat tracking the sun can be classified as either open loop system or the closed loop system (Chen *et al*., 2004). The open loop system is based on astronomical formulas concerning the position of the sun to the geometry of the system. This system is reliable, low cost and is recommended for larger solar field because the heliostat is under the control of the computer. On the other hand, the closed loop system uses the probe to follow the sun. This system is so precise and very useful for small fields of heliostats. Because of wind, own weight and mechanical error heliostats, concentrates sunlight will deviate from their pre-selected path when heliostat operation, which also affects the distribution of the final heat flow in absorber. As the tracking error increases, the light spot on the focal plane array of heliostats field is expanded and the loss increases. Thus, there will be less of solar rays entering the receiver and absorbed by the absorber. And that means that when the error increases, the average annual field efficiency of the plant decreases (He *et al*., 2013).

We will present in this study a basic configuration for optimal DC motor control using artificial intelligence methods. The used motor is shown in Fig. 2.

For the DC motor, the parameters given are the following:

Armature resistance (Ra) 0.5 Ω
Armature inductance (La) 0.02 H
Armature voltage (Va) 200 V
Mechanical inertia (jm) 0.1 Kg.m^2
Friction coefficient (Bm) 0.008 N.m/rad/sec
Back emf constant (k) 1.25 V/rad/sec
Motor torque constant (k) 0.5 N.m/A

THE TRAJECTORIES OF A HELIOSTAT IN A SOLAR TOWER

Numerical experiments were performed for specific day of the year: 1st January (J1-an average day of winter) (Fig. 3), between 09:00 and 10:00 (Fig. 4). It was considered a typical heliostat located opposite the central tower receiver has a square shape (L = 3 m, D = 100 m).

For the calculation of theoretical positions heliostats, it was assumed that the heliostat and tower were located in Toulouse, France. The case of Central THEMIS. The two paths are shown in the following figures.

On most plants, heliostats do not pursue the Sun a continuous basis. They perform corrections on each axis every few seconds or every minute.

That is to say, when the sun has shifted his thoughtful work on the receiver has also shifted the heliostat if not corrected. This correction is performed on the central THEMIS whenever the azimuth (or elevation) shows an error reaching 0.15 mrad. This is a fixed angular correction and not the impact on the

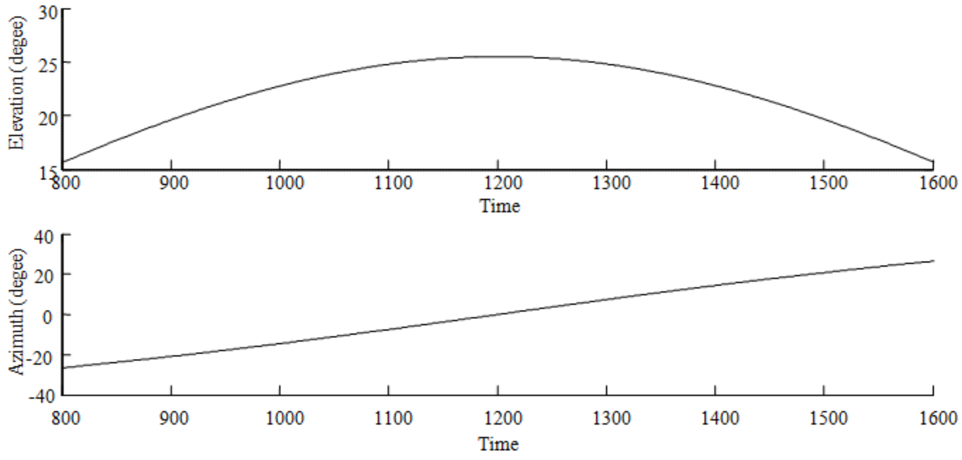


Fig. 3: Heliostat's movements (azimuth and elevation)-day 01

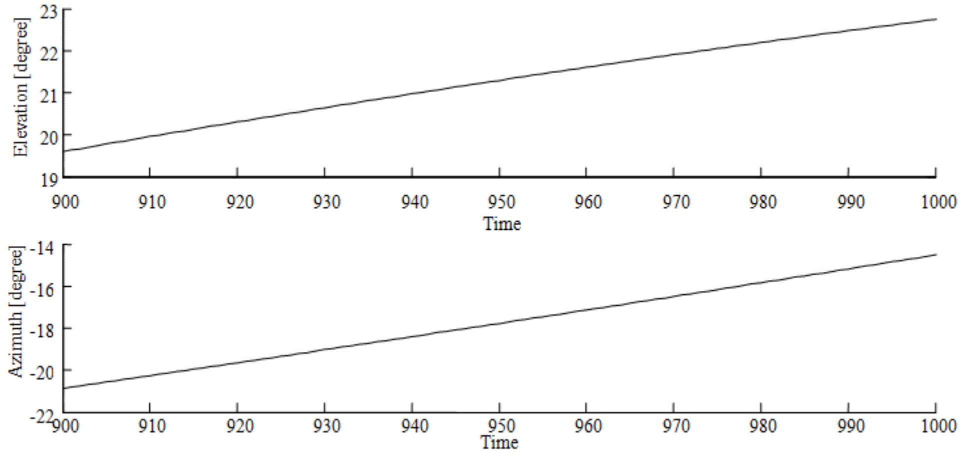


Fig. 4: Heliostat's movements (azimuth and elevation)-day 01 between 09:00 and 10:00

