# Study on the Extent of the Impact of Data Set Type on the Performance of ANFIS for Controlling the Speed of DC Motor

**Mohamed Elhaj Ahmed Mohamed & Guo Yanling***

College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin, 150040, PR China
*E-mail: guo.yl@hotmail.com

**Abstract.** This paper introduces an adaptive neuro-fuzzy inference system (ANFIS) for tracking SEDC motor speed in order to optimize the parameters of the transient speed response by finding out the perfect training data provider for the ANFIS. The controller was adjusted using PI, PD and PIPD to generate data sets to configure the ANFIS rules. The performance of the ANFIS controllers using these the different data sets was investigated. The efficiencies of the three controllers were compared to each other, where the PI, PD, and PIPD configurations were replaced by ANFIS to enhance the dynamic action of the controller. The performance of the proposed configurations was tested under different operating situations. Matlab's Simulink toolbox was used to implement the designed controllers. The resultant responses proved that the ANFIS based on the PIPD dataset performed better than the ANFIS based on the PI and PD data sets. Moreover, the suggested controller showed a rapid dynamic response and delivered better performance under various operating conditions.

**Keywords**: *adaptive neuro-fuzzy inference system (ANFIS); data sets; FIS and Matlab Simulink; motor control; SEDC motor.*

## 1 Introduction

The core idea of motor control is to make the motor work reliably and to achieve an ideal operating process. DC motor control means regulating the speed to the desired value to realize all scheduled processes. In many situations, variations in the load can influence the speed. Therefore, the DC motor needs precise control to achieve the desired speed. The portability used in various speed ranges makes the application of an SEDC motor important. Full torque should be obtainable at all speeds. Connecting the armature to a variable voltage source is used to get accurate speed and the speed direction is changed by switching the field polarity [1].

The speed control of DC motors has been broadly applied through the use of conventional control techniques. Nevertheless, these still have some drawbacks. For example, traditional PI, PD, and PID controllers cannot perform the desired speed control, especially under variant loads [2-6]. A control based on fuzzy

Received July 24th, 2018, 1st Revision December 7th, 2018, 2nd Revision December 17th, 2018, Accepted for publication January 11th, 2019.
 ISSN: 2337-5779, DOI: 10.5614/j.eng.technol.sci.2019.51.1.6

logic has been developed to overcome the weakness of conventional PIDs [7], but the efficiency of fuzzy control is limited because it is built on human experiences. This has led several researchers to develop modern methods aimed at improving the performance of the DC motor in order to avoid the shortcomings of conventional PIDs and the limitations of fuzzy control [8]. ANFIS, for instance, is one of the most useful techniques exploited to control DC motor speed. This method was developed by Jang in the 90s of the last century [9]. It is a combination of a fuzzy and a neural network by employing a hybrid learning process and has been applied by many researchers in the field of motor control. ANFIS control was developed specifically to control DC motors and good results have been obtained. It was found that ANFIS has less overshoot and settling time in the speed response and a fast dynamic response compared to fuzzy and conventional PIDs [10].

In addition, a neuro-fuzzy configuration has been recommended to control SEDC motors, where a PI scheme was used to build the training data. The configuration was operated under varying and constant load and tested in a simulation at inconstant speed [11]. The SEDC motor speed was regulated using an ANFIS, but in this system a chopper circuit was utilized. Moreover, the speed response was compared with fuzzy, PI and PID, where the effect of temperature on the speed was considered [12].

A supervisory learning algorithm has been used with ANFIS to track the DC motor speed. However, the results showed high values in the transient response characteristics [13]. In a study by Zhang, once again high overshoot and substantial settling time were observed [14]. The simulation results for an ANFIS controller proposed for SEDC motor speed control showed that its performance with a conventional PI was somewhat acceptable [15]. In [16], fuzzy logic online learning for RBFNN was implemented to control the speed of a DC motor, where the controller showed superior performance compared with conventional PID.

A bat algorithm optimized ANFIS has been designed for controlling a DC motor [17]. The performance of the suggested technique was compared with an ANFIS based genetic algorithm and PSO, Fuzzy-PID, PID based bat algorithm and adaptive FLC. The proposed method showed superior performance in all aspects compared with the other techniques. However, the controller showed different performance between the simulation and experimental verification. Emotional learning algorithms utilizing a proportional-derivative based on ANFIS have been proposed in [18]. However, tuning the PD gain resulted in an observable overshoot and large settling time. In another study, a neuro-fuzzy speed regulator was designed. Its effectiveness in a simulation compared with a conventional speed regulator showed that the steady state was slightly improved

[19]. Likewise, a hybrid control has been established for DC motors, where the presented configuration slowed down the controller response. In addition, the controller exhibited a drawback at variant loads [20]. Another shortcoming was observed when an ANFIS controller was assessed in offline mode [21]. Nevertheless, the performance of ANFIS models in providing correct predictions is comparatively superior to that of ANN [22].

In [23] the genetic algorithm appears again, where it was developed for FLC. However, setting the parameters affects the algorithm, which leads to repeating similar suboptimal solutions. Furthermore, a hybrid GA-PSO algorithm optimized online ANFIS has been developed to control DC motor speed, where the learning parameters were optimized online for different speed torques [24]. In [25], an ANFIS-based composite controller was developed for a static VAR compensator in a power system.

The performance of the introduced method improved the steady-state response but deteriorated the transient response. In [26], the speed of a DC motor was regulated using an ANFIS based on neuro-fuzzy logic algorithms. This controller exhibited larger overshoot and undershoot in its speed response. An ANFIS controller based on PID has been proposed, where the controller showed outstanding performance in all aspects, except for the appearance of overshoot in the speed response of the motor.

In [27], the performance and stability of ANFIS were analyzed for constant and variant speed, sudden load and changes in several motor parameters, i.e. inertia, resistance, inductance and magnetic flux. In a number of studies, algorithms have been developed to cope with the accelerated progression of the motor industry, where [8] and [28] developed a novel bacterial foraging and antlion algorithm to enhance ANFIS performance.

Most of the previous studies compared their proposed designs with one or more conventional schemes. Herein, we present an ANFIS technique based on PI, PD, and PIPD control, where PIPD combines PI and PD training data, implemented through a particular algorithm. This study’s contributions are: firstly, generating three types of training data sets; secondly, comparing the performance of ANFIS between the three models; and thirdly, investigating the response efficiency of the suggested controllers under several operating conditions. This article is organized as follows: Section 2 describes the mathematical modeling of the DC motor. The control techniques are presented in Section 3. The results of the ANFIS methodologies implemented in this work and the discussion are detailed in Section 4, followed by the conclusion in the final section.

