International Journal of Innovative Computing, Information and Control
Volume **15**, Number **3**, June **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **939–954**

HYBRID PARTICLE SWARM OPTIMIZATION AND RECURSIVE LEAST SQUARE ESTIMATION BASED ANFIS MULTIOUTPUT FOR BLDC MOTOR SPEED CONTROLLER

Heri Suryoatmojo, Mohamad Ridwan, Dedet Candra Riawan
Eko Setijadi and Ronny Mardiyanto

Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Kampus ITS, Sukolilo, Surabaya 60111, Indonesia
suryomgt@gmail.com

Received August 2018; revised December 2018

Abstract. *Brushless Direct Current (BLDC) motor speed control has been widely developed to obtain high performance in its operation. However, most of the controllers still used conventional controllers that have some drawbacks whenever operated for the different BLDC motor. This paper proposes BLDC speed controller by implementing multi-output Adaptive Neuro Fuzzy Inference System (ANFIS). ANFIS algorithm is able to control the speed of the BLDC motor according to the desired reference value. The average of steady state error achieved using ANFIS is 0.1% and the rise time is 2.7437 s when the reference speed is 4000 rpm. ANFIS learning process uses hybrid Particle Swarm Optimization (PSO) and Recursive Least Square Estimation (RLSE) methods supervised by Fuzzy-PID. PSO and RLSE can train the multi-output ANFIS data very well. The best training data is achieved when the value of λ is 1 with RMSE error of 0.05364. The execution time of ANFIS algorithm on microcontroller is 96 μs.*

Keywords: Brushless direct current motor, Control system, Fuzzy logic, Adaptive neuro fuzzy inference system, Particle swarm optimization, Recursive least square estimation, Electric vehicle

1. **Introduction.** Brushless Direct Current (BLDC) motors have an important role in Electric Vehicle (EV) development. BLDC motors are often used as the prime mover of EVs since BLDC motors are the most efficient machine among other types of electric machines [1,2]. Moreover, BLDC motor has high efficiency, high torque, wide speed range, and low maintenance, since it has no brushes [3-5].

BLDC motors are similar to Direct Current (DC) motors with no brushes [5], consisting of two main parts: electrical and mechanical parts. BLDC motors have three windings, of which it has to be energized with a proper sequence. BLDC motors are usually equipped with Hall Effect sensors to detect rotor position, to measure the speed, and to synchronize the inverter, so that it can energize BLDC motor windings with the proper sequence.

BLDC motors should be controlled to get a desired value, such as speed, torque, and position. In EV application, torque is an important parameter to obtain, so the power needed to drive the vehicle is met. BLDC motor speed, however, also needs to be controlled to get a desired acceleration or to minimize slip in a traction control. Speed control of BLDC motor has been widely developed from using a conventional method, such as Proportional, Integral, and Derivative (PID) controller, until using advanced methods, such as fuzzy logic or neural network. PID controller is the simplest control method with simple structure and easy to implement, but it can stand for a linear system only

DOI: 10.24507/ijicic.15.03.939

[7-10]. Fuzzy Logic Controller (FLC) is one of advanced control methods which is able to overcome uncertainty and nonlinear parameters. Nevertheless, FLC is still not adaptive and has no learning mechanism, so the difficulties come up in redesigning and adapting when new rules are applied [11-15]. Neural Network (NN) is a control method which has a learning mechanism and can overcome uncertainty and nonlinear parameters. In practice, however, NN needs more storage capacity, to store weights and neuron parameters. Adaptive Neuro Fuzzy Inference System (ANFIS) is a fuzzy system which has a learning mechanism, so it can overcome the drawbacks coming from both FLC and NN [16].

Gradient descent method is usually used for learning algorithm in NN or ANFIS. The main limitation of back propagation with gradient descent is guaranteed to find local optimum instead of global optimum value [17]. In this study, Particle Swarm Optimization (PSO) is proposed for ANFIS learning algorithm. PSO is a technique to find a solution in an optimum area through interactions of individual in a population. Many studies report that PSO can overcome the drawback of getting stuck at local optimum value [18-20]. In the proposed method, PSO is combined with Recursive Least Square Estimation (RLSE) algorithm to train premise and consequent parameters of ANFIS.

Usually, to get the best performance of control system, a hybrid method is used, such as, Fuzzy-PID or NN PID, since PID control is widely used in industry due to its best performance and simplicity. It, however, can be difficult in practice to tune the parameters. In the study, ANFIS-PID based on a hybrid PSO-RLSE algorithm is proposed for the designed control system to control a BLDC motor speed. It stimulates to develop tuning method of PID parameters in order to be incorporated to a hardware implementation [16]. The output of ANFIS is designed for three output values which are used for tuning PID parameters adaptively. ANFIS multioutput structure is proposed to minimize a needed memory of parameters which will be stored in a hardware system.