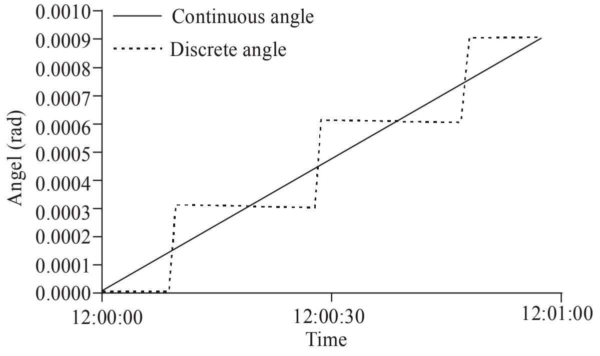


Fig. 5: Discretization references

receiver error. Thus it reaches angular threshold, an absolute correction 0.3 mrad is performed in the opposite direction of movement to the task. The azimuth is always fixed in the same direction as the azimuth is growing over a day. For elevation, the correction can be performed in either direction since the Sun rises initially and down after solar noon. The correction of 0.3 mrad.

The azimuth and elevation angles are discretized every 0.3 mrad during the day. This principle leads to a temporary lead of 0.15 mrad (Fig. 5).

FUZZY AND NEURAL CONTROLLER DESCRIPTION AND DESIGN

In this section, we will present a detailed description, conception and configurations used in the simulation part for both fuzzy and neural controllers.

Fuzzy logic controller: There are specific components characteristic of a fuzzy controller to support a design procedure. Figure 6 shows the controller between the preprocessing block and post processing block (Vadher, 2014).

Preprocessing: The physical inputs of some measuring equipment are often hard or crisp compared to linguistic inputs. Hence, a preprocessing of inputs is needed, as illustrated in the first block in Fig. 6, which shows the measurement conditions before enter the controller (Vadher, 2014).

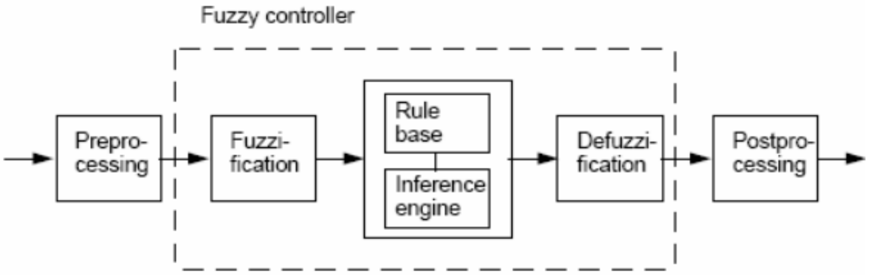


Fig. 6: Structure of fuzzy logic controller

Fuzzification: The first block inside the controller is fuzzification, which converts each piece of input data to membership by a lookup in one or several membership functions. The fuzzification block matches the input data with the conditions of the rules to be determined. There is a degree of membership for each linguistic term that applies to the input variable (Vadher, 2014).

Rule base: The collection of rules is called a rule base. The rules are in "If-Then" format and formally the 'If' side is called the conditions and the 'Then' side is called the conclusion. The computer is able to execute the rules and compute a control signal depending on the measured inputs error (*e*) and change in error (*de*). In a rule based controller the control strategy is stored in a more or less natural language. A rule base controller is easy to understand and to maintain for a non-specialist end user and an equivalent controller could be implemented using conventional techniques (Vadher, 2014).

Defuzzification: Defuzzification is when all the actions that have been activated are combined and converted into a single non-fuzzy output signal, which is the control signal of the system. The output levels depend on the rules of the systems .The positions depend on the non-linearities existing to the systems. To achieve the result, develop the control curve of the system representing the input/output relation of the systems and based on the information; define the output degree of the membership function with the aim to minimize the effect of the non-linearity (Vadher, 2014).

Postprocessing: The post processing block often contains an output gain that can be tuned and also become as an integrator (Vadher, 2014).

Mamdani fuzzy inference: The most commonly used fuzzy inference technique is the Mamdani method (Mamdani and Assilian, 1975). They attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators.

It uses the error *e* (*k*) and the variable error *d* (*k*) to produce changes in function of the output driver (may be *μ* (*k*) or *Dμ* (*k*)):

- $e(k) = w(k) - y(k)$
- $De(k) = e(k) - e(k-1)$
- $u(k) = F(e(k).De(k))0$
- $Du(k) = F\big(e(k).De(k)\big)$

e (*k*) is defined as the point minus the output:

- $if\ e(k) > 0\ then\ w(k) > y(k)$
- $If\ De(k) > 0\ then\ w(k) > y(k)$

FLC1 and FLC2 configurations: To design the Fuzzy Controller two different approaches (FLC1 and FLC2) are taken into account. Two rules with different sets have been defined to show that these controllers can be done in several ways and solutions that may be better than each other.

First approach: The following linguistic variables are shown in Fig.7 to 9:

- **Error inputs:** Strong Negative (Nfo), small Negative (Nfai), Null (Nu), strong Positive (Pfo), small Positive (Pfai).
- **Variation of error:** Strong Negative (Nfo), strong Positive (Pfo).
- **Output:** Decreasemuch (Dbcp), Not decreasemuch (Ndbcp), Set (T), increase much (Abcp), Not increase much (Nabcp).
- **Rules :** The rules are defined as follows:
  - If the error is Pfai then control is Abcp.
  - If the error is Nfo then control is Dbcp.
  - If the error is Pfo then control is Nabcp.
  - If the error is Nfai then control is Ndbcp.
  - If the error is Nu then control is T.
  - If the error is null and the variance is Pfo then the control is Dbcp.
  - If the error is null and variable is Nfo then control is Abcp.