## 2 Motor Mathematical Modelling

From Figure 1 and by applying Kirchhoff's voltage law to the circuit, the voltage equation can be formed as follows:

$$E_a(t) = R_a.i_a(t) + L.\frac{di_a(t)}{dt} + e_b(t) \tag{1}$$

where $e_b(t) = K_b.\omega_m(t)$

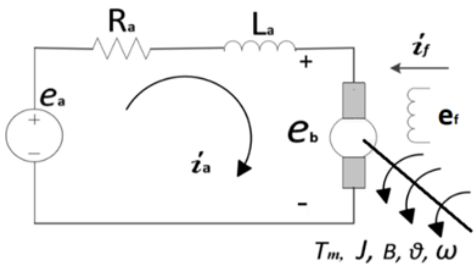


**Figure 1** Electrical circuit of an SEDC motor.

After substitution in Eq. (1), it becomes Eq. (2) as follows:

$$E_a(t) = R_a.i_a(t) + L.\frac{di_a(t)}{dt} + K_b.\omega_m(t) \tag{2}$$

The mechanical part can be formulated as follows:

$$T(t) = J_m.\frac{d\omega_m(t)}{dt} + B_m.\omega_m(t) \tag{3}$$

where $T(t)=K_T.i_a(t)$

After substitution in Eq. (3) we derive Eq. (4):

$$K_T.i_a(t) = J_m.\frac{d\omega_m(t)}{dt} + B_m.\omega_m(t) \tag{4}$$

By applying Laplace transform we get Eqs. (5) and (6):

$$E_a(s) = L.s.I_a(s) + R_a.I_a(s) + K_b.\omega_m(s) \tag{5}$$

$$K_T.i_a(s) = J_m.\omega_m(s).s + B_m.\omega_m(s) \tag{6}$$

To obtain the final motor transfer function formula as in following Eq. (7):

$$\frac{\omega_m(s)}{E_a(s)} = \frac{K_T}{L_a.J_m.s^2 + (R_a.J_m + B_m.L_a).s + (K_T.K_E + R_a.B_m)} \tag{7}$$

Figure 2 shows a block diagram of an SEDC motor. Figure 3 illustrates the model made in Matlab Simulink, while the parameters applied to the model are specified in Table 1.

**Table 1** Motor specifications.

| Parameter | Value |
|---|---|
| Armature inductance $L_a$ | 0.1215H |
| Armature resistance $R_a$ | 11.2 Ω |
| Rotor inertia $J_m$ | 0.02215Kgm |
| Viscous friction coefficient $B_m$ | 0.002953Nms/rad |
| Back EMF constant $K_b$ | 1.28Vs/rad |
| Torque constant $K_T$ | 1.28Nm/A |

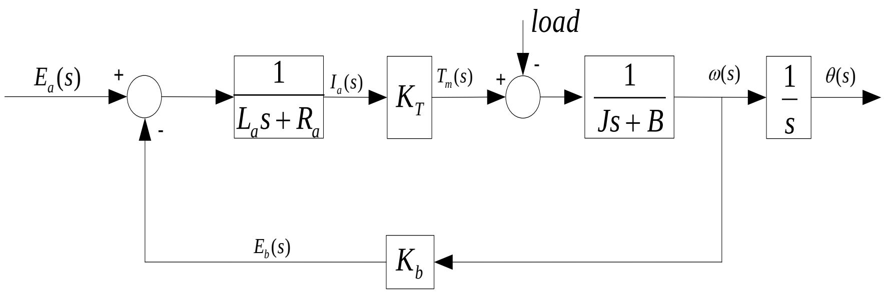


**Figure 2** SEDC motor structure.

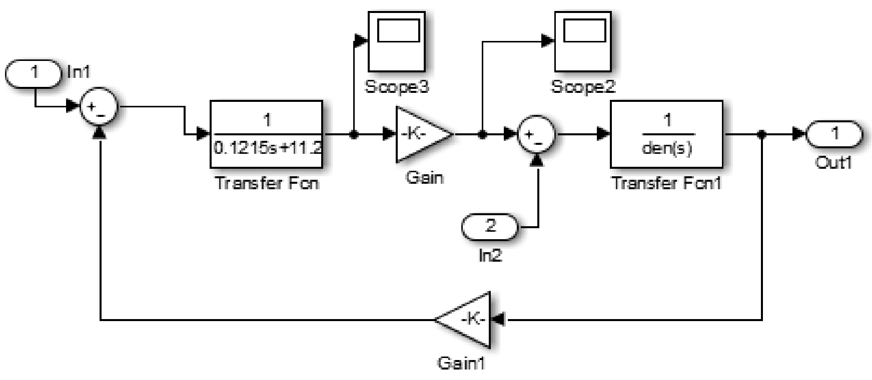


**Figure 3** Simulink scheme of the SEDC motor.

# 3 The Adaptive Neuro-Fuzzy Inference System

## 3.1 Adaptive Neuro-Fuzzy Principle

An artificial neural network is a new control technique that holds numeric entities and manipulates them by figuring out the convergence and divergence between them. Fuzzy logic is a flexible and autonomous methodology that has reasonable interpretation capabilities and can be easily integrated with similar systems. On the other hand, a neural network has superior efficiency with numerical entities. Hence, by incorporating both strategies, a modern method can be acquired.

The new configuration has the characteristics of both systems and results in a significant improvement in modeling, nonlinear mapping, learning and pattern recognition. As for their general structures, fuzzy logic and ANFIS have the same parts except that ANFIS has a neural network portion. This is arranged in four major components: fuzzification, rule base, neural network, and defuzzification, as shown in Figure 4.

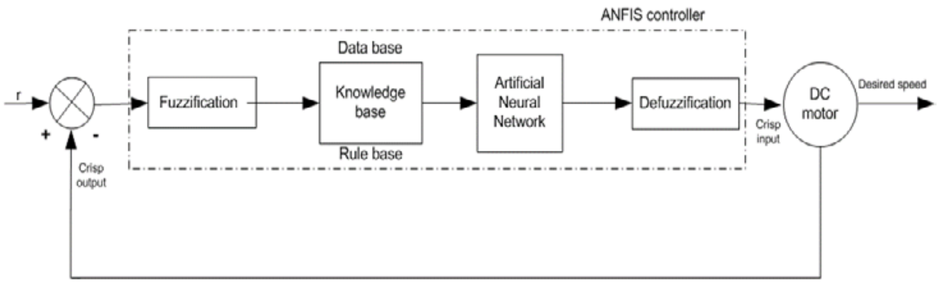


**Figure 4** The general structure of the ANFIS configuration.

The network scheme contains a set of elements structured in five coupled layers. The nodes in layer 1 represent the fuzzy inputs. The weight of the membership functions is checked to select the minimum input values in the second layer. Then, layer 2 sends its output to layer 3, where each neuron is matched with a fuzzy rule and normalized by calculating its weight. Hence, at this level, the number of layers is equal to the number of fuzzy rules.