The paper is organized as follows. Section 2 describes the mathematical model of BLDC motor. Section 3 explains about fuzzy design and the proposed method. Section 4 shows the simulation result, both ANFIS training and BLDC motor speed control results. The conclusion would be described in Section 5.

2. **Mathematical Model of BLDC Motor.** BLDC motor consists of two components: electrical and mechanical. In principle, the electrical parts of BLDC motor are similar to DC motor but with three phase windings. Permanent magnets are as components to generate the magnetic field on the rotor side of a BLDC motor. Figure 1 shows modeling of BLDC motor fed mechanical load [21-25].

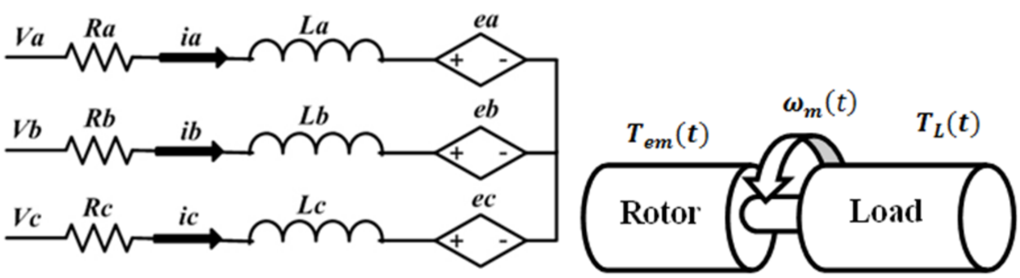

FIGURE 1. Electrical and mechanical components of BLDC motor

From Figure 1, the equation of three phase windings in the electrical side of BLDC motor can be expressed as:

$$v_a(t) = R_a i_a(t) + L_a \frac{di_a}{dt}(t) + e_a(t) \tag{1}$$

$$v_b(t) = R_b i_b(t) + L_b \frac{di_b}{dt}(t) + e_b(t) \tag{2}$$

$$v_c(t) = R_c i_c(t) + L_c \frac{di_c}{dt}(t) + e_c(t) \tag{3}$$

where:
$v_a(t)$, $v_b(t)$, $v_c(t)$ = instantaneous voltage of each phase.
$i_a(t)$, $i_b(t)$, $i_c(t)$ = instantaneous current of each phase.
$e_a(t)$, $e_b(t)$, $e_c(t)$ = instantaneous Back-Electromotive Force (BEMF) of each phase.
R_a, R_b, R_c = stator winding resistance of each phase.
L_a, L_b, L_c = stator winding inductance of each phase.

In fact, in every sequence of energizing windings of BLDC motor, there are only two windings which are energized. According to the condition, by considering that the resistances and inductances are constant and equal, so, it can be expressed as:

$$v_{ab} = R(i_a - i_b) + L\frac{d(i_a - i_b)}{dt} + (e_a - e_b) \tag{4}$$

$$v_{bc} = R(i_b - i_c) + L\frac{d(i_b - i_c)}{dt} + (e_b - e_c) \tag{5}$$

$$v_{ca} = R(i_c - i_a) + L\frac{d(i_c - i_a)}{dt} + (e_c - e_a) \tag{6}$$

where, the magnitude of v_{ab}, v_{bc}, and v_{ca} equals supply voltage used on the system (V_{dc}). On the other hand, the mechanical component is expressed as:

$$T_{em}(t) = J\frac{d\omega_m(t)}{dt} + B\omega_m(t) + T_L(t) \tag{7}$$

where,
$T_{em}(t)$ = total electromotive torque
$\omega_m(t)$ = rotor angular velocity
B = viscous friction
J = moment of inertia
$T_L(t)$ = load torque

According to the equation of BLDC motor, then it is modeled and simulated. The speed of BLDC motor is controlled using Adaptive Neuro Fuzzy Inference System (ANFIS) based on Particle Swarm Optimization (PSO) and Recursive Least Square Estimation (RLSE) as forward and backward learning method.

3. Adaptive Neuro Fuzzy Inference System Based Speed Controller.

Adaptive Neuro Fuzzy Inference System, or usually abbreviated as ANFIS, is a fuzzy inference system structure which has a learning mechanism as owned by neural network structure. ANFIS has been proven to overcome the nonlinearities in a system, as fuzzy does. Fuzzy rule used in ANFIS is a Takagi-Sugeno model which uses "if-then" rule method. For example, there are two inputs x and y, and an output f, so the rules can be defined as:

Rule 1: If x is A_1 and y is B_1 then $f_1 = p_1x + q_1x + r_1$
Rule 2: If x is A_2 and y is B_2 then $f_2 = p_2x + q_2x + r_2$

A_1, A_2 and B_1, B_2 are the membership functions of each input x and y (premise), while p_1, q_1, r_1 and p_2, q_2, r_2 are the linear parameters (consequent) of Takagi-Sugeno fuzzy inference model. Figure 2 shows a reasoning mechanism and architecture of general ANFIS.