The fuzzy controller SIMULINK diagram used in our simulation is shown in Fig. 10.

Second approach: In the second approach, we only used the error as an input. The configurations used in FLC2 are presented in Fig. 11 and 12:

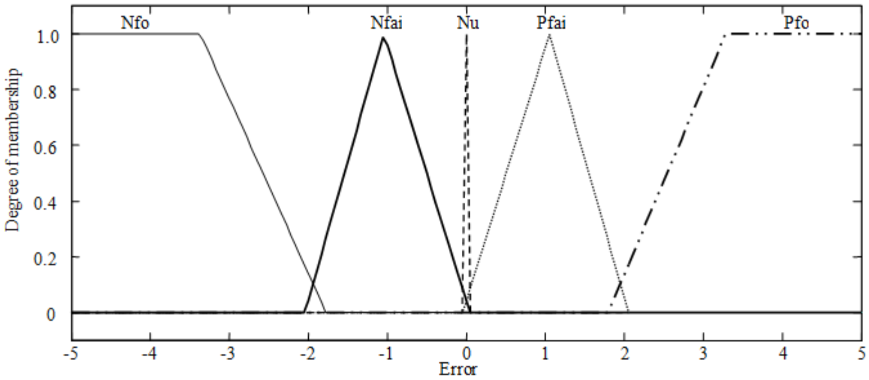


Fig. 7: Fuzzy logic 1 first input variable, error

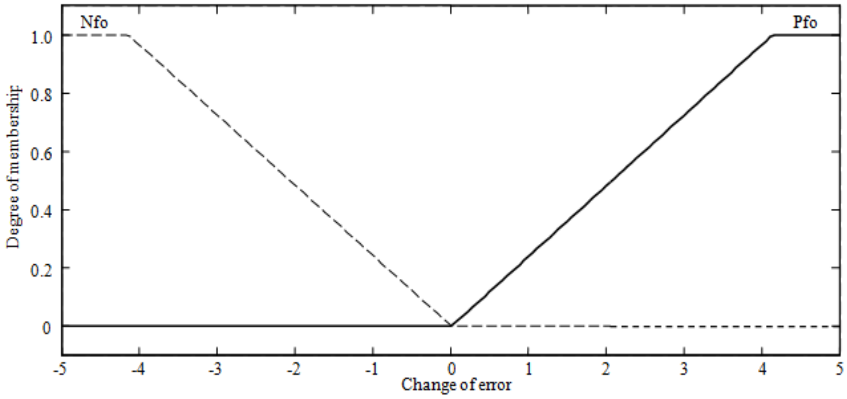


Fig. 8: Fuzzy logic 1 second input variable, change of error

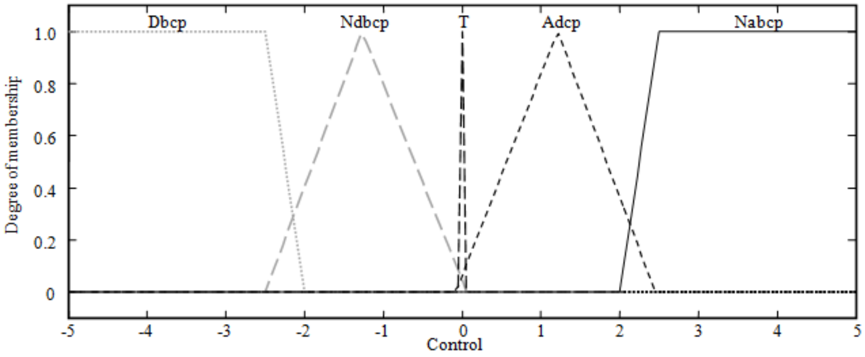


Fig. 9: Fuzzy logic 1 output variable, control

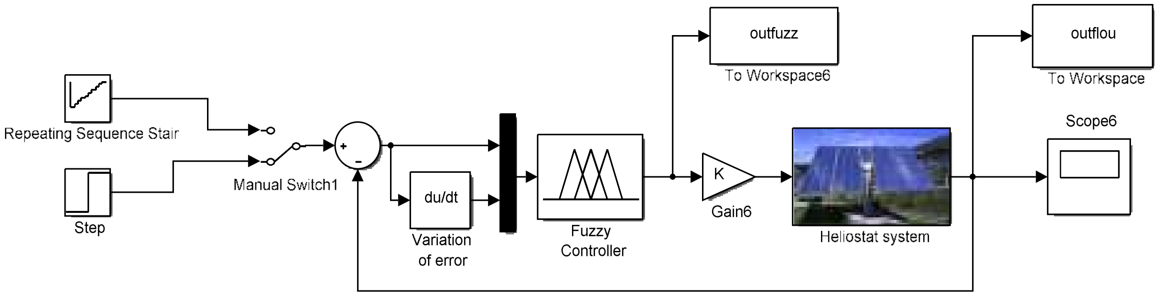


Fig. 10: Closed loop system using FLC 1

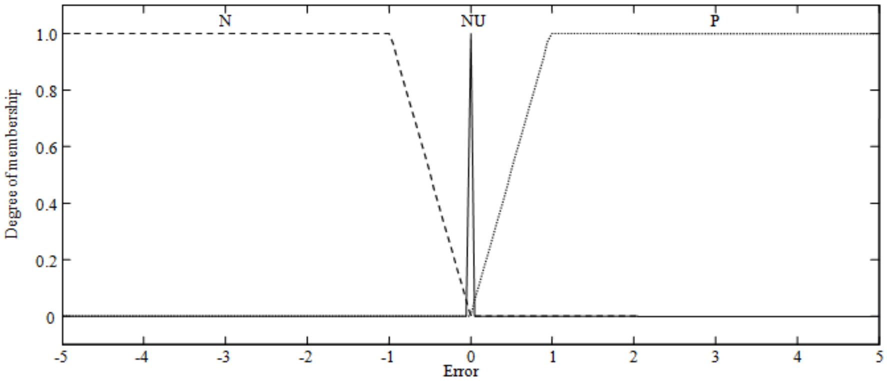


Fig. 11: Fuzzy logic 2 first input variable, error

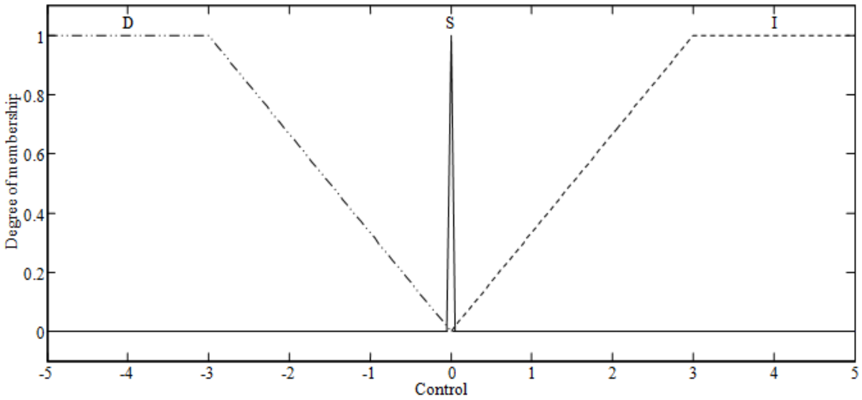


Fig. 12: Fuzzy logic 2 output variable, control