The fourth layer is called the defuzzification layer and produces the output that results from the fuzzy rule layer. All of these are summed up in layer five to provide crisp values. The general scheme of the neuro-fuzzy network is shown in Figure 5, where the circles and squares represent fixed and adaptive nodes respectively.

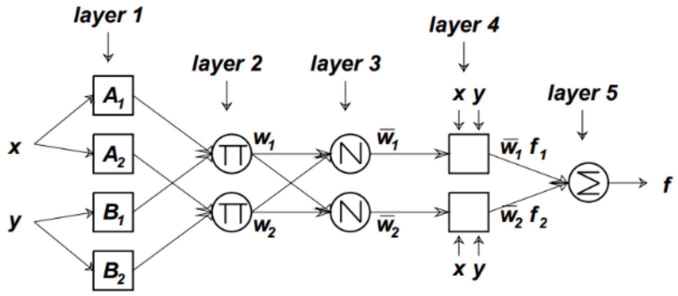


**Figure 5** The general scheme of the neuro-fuzzy network.

## 3.2 The Methodology for Generating Training Data for ANFIS

In this study, three models were built to generate data for training the ANFIS. The first and second were the regular PI and PD while the third one was PIPD, which is a combination of the first two configurations. They were then manipulated by a special algorithm called SADU to ensure that they are able to cover all possible operation conditions (SADU stands for symmetric difference of unions). This algorithm is designed to find out the convergence and divergence between data generated by models. The performance of PI and PD were predictable, but the performance of PIPD could be more efficient. Figure 6 shows the structure of the PIPD model that was used to generate the training data for the ANFIS controller.

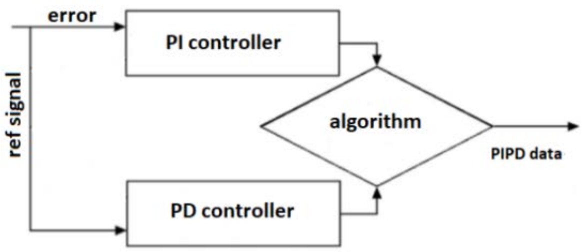


**Figure 6** Architecture of the PIPD model.

SDAU algorithm:

1. Set i, k, j = 0,
2. Define $A_k, B_k, U_k, S_k, R_k$,
3. Repeat ,
4. $U_k = (A_k \cap B_k)$,
5. $S_k = \{ j : (j \in A_k) \oplus (j \in B_k) \}$,
6. $R_k = S_k \cup U_k$,

7. $k \leftarrow k+1$,
8. Stop when the criterion is satisfied,
9. where: Set i, k, j = 0 as index and counter,
10. $A_k, B_k, U_k, S_k, R_k$ are defined as the variables,
11. $U_k = (A_k \cap B_k)$ is the intersection between two inputs that represent the data in $\mathrm{A_k}$ and $\mathrm{B_k}$,
12. $S_k = \{ j : (j \in A_k) \oplus (j \in B_k) \}$ is the symmetric difference between the two data sets,
13. $R_k = S_k \cup U_k$ is the union between the two data sets,
14. $k \leftarrow k+1$ is used to read the data element from the data set,
15. Repeat the condition until the criterion is satisfied.

### 3.3 ANFIS Controller Scheme

The training data from the PI, PD, and PIPD configurations are generated according to observation of the SEDC motor's behavior and then saved in separate files. The model used to generate the data is a predetermined model, which means that the assumed parameters related to the input/output membership function and rules are adjusted to attain optimum performance. The generated data files are uploaded to the fuzzy inference system in order to train it.

The fuzzy system then learns the data and tracked the I/O provided data. The adaptive mechanism must continuously perform online identification of the controlled object during system operation. It is required that the structure of the selected neural network should be suitable for the work characteristics of the adaptive learning mechanism for online learning.

Moreover, the learning speed should be increased. In case the network structure is more complicated, the number of weights that need to be adjusted is higher. This inevitably affects the learning speed of the adaptive mechanism. It also makes it impossible to properly track changes of the controlled object and the dynamic learning theory of the multi-layer neural network is not perfect enough. Based on comprehensive consideration of the above factors, the speed controller in this study used neuron network dynamic learning to achieve the adaptive mechanism as explained in Figure 7.

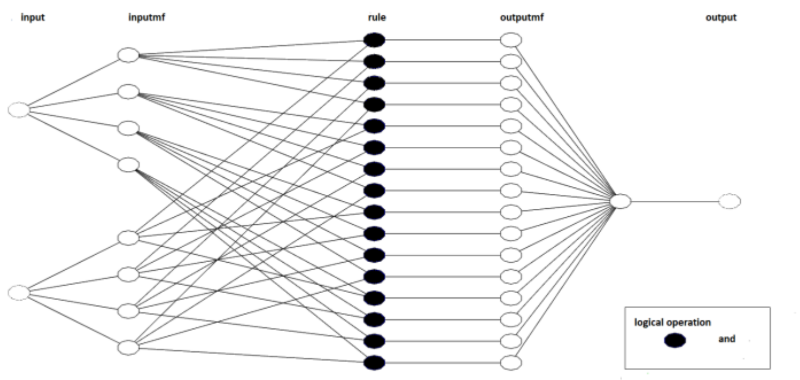


**Figure 7** Architecture of the ANFIS network.

The PI, PD, and PIPD controllers were built and correctly adjusted. The generated data were investigated several times in order to ensure that they were able to cover all situations. In this study, the data were minimized as much as possible to increase the dynamic response speed. Moreover, they were rearranged in a usable form for training in the ANFIS. Subsequently, the ANFIS was utilized to train the data set. The zero-error criterion was used to modify the membership functions. The model was validated to ensure that the FIS model was successful in predicting the values of the equivalent data set output; the models' efficiency and capability were verified; and the tested data were able to cover all different possibilities of load variation. The system was repeatedly tested by implementing different load signals and values. This resulted in the membership functions, method optimization and several modifications of the error tolerance to obtain an ideal response. If the training data prepared for the ANFIS completely represent the features of the optimal response, this kind of modeling will work admirably. However, if the training data are prepared via noisy measurements and cannot represent all features of the data that will be presented to the ANFIS, then validating the model is helpful. As an example, the data generated using the PIPD scheme to perform ANFIS are shown in Figures 8, 9 and 10 respectively.