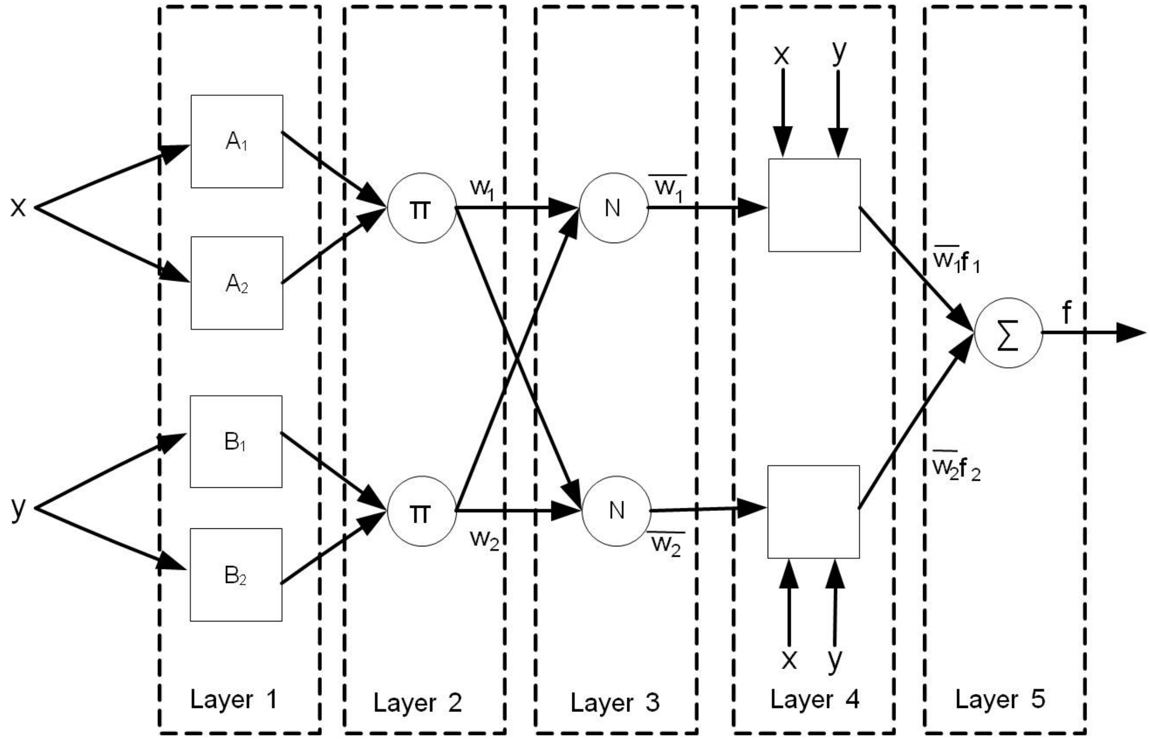

FIGURE 2. General ANFIS structure

- Layer 1

Each of the i^{th} node in this layer is the result of a function as shown as:

$$O_i^1 = \mu_{A_i}(x) \tag{8}$$

x is an input to node i. A_i is a linguistic variable related to the node, while μ_{A_i} is a membership function of A_i. $\mu_{A_i}(x)$ is usually chosen as a Gaussian or Bell-shape function which is shown in Equations (9) and (10) respectively.

$$\mu_{A_i}(x) = \exp\left\{-\frac{1}{2}\left(\frac{x - c_i}{a_i}\right)^2\right\} \tag{9}$$

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}} \tag{10}$$

where x is the input while a_i, b_i and c_i are the premise parametes.

- Layer 2

Each node in this layer is a fixed node resulting in weight w_i of a fuzzy rule. The output of this node is resulted by multiplying all inputs to this node. This is shown as equation below.

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \tag{11}$$

- Layer 3

Each node in this layer is also a fixed node. The output of this node is a normalized value of w_i. This can be figured as:

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{12}$$

• Layer 4

This layer is the layer for the consequent parameters. Each node of this layer is an adaptive node, where the parameters can be set using a learning mechanism. The output of this layer is given as:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i), \quad i = 1, 2 \tag{13}$$

$\overline{w_i}$ is the output of layer 3, while $\{p_i, q_i, r_i\}$ are the consequent parameters.