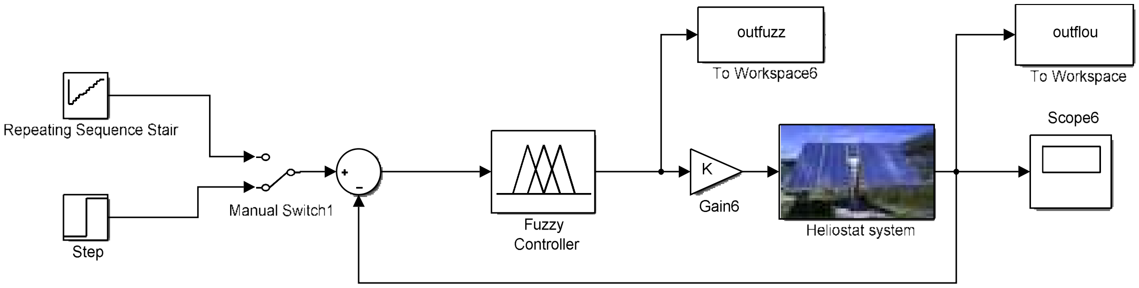


Fig. 13: Closed loop system using FLC 2

- **Inputs (error):** Negative (N), Null (Nu), Positive (P).
- **Output:** Decrease (D), constant (S), Increase (I).
- **Rules :** The rules are defined as follows:
  - If the error is N then control is D.
  - If the error is Nu then control is S.
  - If the error is P then control is I.

In SIMULINK, FLC2 diagramm is shown in Fig. 13.

Neural controller: Neural networks have been applied very successfully in the identification and control of dynamic systems. The universal approximation capabilities of the multilayer perceptron make it a popular choice for modeling nonlinear systems and for implementing general-purpose nonlinear controllers (Hagan and Demuth, 1999). There are various neural