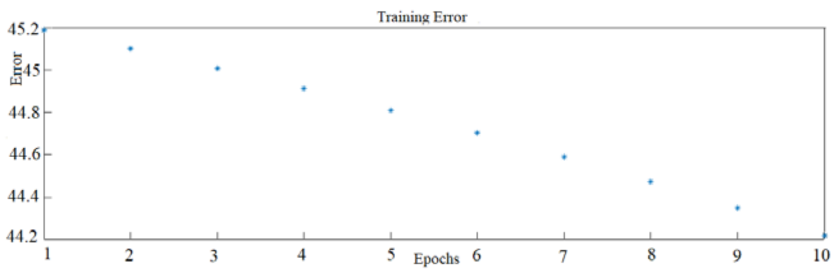


**Figure 8** PIPD training errors for ANFIS.

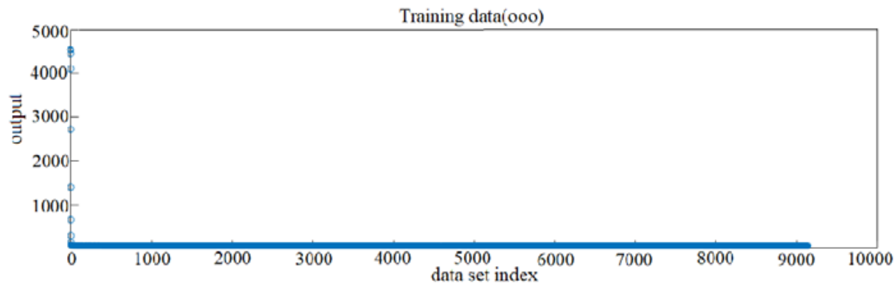


**Figure 9** PIPD training data for ANFIS.

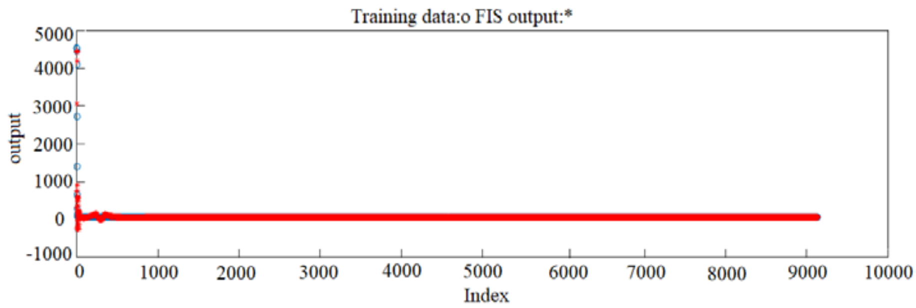


**Figure 10** PIPD tested data for ANFIS.

For all the introduced controllers, a fuzzy first-order Takagi-Sugeno model was utilized to perform the configuration of the ANFIS. It had two inputs, namely error E and its changing rate EC, and one output to represent the control signal to the motor. The rules were defined as follows:

Rule 1: If $X$ is $A_1$ and $y$ is $B_1$, then

$$f_1 = p_1 x + q_1 y + r_1;$$

Rule 2: If $X$ is $A_2$ and $y$ is $B_2$, then

$$f_2 = \mathrm{p}_2 x + q_2 y + r_2;$$

Since A and B are the fuzzy antecedent sets, $f\,(x.y)$ is a consequent crisp function. In this configuration, the ANFIS controller has two inputs, each consisting of four triangle membership functions and one linear output type. The system has 16 possible rules, with zero-error tolerance and 10 epochs.

The ANFIS structure was tuned automatically by a back propagation optimization algorithm for training the FIS because it is flexible and showed perfect performance with load variation, while the hybrid optimization method was found to be unsuitable for changes in load. Figure 11 shows a schematic diagram of the ANFIS Simulink model for controlling SEDC motor speed, where the reference and real speed difference are input into the ANFIS configuration, while the resulting output is the voltage to the motor.

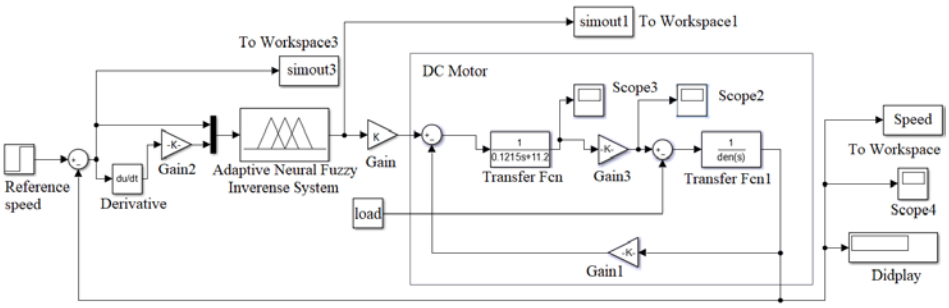


**Figure 11** ANFIS Simulink model scheme for controlling SEDC motor speed.

## 4 Results and Discussion

The system was tested in several stages. In the first stage, the motor response was tested without controller for the purpose of clarifying the overall impact of the control. As expected, the response showed a sizeable steady-state error as shown in Figure 11. Secondly, it was operated with a fixed load equal to 10% of the input signal. The response was very far removed from the optimal response, as illustrated in Figures 12 and 13.