• Layer 5

The last layer is the output of fuzzy system. The output of this layer is obtained by summing all the inputs towards this node.

$$O_i^5 = \sum\nolimits_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{14}$$

3.1. **Proposed ANFIS multioutput design for BLDC speed controller.** In the study, ANFIS multioutput is proposed. The output of ANFIS controller is designed to set PID parameters which would be used to compensate the speed error. If it is using a conventional ANFIS structure, it needs three ANFIS structures to obtain the PID parameters, since the total PID parameters needed are three. Therefore, to minimize the memory and time consuming, ANFIS multioutput is presented. It only needs an ANFIS structure which can result in multiple output at once. Figure 3 shows the proposed ANFIS multioutput structure.

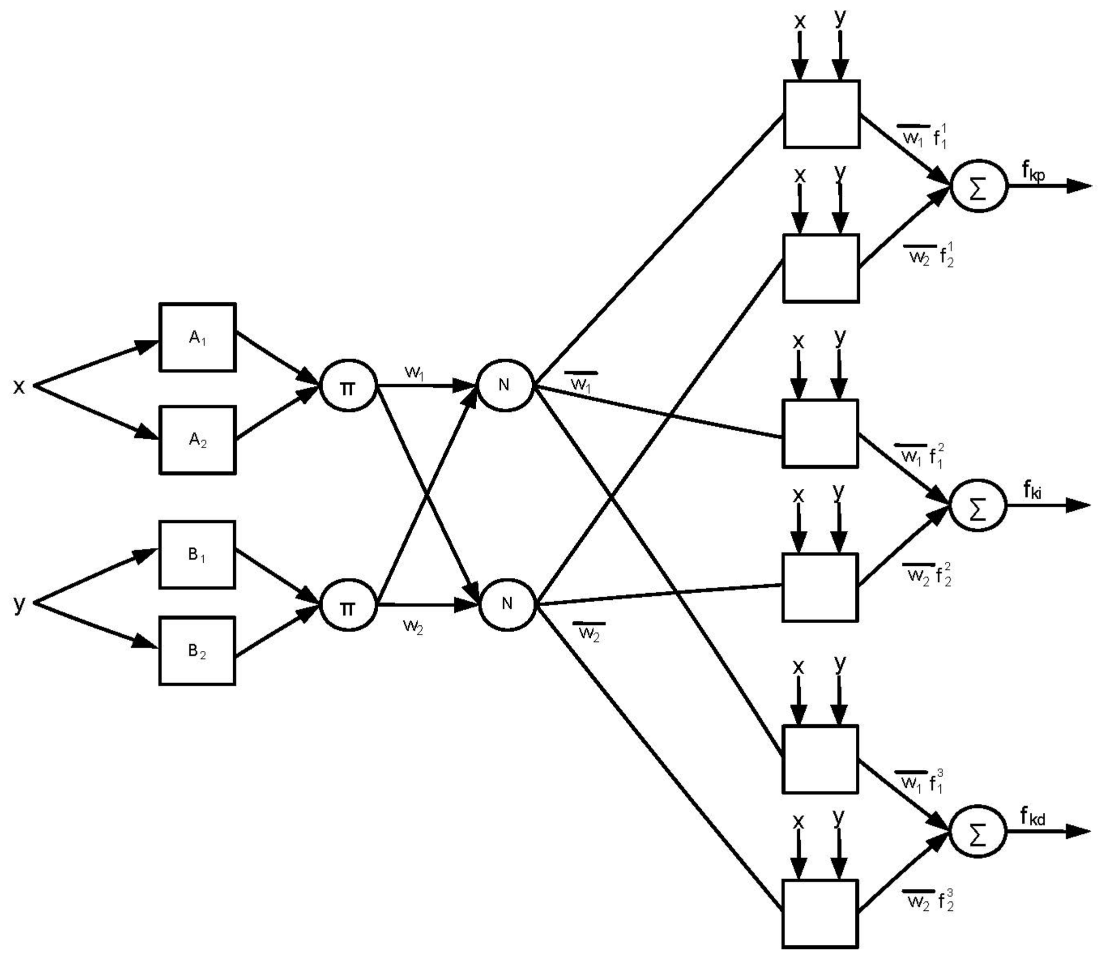

FIGURE 3. Proposed ANFIS multioutput structure

The proposed ANFIS multioutput contains two inputs, error and delta error speed, represented as x and y respectively. On the output side, there are three output values, f_{kp}, f_{ki}, and f_{kd}. Those outputs define the PID parameters used as speed controller of BLDC motor. The output of PID controller is designed to obtain a torque reference since an electric vehicle has to provide an appropriate torque so that the vehicle can run as the speed reference. The scheme of torque referenced for the vehicle is designed in such a way as a direct torque controller. The full diagram of the control scheme is shown in Figure 4.