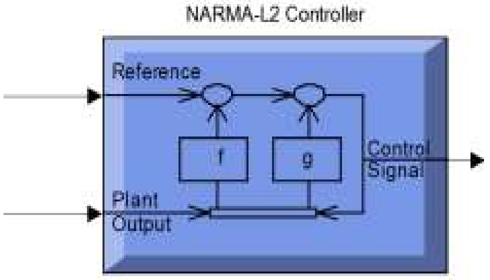


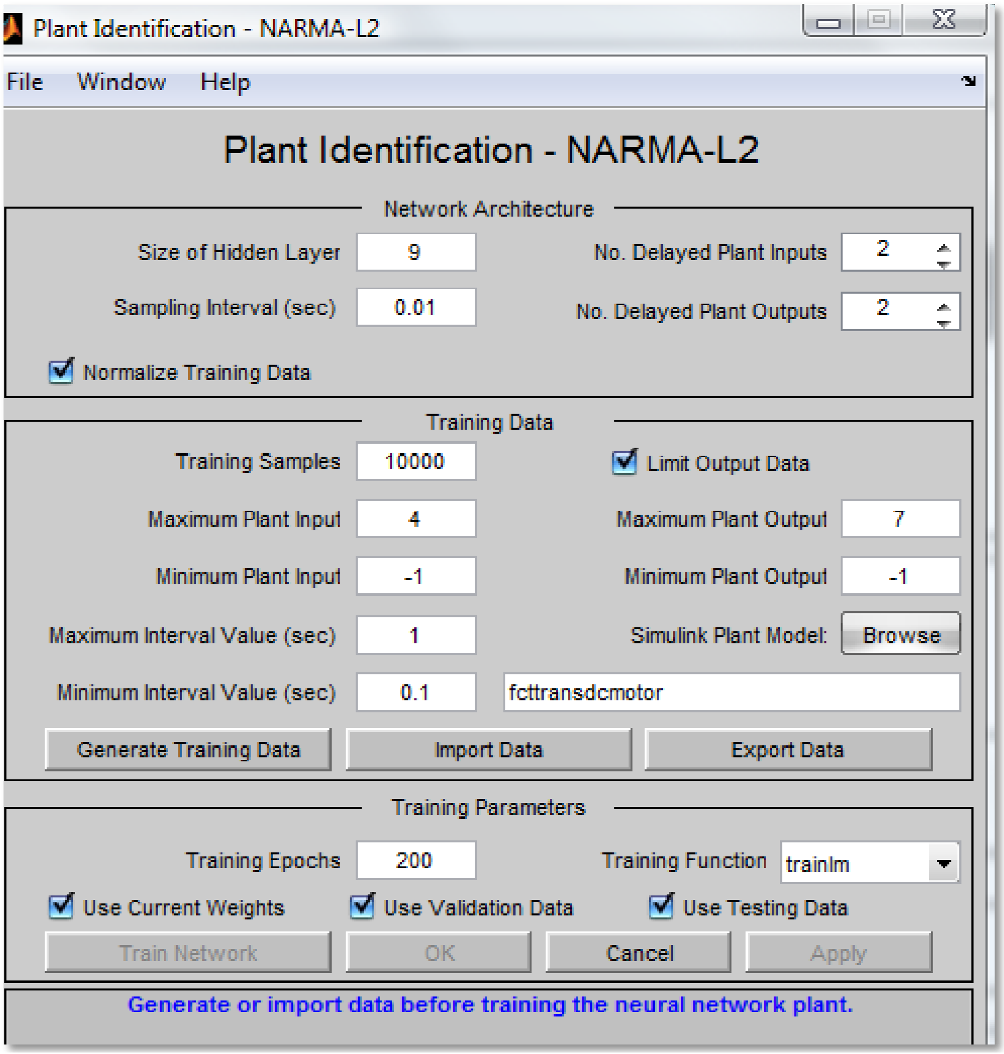
Fig. 14: Reference model controller

control strategies. Neural Networks Toolbox of MATLAB 7.1 offers three of these strategies: Model Reference Controller, NARMA-L2 controller type and Predictive Neural Network (NN Predictive) (Cosme, 2004). Here, we will utilize the NARMA-L2 controller; the first step in using feedback linearization (or NARMA-L2 control) is to identify the system to be controlled. We train a neural network to represent the forward dynamics of the system. The first step is to choose a model structure to use it (Cosme, 2004). The neural controller NARMA-L2 in the SIMULINK model is shows in Fig. 14.

The neural controller described in this section is referred to by two different names: feedback linearization control and NARMA-L2 control. It is referred to as feedback linearization when the plant model has a particular form (companion form). It is referred to as NARMA-L2 control when the plant model can be approximated by the same form. The central idea of this type of control is to transform nonlinear system dynamics into linear dynamics by canceling the nonlinearities (Howard and Mark, 2000). It is very important to note the necessity of identifying the plant, first, before proceeding with the neural controller training.

Controller requires the reference model output as an input in order to establish the behavior the plant has to develop. A good description of the plant we want to control is needed as input of the neural controller, in order to compare the desired output with the actual one (Cosme, 2004).

It is necessary to set up parameters like hidden neurons number of the neural network that will identify the plant. In addition, the sampling time T, minimum and maximum time for taking samples during training and the minimum and maximum plant input signal values in order to establish the inputs range for the controller. Figure 15 shows the input data used to identify the plant.



Plant Identification - NARMA-L2

Network Architecture

Size of Hidden Layer: 9
No. Delayed Plant Inputs: 2
Sampling Interval (sec): 0.01
No. Delayed Plant Outputs: 2
☑ Normalize Training Data

Training Data

Training Samples: 10000
☑ Limit Output Data
Maximum Plant Input: 4
Maximum Plant Output: 7
Minimum Plant Input: -1
Minimum Plant Output: -1
Maximum Interval Value (sec): 1
Simulink Plant Model: fcttransdcmotor
Minimum Interval Value (sec): 0.1

Training Parameters

Training Epochs: 200
Training Function: trainlm
☑ Use Current Weights
☑ Use Validation Data
☑ Use Testing Data

Generate or import data before training the neural network plant.