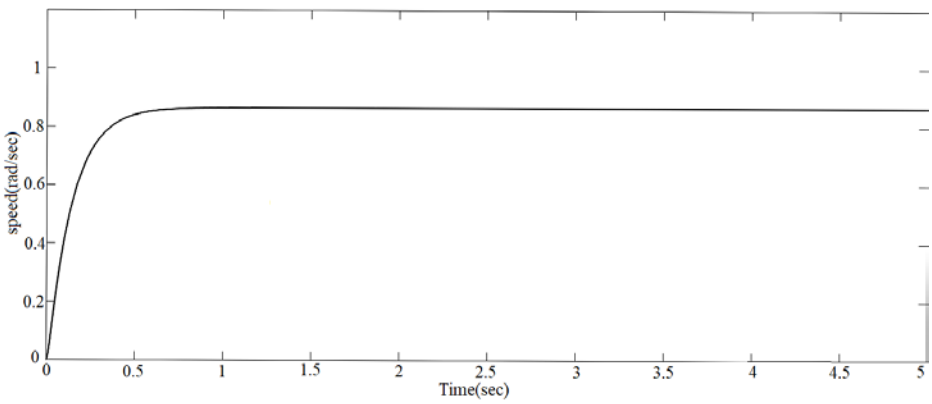


**Figure 12** Motor response without control and load.

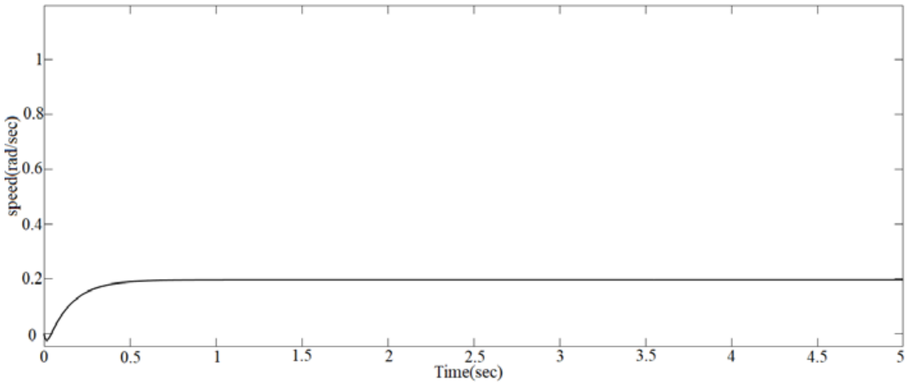


**Figure 13** Motor response without control and with constant 1/10 of referenced speed.

In the second, and for the purpose of demonstrating the disparity in the performance of the simulation system tests, the ANFIS controller operated based on the data generated by the PI, PD, and PIPD models. First, the speed response performance was accomplished without load. For the ANFIS-based PI data controller, the speed response showed a reasonable response performance with a slight overshoot and an inconsiderable oscillation before reaching a stable state, as shown in Figure 14. As for the ANFIS-based PD data controller, its response showed a distinct overshoot. Moreover, it also showed a small oscillation and more rapidly reached a steady state compared with the ANFIS-based PI data controller, as can be seen in Figure 15. Figure 16 illustrates the speed response produced by the ANFIS-based PIPD data controller, which showed optimal performance and superiority in all aspects.

The second test was done by applying a constant external load to the motor in order to determine the ability of the controller for disturbance avoidance. The ANFIS-based PI data controller got stuck in an oscillating state, thus causing an inadmissible instability, as can be seen in Figure 17. This obviously proves that the controller failed to bear a constant load under the same circumstances, other than the other two controllers. PD and PIPD demonstrated their ability in dealing with an unvarying load without any change in their responses, as illustrated in Figures 18 and 19. In the third test, the constant load was increased tenfold. For the ANFIS-based PI data and the ANFIS-based PD data, this resulted in a massive overshoot in the speed response. However, so far the ANFIS-based PIPD data maintained their efficiency without any change. Besides, their response was not affected even if the load was increased ten- or hundredfold. For the ANFIS-based PIPD data controller response, the overshoot was so small that it was difficult to measure. As for the rise time and settling time, the controller showed exceptional performance and a very speedy dynamic response with no oscillation. Figures 20-22 display the three controllers' speed responses.