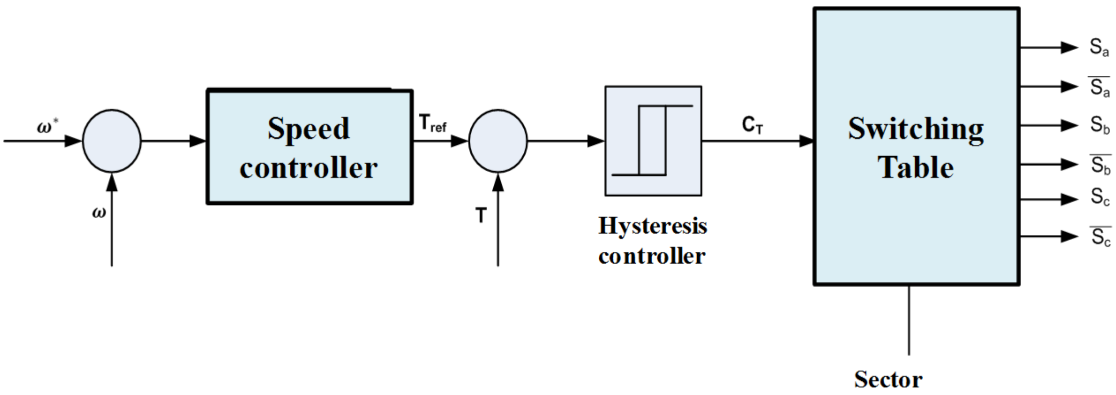

FIGURE 4. BLDC speed controller based on direct torque control

3.2. **Proposed ANFIS multioutput training method.** The proposed ANFIS multioutput is trained using Particle Swarm Optimization (PSO) algorithm for premise parameters and Recursive Least Square Estimation (RLSE) for consequent parameters. PSO is an evolutionary algorithm which is inspired through individual interactions in a population. The algorithm is firstly presented by J. Kennedy in 1997 [18].

Particles in PSO are stated in a D-dimensional space to track and obtain an optimum particular value. Every particle contains information about position and velocity represented as $X_i = (X_{i1}, X_{i2}, X_{i3}, \ldots, X_{iD})$ and $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$ respectively. Every group of particles has the best particle called p_{best}. The value of p_{best} is recorded and represented as $P_i = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{iD})$. Among p_{best} values, then, it is chosen the best value called g_{best}. The velocity and position of each particle are updated in every iteration using Equations (15) and (16).

$$v_i^{n+1} = wv_i^n + c_1 r_{1i}^n \left(p_{best_i}^n - x_i^n\right) + c_2 r_{2i}^n \left(g_{best}^n - x_i^n\right) \tag{15}$$

$$x_i^{n+1} = x_i^n + v_i^{n+1} \tag{16}$$

x_i^n and v_i^n refer to position and velocity of a particle. w is an inertia weight which can improve the performance of PSO itself, c_1 and c_2 are positive constants. In every iteration, it needs random value between 0 and 1 represented as $rand()$. The best position of particles defines the best parameters to be used in ANFIS system and it is obtained depending on a pre-defined fitness function. The fitness function is presented as root mean square error of the output ANFIS compared to training data given.

$$\mathit{fitness} = F(\bar{x}_i) = \min\left(\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(y_{d_n} - y_{out_n}\right)^2}\right) \tag{17}$$

The training data set used in the study is collected from a Fuzzy-PID controller. ANFIS is trained offline using MATLAB m-file code developed through this study. The full algorithm is explained below.

1) In the study, grid partition clustering method is used to initialize the particles' position of PSO.
2) In MATLAB library, it only generates for a single output. Therefore, it needs to modify rule base by adding the output parameters of ANFIS structure.
3) Update the consequent parameters using Recursive Least Square Estimation (RLSE) algorithm. These parameters are firstly set, once before PSO algorithm is run. More about RLSE will be explained on the following sub-chapter below (3.3).
4) Evaluate the ANFIS performance for each particle's position based on the fitness function. The fitness function is the average of root mean square error of each output performance.

$$\mathit{fitness} = F(\bar{x}_i) = \min\left(\frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_{d_n} - y_{out_n})_k^2}\right) \tag{18}$$

   Root Mean Square Error (RMSE) is generally used to evaluate the performance of ANFIS. Since ANFIS has three outputs, the performance evaluation is by calculating the average of RMSE of each output. The best performance is reached when ANFIS obtains the most minimum value of RMSE. The position of particles on PSO algorithm represents as premise parameters of ANFIS. The performance of each particle is evaluated using the value of RMSE.
5) Compare the evaluation results and find the individual with the best fitness value. If $F(\bar{x}_i) < p_{best_i}$, then

$$p_{best_i} = F(\bar{x}_i) \qquad \bar{x}_{p_{best_i}} = \bar{x}_i$$

   The evaluation of each particle is to find the best performance as obtained by fitness function in step 4). The best fitness value is defined as p_{best_i}. According to this performance value, the best position of particle is obtained as $\bar{x}_{p_{best_i}}$.
6) Compare the best individual with all p_{best} value to find the best of the best particle, g_{best}. If $p_{best_i} < g_{best}$, then