Fig. 15: Plant identification input data

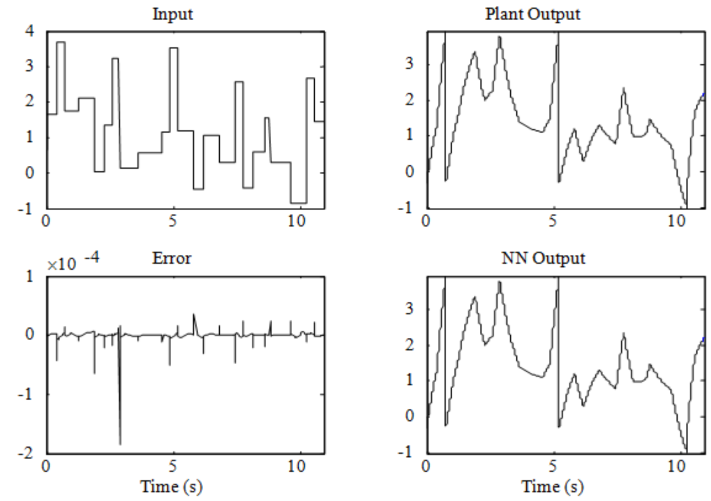


Fig. 16: Training data for NN NARMA L2

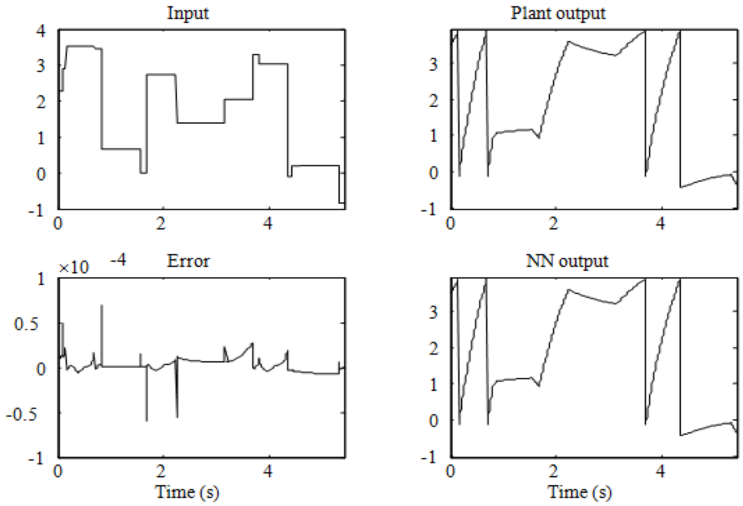


Fig. 17: Testing data for NN NARMA L2

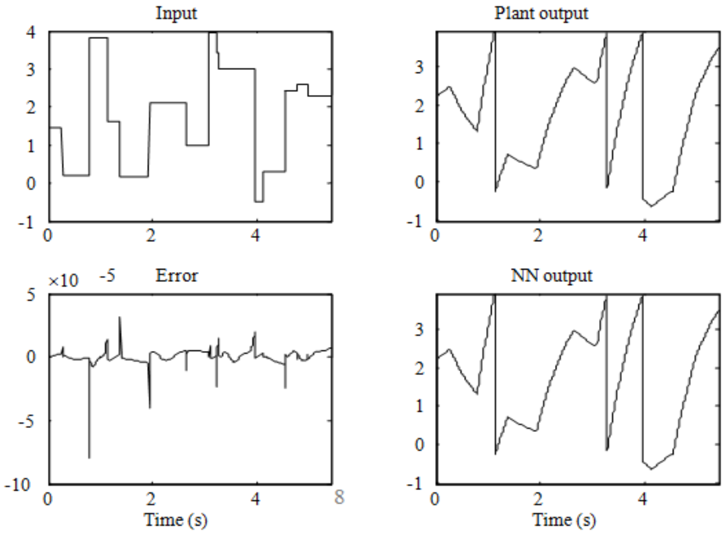


Fig. 18: Validation data for NN NARMA L2

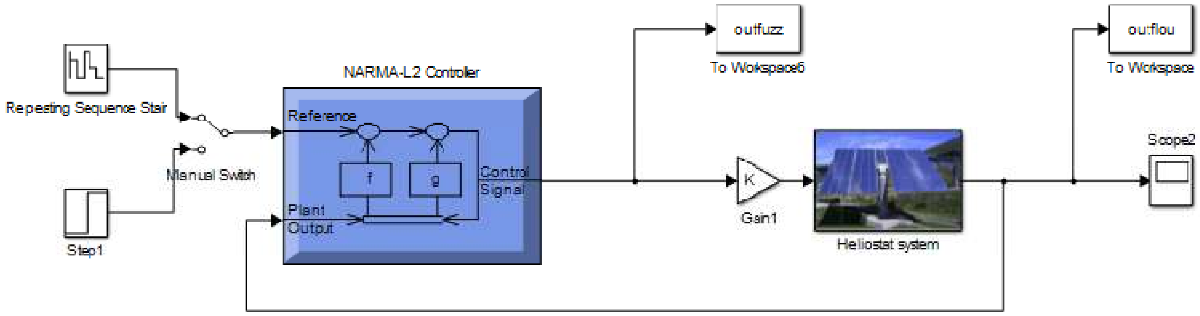


Fig. 19: Simulation diagram for SIMULINK

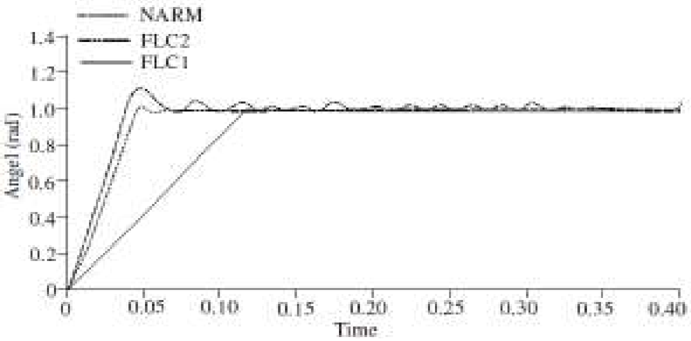


Fig. 20: Comparison between regulators NC, FLC1 and FLC2

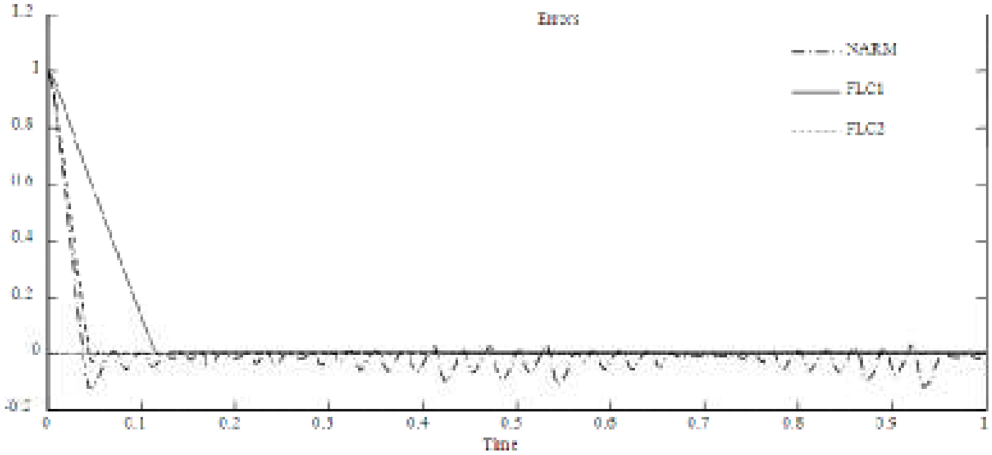


Fig. 21: Errors of three controllers

Training, Testing and validation data of the plant are shown in Fig. 16 to 18.

After the end of the training process controllers, the results must be recorded and the control block is applied to the "real" plant as shown in Fig. 19.

SIMULATION RESULTS

In this part, we are interested to demonstrate the results through the design and simulation of a Neural and fuzzy logic Controller in order to learn a little more about the design of controllers and determine the main characteristics of intelligent control.

The results of the analysis with a comparison of the three controllers are presented in Fig. 20 to 22.

Figure 20 shows the comparaison result between the three controllers for a constant reference and Fig. 21 shows the error between refrence and simulted data. From these figures we can note that FLC2 gives the best results compred to other methods.

Figure 22 shows the results for a given scenario.

From this figure, it is clearly shown that NC give good results. However, it presents some fluctuations that can be harmful for the DC motor. But, we can see that FLC2 give good results without fluctuations.