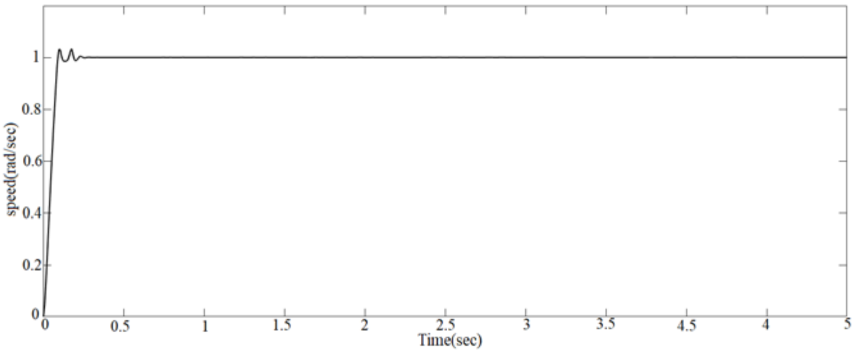


**Figure 14** Speed response of ANFIS-based PI training data without load.

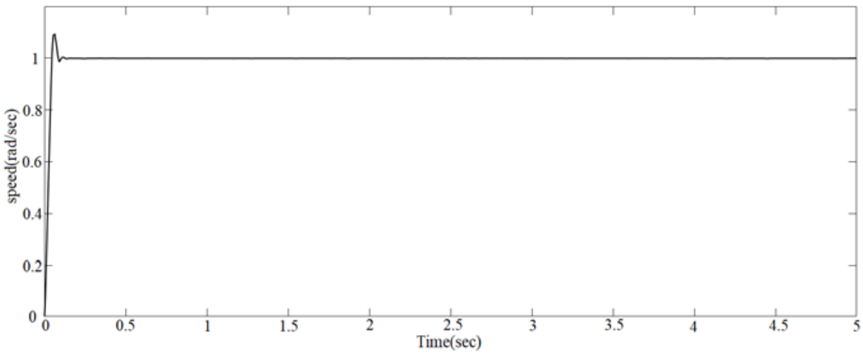


**Figure 15** Speed response of ANFIS-based PD training data without load.

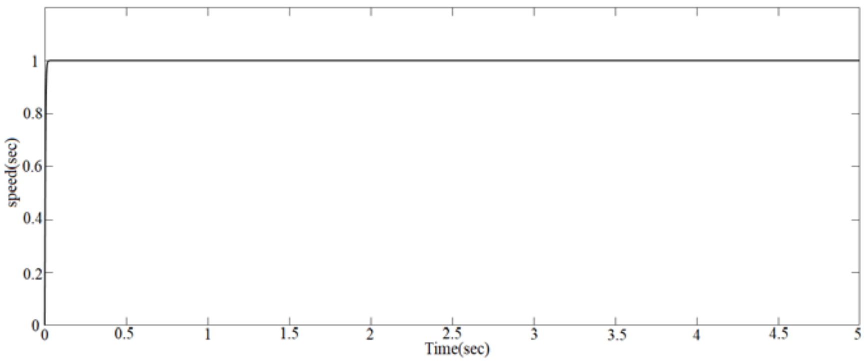


**Figure 16** Speed response of ANFIS-based PIPD training data without load.

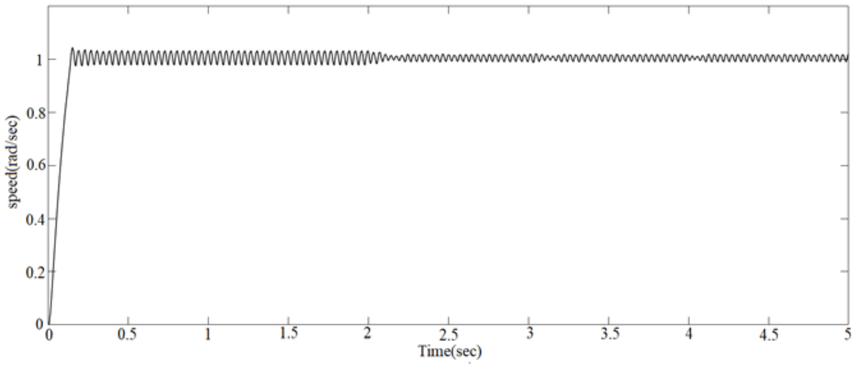


**Figure 17** Speed response of ANFIS-based PI training data with a constant load.

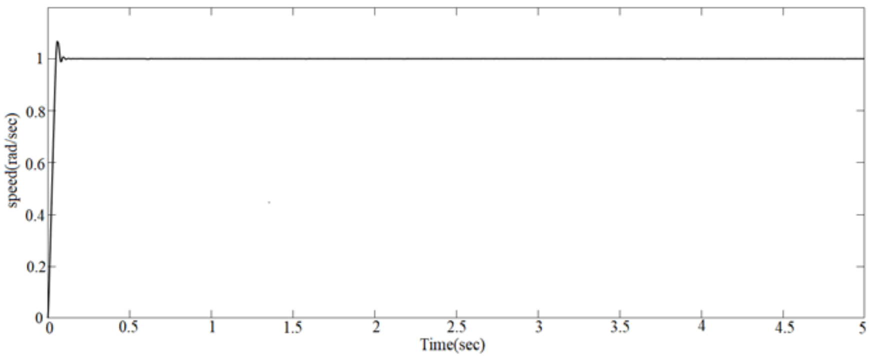


**Figure 18** Speed response of ANFIS-based PD training data with a constant load.

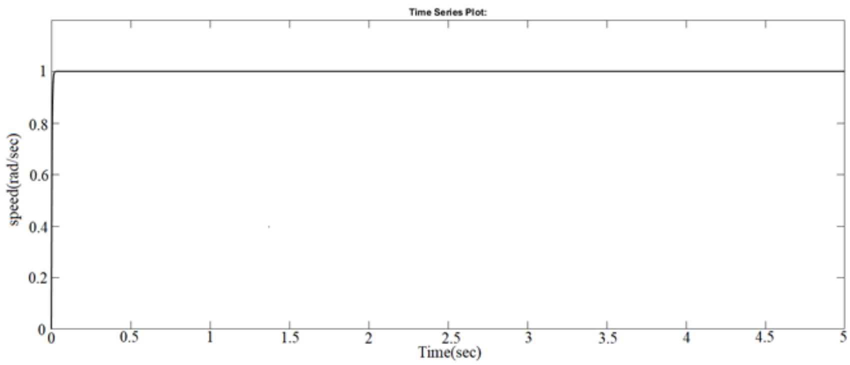


**Figure 19** Speed response of ANFIS based on PIPD training data with a constant load.

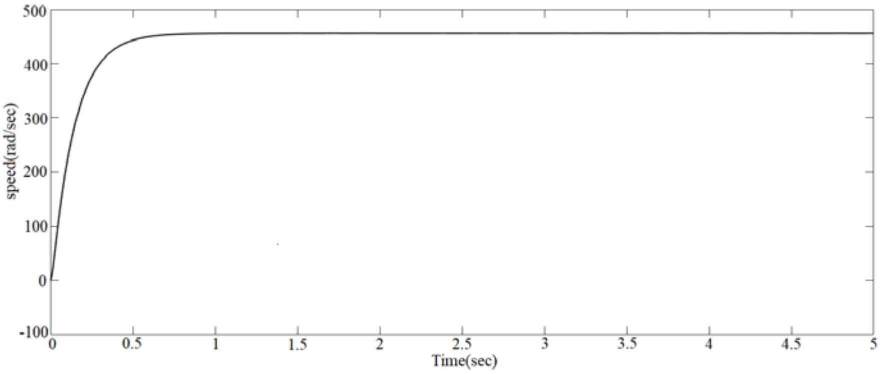


**Figure 20** Speed response of ANFIS-based PI training data with 10 times constant load.

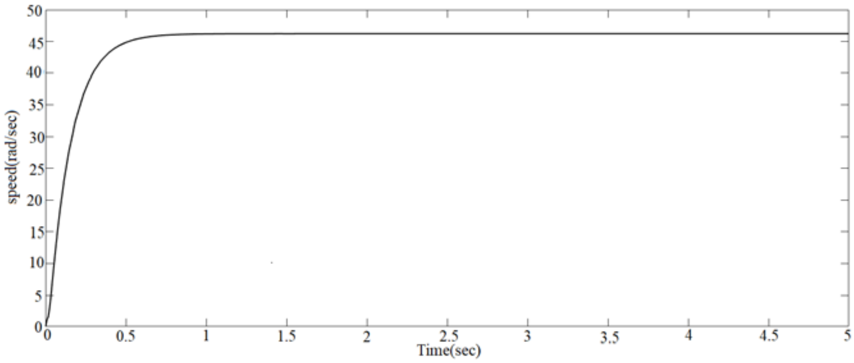


**Figure 21** Speed response of ANFIS-based PD training data with 10 times constant load.

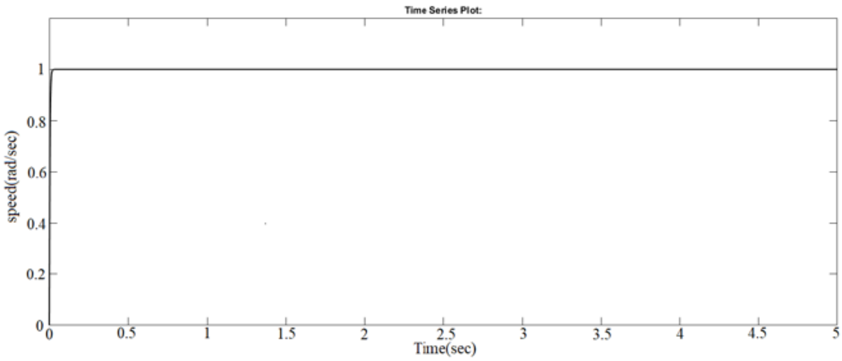


**Figure 22** Speed response of ANFIS-based PIPD training data with 10 and 100 times constant load.

Furthermore, for further confirmation the system was tested under sudden load. First, the system was operated under a slight sudden load. For the ANFIS-based PI data controller, apparently, the slight sudden load had a large effect on the speed response, as can be seen in Figure 23. As a result, the motor speed oscillated during the load period and only reached stable state after the load was removed. Meanwhile, for the ANFIS-based PD data, there was a small change observed in the speed response to the sudden load, as shown in Figure 24. With respect to the ANFIS-based PIPD data no difference was seen, as is evident from Figure 25, where the controller succeeded to absorb the load successfully. For more confirmation, the amplitude of the sudden load was increased ten- and hundredfold. The ANFIS-based PIPD data controller displayed outstanding performance for the sudden heavy load, as shown in Figure 26. In contrast, for the other two controllers the increase of the sudden load affected the system's stability.