$$g_{best} = p_{best_i} \qquad \bar{x}_{g_{best}} = \bar{x}_{p_{best_i}}$$

   From the best fitness value, then evaluate again to find best of the best value, which is called as global best, g_{best}. According to the global best performance value, the best position of particle is called $\bar{x}_{g_{best}}$.
7) Update the velocity of each particle using Equation (15).
8) Update the position of each particle using Equation (16).
   Velocity and position of each particle are always updated in every iteration until the iteration ends.
9) Loop from 4)-8) until the stop condition is obtained.
   The process of finding the best position of particle is looped until the stop condition is obtained. Stop condition could be that RMSE reaches a particular value as user pre-defined or the number of iteration is reached.

3.3. **Recursive Least Square Estimation (RLSE).** RLSE algorithm is widely used to estimate for linear problems [23]. Since the consequent parameters of ANFIS (p_i, q_i, r_i) are linear, RLSE is used in the study. The prediction equation of the linear regression is written as:

$$\hat{y}(t) = \varphi^T(t)\theta \tag{19}$$

$\varphi^T(t)$ is a regression vector, while θ is a vector from unknown parameters. For the t^{th} data, the estimation using LSE is defined as:

$$\hat{\theta}(t) = \left[\sum_{s=1}^{t} \varphi(s)\varphi^T(s)\right]^{-1} \sum_{s=1}^{t} \varphi(s)y(s) \tag{20}$$

Equation (20) contains parameter which is needed to be inversed and it needs more memory and time consuming. The solution is processing it recursively. Equation (20) is changed into a recursive equation as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)\left[y(t) - \varphi^T(t)\hat{\theta}(t-1)\right] \tag{21}$$

$$P(t) = \frac{1}{\lambda}\left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right] \tag{22}$$

This method needs less memory and does not need a complex calculation, since it has no longer an inverse matrix. For a new iteration, it only needs to update P matrix. λ is a forgetting factor parameter affecting the tracking process of parameters. The value of λ is between 0 and 1. The less the value of λ is, the faster the tracking process is, but the more the sensitivity of noise occurs. In general, λ is chosen between 0.94-0.999.

4. **Result and Analysis.** The proposed ANFIS controller design is evaluated to control a 5-kW BLDC motor used in an electric motorcycle developed in our institution and modeled using MATLAB simulation. The parameters of BLDC motor are shown in Table 1.

TABLE 1. BLDC motor parameters

Parameters	Value	Unit
Rated power	5	kW
Rated voltage	100	volt
Rated speed	5000	rpm
R	0.04335	Ω
L	105.2665	μH
k_e	18.935	volt/krpm
k_t	0.180815	N·m/A
B	0.016158	N·m·s
J	0.059009	Kg·m^2
Pole pairs	4	–

R and L are resistance and inductance of phase windings respectively. k_e refers to BEMF constant which defines the voltage generated by the windings when motor is rotated. k_t is a torque constant which defines how much torque produced when current is supplied to the motor. B is a friction constant, the friction between rotor and other mechanical components. J is the total inertia of motor, measured when motor is coupled with gear or belt connected to the wheels.

4.1. **ANFIS training result.** The training data set is collected through Fuzzy-PID controller result. Input of *error* and Δ*error* has range between -500 and 500 and is chosen as 20 data for each input. Therefore, through combination between *error* and Δ*error*, it would be 400 training data used to train ANFIS multioutput. Each input of ANFIS has 2 membership functions: N (Negative) and P (Positive). Figure 5 shows the initial membership function of ANFIS.