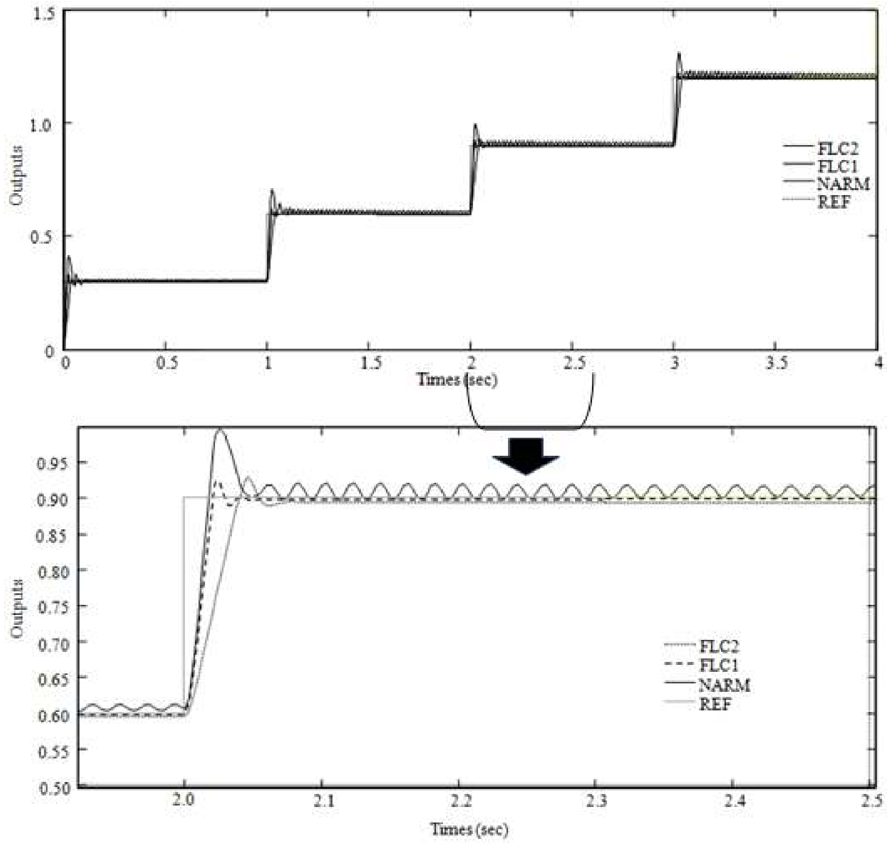


Fig. 22: Comparison between regulators NC, FLC1 and FLC2 at different stages

Table 1: Performance indexes of fuzzy logic and neural controllers

	NARM	FLC1	FLC2
ITAE	0.01676	0.005481	0.0010520
IAE	0.05169	0.064260	0.0253000
ISE	0.01658	0.039790	0.0169000
ITSE	0.001101	0.001152	0.0001925

Table 2: Comparison of response parameters for constant load condition

	NARM	FLC1	FLC2
Rise time	0.0387	0.0445	0.1168
Peak time	0.0461	0.0485	0.1200
Peak value	1.1240	1.0280	1.0070
Overshoot	0.1240	0.0280	0.0070
Setting time	0.1134	0.0629	0.1439

The following performance criterion was used for measuring the effectiveness of the controller in meeting the control objectives (Arrofiq and Saad, 2010). Different speed and load requirements are applied to the system and the performances are taken for analysis and comparisons. The error, *e*, presented in Fig. 21, which calculate the difference between reference and actual value, is commonly characterized into several quantities. Their formulas are presented as follow (Nordin and Arrofiq, 2012):

- The Integral of Squared Error (ISE):

$$ISE = \int_0^\infty e^2(t)dt \tag{1}$$

- The Integral of Absolute Error (IAE):

$$IAE = \int_0^\infty |e(t)|dt \tag{2}$$

- The Integral of Time multiply Squared Error (ITSE):

$$ITSE = \int_0^\infty te^2(t)dt \tag{3}$$

- The Integral of Time multiply Absolute Error (ITAE):

$$ITAE = \int_0^\infty t|e(t)|dt \tag{4}$$

The comparison of neural network controller and Fuzzy logic controller step response specifications are presented in Table 1.