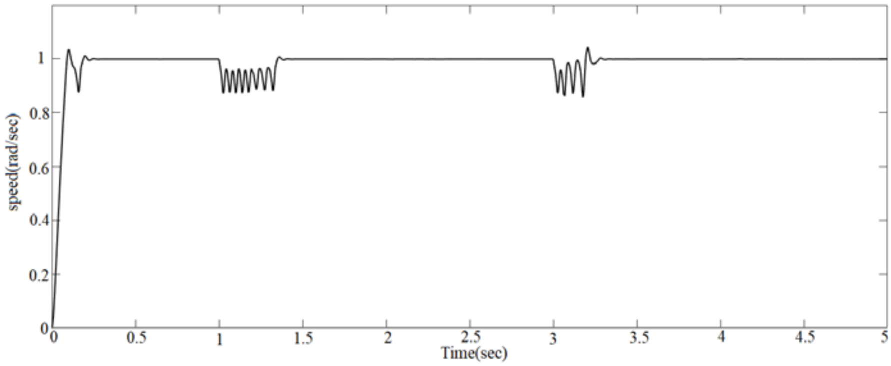


**Figure 23** Speed response of ANFIS-based PI training data with a sudden load.

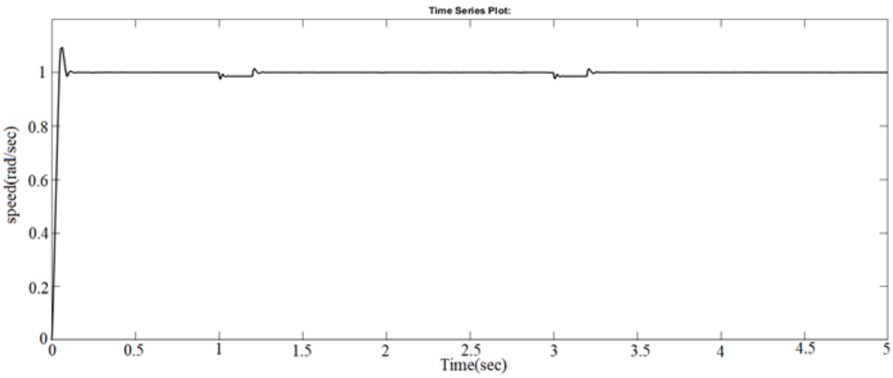


**Figure 24** Speed response of ANFIS-based PD training data with a sudden load.

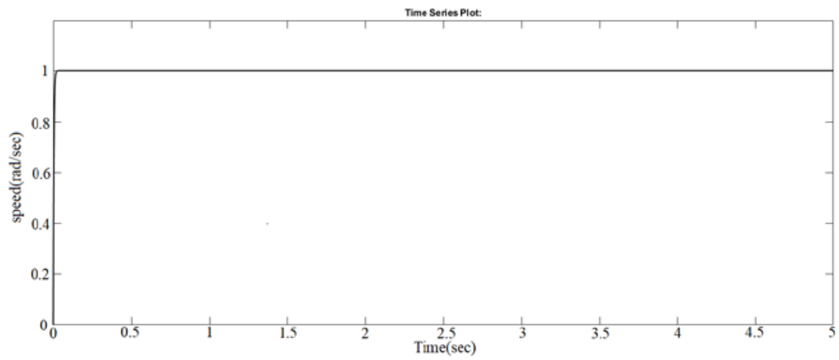


**Figure 25** Speed response of ANFIS-based PIPD training data with a sudden load.

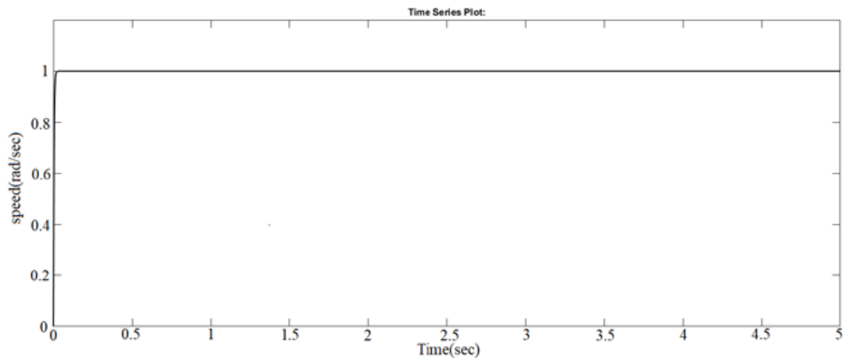


**Figure 26** Speed response of ANFIS-based PIPD training data with a sudden load increased 100 times.

## 5 Conclusion

In this study, the performance of ANFIS was assessed according to the feed-forward data type used to set the rules for the speed controller of an SEDC motor. PI, PD, and PIPD models were used to generate the data for the ANFIS. The created data were integrated into the ANFIS configuration and operated under different conditions to evaluate the transient response parameters while absorbing constant and sudden load, and steady-state error. In general, the designed ANFIS configuration has several advantages, for instance, the simplicity of its structure and learning susceptibility. Implementation of its rules is relatively rapid and easy compared to traditional methods. In addition to these features, the ANFIS-based PIPD data also showed high robustness under changing load and superior performance for speed control. Added to this, it showed low oscillation and more accuracy in its speed response. The controller also proved to have a perfect tracking response for the desired speed without overshoot and short settling time. The Matlab simulation results proved that the ANFIS-based PIPD data provided good performance since they passed all different tests conditions and showed high efficiency. Moreover, the controller showed a distinct performance increase compared to the ANFIS-based PI or PD training data and the controllers previously mentioned in the literature review.

## References

[1] Azman, M.A.H., Aris, J.M., Hussain, Z., Samat, A.A.A. & Nazelan, A.M., *A Comparative Study of Fuzzy Logic Controller and Artificial Neural Network in Speed Control of Separately Excited DC Motor*, 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 336-341, 24-26 Nov., 2017.

[2] Premkumar, K. & Manikandan, B., *Bat Algorithm Optimized Fuzzy PD Based Speed Controller for Brushless Direct Current Motor*, Engineering Science and Technology, an International Journal, **19**(2), pp. 818-840, 2016.