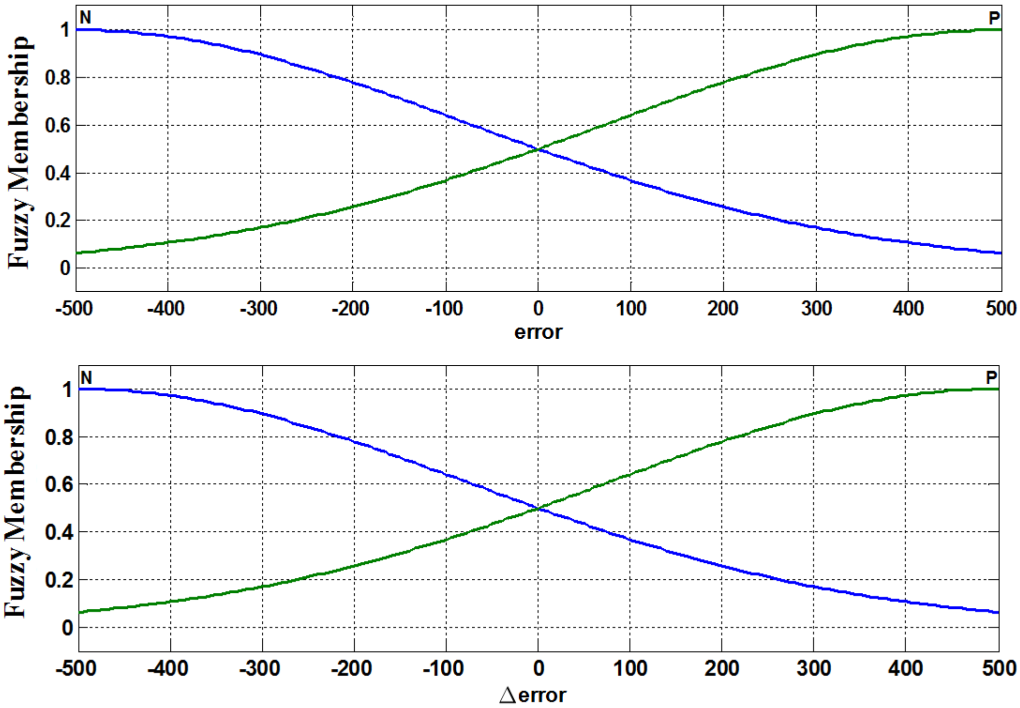

Figure 5. Initial membership function of ANFIS multioutput

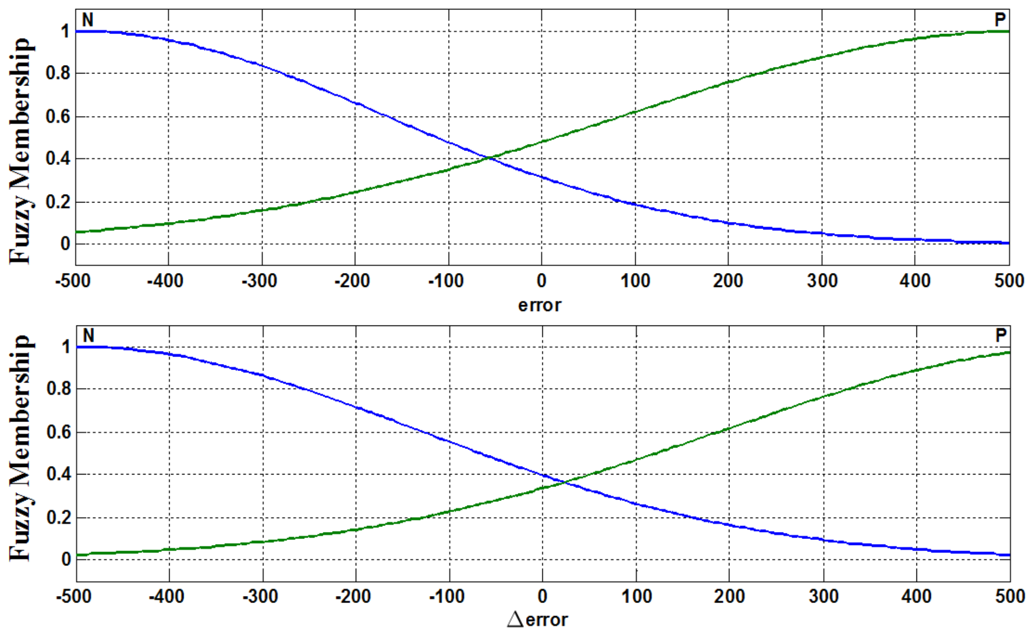

Figure 6. Final membership function of ANFIS multioutput after training ($\lambda = 1$)

The first training is done by setting $\lambda = 1$, and then decreased from 0.99 to 0.94. Figures 6 and 7 show the results of ANFIS training for $\lambda = 1$. According to the result, it is obtained that for $\lambda = 1$, the convergence reached when RMSE at 0.05364.

4.2. **Simulation BLDC speed controller with ANFIS.** From the ANFIS training results that have been obtained, further testing is performed on the speed control system simulation. Comparison of speed response for each result is done with different forgetting