The important parameters such as rise and peak time overshoot, peak value and setting time has been compared for the above controllers and the results are presented in Table 2.

From the simulation results of Table 2, it is concluded that setting time, peak value and overshoot has been improved greatly by using Fuzzy Logic Controller (FLC1 or FLC2). The neural controller is better for a good rise and peak time. Fuzzy logic controller results in less fluctuation at the control system response. The proposed fuzzy logic controller has more advantages.

Similarly, Table 1 shows the performance indexes of fuzzy and neural controllers. The calculated values of the IAE, ISE, ITAE and ITSE shown in Table 1 confirm the simulation results and values of the Table 2.

CONCLUSION

This study shows an application example of artificial intelligence controller based on the neural networks and fuzzy logic models. In the design of intelligent controllers, it is more important to know how the system works as it is evident in the dynamic case, a set of rules can be created in the case of fuzzy logic controller and learning neural network model to achieve the desired results. At first stage, simulation of different configurations of neural and fuzzy controllers has been tested. At second stage, a comparison phase is needed in order to get the appropriate model.

Comparing the two controllers fuzzy with neural. We can note that the second method (FLC2), despite having fewer rules, it shows better response to step changes and very close to zero (Table 1). The neural controller is better according to the ISE and RISE TIME parameter.

The results presented in this study are very encouraging because from the configuration model of neural network and a change in the rules of fuzzy logic; it will be possible to further increase system performance.

REFERENCES

Arrofiq, M. and N. Saad, 2010. Control of induction motor drives using modified-fuzzy logic methods. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp: 612-619.

Besir, D., 2009. Fuzzy neural network IP controller for robust position control of induction motor drive. Expert Syst. Appl., 36: 4528-4534.

Chen, Y.T., A. Kribus, B.H. Lim, C.S. Lim, K.K. Chong and J. Karni, 2004. Comparison of two sun tracking methods in the application of a heliostat field. J. Sol. Energ. Eng., 126(1): 638-44.

Cosme, R.M.G., 2004. Plant Identification and Control using a Neural Controller Based on Reference Model. Retrieved form: www.mathworks.co.jp/matlabcentral/fileexchange/downloads/80933/akamai/PICNNRM.pdf.

Hagan, M.T. and H.B. Demuth, 1999. Neural networks for control. Proceeding of the 1999 American Control Conference. San Diego, CA, 3: 1642-1656.

Hasanien, H.M., S.M. Muyeen and J. Tamura, 2010. Speed control of permanent magnet excitation transverse flux linear motor by using adaptive neuro-fuzzy controller. Energ. Convers. Manage., 51: 2762-2768.

He, Y.L., F.Q. Cui, Z.D. Cheng, Z.Y. Li and W.Q. Tao, 2013. Numerical simulation of solar radiation transmission process for the solar tower power plant: From the heliostat field to the pressurized volumetric receiver. Appl. Therm. Eng., 61: 583-595.

Howard, D. and B. Mark, 2000. Neural Network Toolbox for Use with MATLAB. 1992-2000, Version 4, pp: 6-14.

Ibbini, M.S. and W.S. Zakaria, 1996. Nonlinear control of DC machines. Electr. Mach. Pow. Syst., 24: 21-35.

Kadir, M.Z.A., Y. Rafeeu and N.M. Adam, 2010. Prospective scenarios for the full solar energy development in Malaysia. Renew. Sust. Energ. Rev., 14: 3023-3031.

Kim, K.H., I.C. Baik, S.K. Chung and M.J. Youn, 1997. Robust speed control of brushless DC motor using adaptive input-output linearization technique. IEE P-Elect. Pow. Appl., 144: 469-475.

Mamdani, E. and S. Assilian, 1975. An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man Mach. Stud., 7: 1-13.

Michele, M. and C. Bartolomeo, 2014. Control of a distillation column by type-2 and type-1 fuzzy logic PID controllers. J. Process Contr., 24: 475-484.

Nordin, S. and M. Arrofiq, 2012. A PLC-based modified-fuzzy controller for PWM-driven induction motor drive with constant V/Hz ratio control. Robot. Cim-Int. Manuf., 28: 95-112.

Premkumar, K. and B.V. Manikandan, 2014. Adaptive neuro-fuzzy inference system based speed controller for brushless DC motor. Neurocomputing, 138: 260-270.

Rahma, M.A. and M.A. Hoque, 1998. On-line adaptive neural network based vector control of permanent magnet synchronous motors. IEEE T. Energy Conver., 13: 311-318.

Rigatos, G.G., 2009. Adaptive fuzzy control of DC motors using state and output feedback. Electr. Pow. Syst. Res., 79: 1579-1592.

Rubaai, A., D. Ricketts and M.D. Kankam, 2002. Development and implementation of an adaptive fuzzy-neural-network controller for brushless drives. IEEE T. Ind. Appl., 38(2): 441-447.

Solangi, K.H., M.R. Islam, R. Saidur, N.A. Rahim and H. Fayaz, 2011. A review on global solar energy policy. Renew. Sust. Energ. Rev., 15: 2149-2163.

Vadher, A.H., 2014. Speed control of DC motor with fuzzy control method. Int. J. Innov. Stud., 3(4).