[3] Meshram, P. & Kanojiya, R.G., *Tuning of PID Controller Using Ziegler-Nichols Method for Speed Control of DC Motor, Advances in Engineering*, International Conference on Science and Management (ICAESM), pp. 117-122, 2012.

[4] Mondal, R., Mukhopadhyay, A. & Basak, D., *Embedded System of DC Motor Closed Loop Speed Control Based on 8051 Microcontroller,* Procedia Technology, **10**, pp. 840-848, 2013.

[5] Al-Mashakbeh, A.S.O., *Proportional Integral and Derivative Control of Brushless DC Motor*, European Journal of Scientific Research, **35**(2), pp. 198-203, 2009.

[6] Basilio, J. & Matos, S., *Design of PI and PID Controllers with Transient Performance Specification*, IEEE Transactions on Education, **45**(4), pp. 364-370, 2002.

[7] Prabu, M.J., Poongodi, P. & Premkumar, K., *Fuzzy Supervised Online Coactive Neuro-Fuzzy Inference System-Based Rotor Position Control of Brushless DC Motor*, IET Power Electronics, **9**(11), pp. 2229-2239, 2016.

[8] Sivarani, T., Jawhar, S.J. & Kumar, C.A., *Novel Bacterial Foraging-Based ANFIS for Speed Control of Matrix Converter-Fed Industrial BLDC Motors Operated Under Low Speed and High Torque*, Neural Computing and Applications, **29**(12), pp. 1411-1434, 2018.

[9] Jang, J. S., *ANFIS: Adaptive-Network-Based Fuzzy Inference System*, IEEE Transactions on Systems, Man, and Cybernetics, **23**(3), pp. 665-685, 1993.

[10] Chaudhary, H., Khatoon, S. & Singh, R., *ANFIS Based Speed Control of DC Motor*, Second International Innovative Applications of

Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity (CIPECH), pp. 63-67, 2016.

[11] Pavankumar, S., Krishnaveni, S., Venugopal, Y. & Babu, Y.K., *A Neuro-Fuzzy Based Speed Control of Separately Excited DC Motor*, Computational Intelligence and Communication Networks (CICN), International Conference on, pp. 93-98, 2010.

[12] Omar, B., Haikal, A. & Areed, F., *Design Adaptive Neuro-Fuzzy Speed Controller for An Electro-Mechanical System*, Ain Shams Engineering Journal, **2**(2), pp. 99-107, 2011.

[13] Mosavi, M.R., Rahmati, A. & Khoshsaadat, A., *Design of Efficient Adaptive Neuro-Fuzzy Controller Based on Supervisory Learning Capable for Speed and Torque Control of BLDC Motor*, PRZEGLĄD ELEKTROTECHNICZNY (Electrical Review), R, **88**, pp. 238-246, 2012.

[14] Zhang, Q.C. & Jiang, M., *Adaptive Neuro-Fuzzy Control of BLDCM Based on Back-EMF*, Journal of Computer Information Systems, **7**(12), pp. 4560-4567, 2011.

[15] Tripura, P. & Babu, Y., *Intelligent Speed Control of DC Motor Using ANFIS*, Journal of Intelligent & Fuzzy Systems, **26**(1), pp. 223-227, 2014.

[16] Premkumar, K. & Manikandan, B., *Online Fuzzy Supervised Learning of Radial Basis Function Neural Network Based Speed Controller for Brushless DC Motor*, in Power Electronics and Renewable Energy Systems: Springer, pp. 1397-1405, 2015.

[17] Premkumar, K. & Manikandan, B., *Speed control of Brushless DC motor Using bat Algorithm Optimized Adaptive Neuro-Fuzzy Inference System*, Applied Soft Computing, **32**, pp. 403-419, 2015.

[18] Niasar, A., Vahedi, A. & Moghbelli, H., *Speed Control of a Brushless DC Motor Drive via Adaptive Neuro-Fuzzy Controller Based on Emotional Learning Algorithm*, Proceedings of the Eighth International Conference on Electrical Machines and Systems (ICEMS), **1**, pp. 230-234, 2005.

[19] Premkumar, K. & Manikandan, B., *Fuzzy PID Supervised Online ANFIS Based Speed Controller for Brushless DC Motor*, Neurocomputing, **157**, pp. 76-90, 2015.

[20] Mansouri, M., Kaboli, S.H.A., Ahmadian, J. & Selvaraj, J., *A Hybrid Neuro-Fuzzy PI Speed Controller for BLDC Enriched with an Integral Steady State Error Eliminator*, 2012 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 234-237, 2012.

[21] Premkumar, K. & Manikandan, B., *Adaptive Neuro-Fuzzy Inference System Based Speed Controller for Brushless DC Motor*, Neurocomputing, **138**, pp. 260-270, 2014.

[22] Khoshnevisan, B., Rafiee, S., Omid, M. & Mousazadeh, H., *Development of an Intelligent System Based on ANFIS for Predicting Wheat Grain Yield on the Basis of Energy Inputs*, Information Processing in Agriculture, **1**(1), pp. 14-22, 2014.

[23] Shieh, C.S., *Fuzzy PWM Based on Genetic Algorithm For Battery Charging*, Applied Soft Computing, **21**, pp. 607-616, 2014.

[24] Premkumar, K. & Manikandan, B., *GA-PSO Optimized Online ANFIS Based Speed Controller for Brushless DC Motor*, Journal of Intelligent & Fuzzy Systems, **28**(6), pp. 2839-2850, 2015.

[25] Ginarsa, I., Soeprijanto, A. & Purnomo, M., *Controlling Chaos and Voltage Collapse Using an ANFIS-Based Composite Controller-Static Var Compensator In Power Systems*, International Journal of Electrical Power & Energy Systems, **46**, pp. 79-88, 2013.

[26] Al-Mashhadany, Y.I., *Modeling And Simulation of Adaptive Neuro-Fuzzy Controller for Chopper-Fed Dc Motor Drive*, 2011 IEEE Applied Power Electronics Colloquium (IAPEC), pp. 110-115, 2011.

[27] Premkumar, K. & Manikandan, B., *Stability and Performance Analysis of ANFIS Tuned PID Based Speed Controller for Brushless DC Motor*, Current Signal Transduction Therapy, **13**(1), pp. 19-30, 2018.

[28] Premkumar, K., Manikandan, B.V. & Kumar, C.A., *Antlion Algorithm Optimized Fuzzy PID Supervised On-line Recurrent Fuzzy Neural Network Based Controller for Brushless DC Motor*, Electric Power Components and Systems, **45**(20), pp. 2304-2317, 2017.