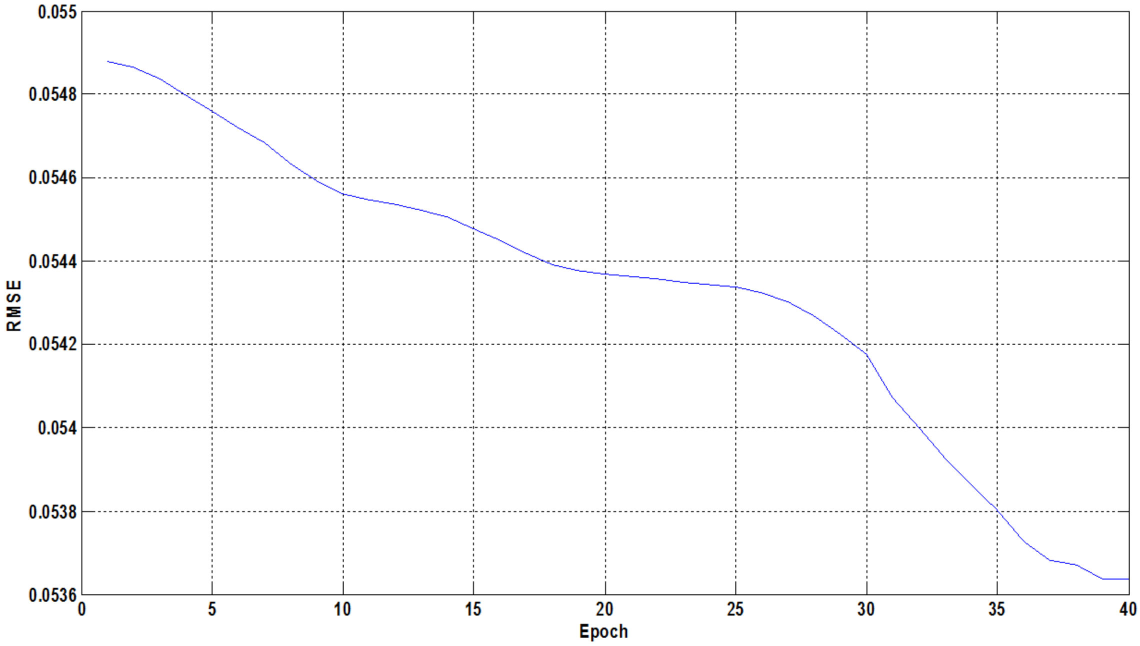

Figure 7. ANFIS training performance using RMSE ($\lambda = 1$)

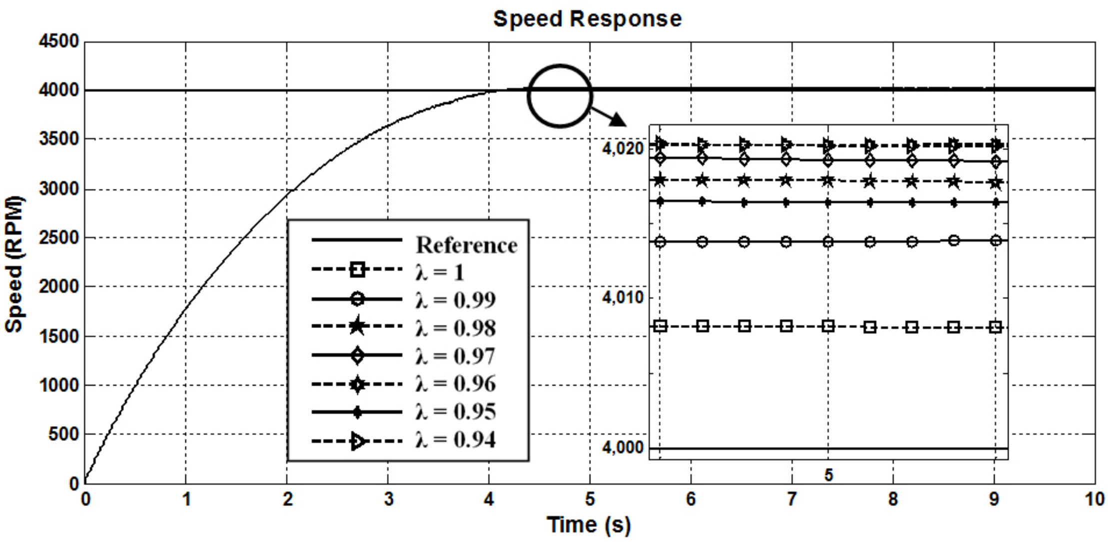

Figure 8. Speed response with ANFIS

factor (λ). The best results will be the parameters to be implemented on the hardware. The results of the speed response with ANFIS control are shown in Figure 8. Based on the test results, the best ANFIS model in testing the speed response is when $\lambda = 1$, with a steady state error of 0.2% with a rise time of 2.7437 s.

4.2.1. *Simulation with fixed speed reference and no loads.* At the condition of fixed reference speed and no load, we will observe the current and torque characteristics in the BLDC motor system to be tested. Speed is set at 1000 rpm. Simulation results for fixed and no-load speed are shown in Figures 9-11. The speed error at steady state for ANFIS controller is 0.10%, while 0.15% for Fuzzy-PID, so it can be said that the speed control system using ANFIS has better response than Fuzzy-PID. With a fixed reference speed of 1000 rpm, the start current is limited to 100 A and the steady state peak current

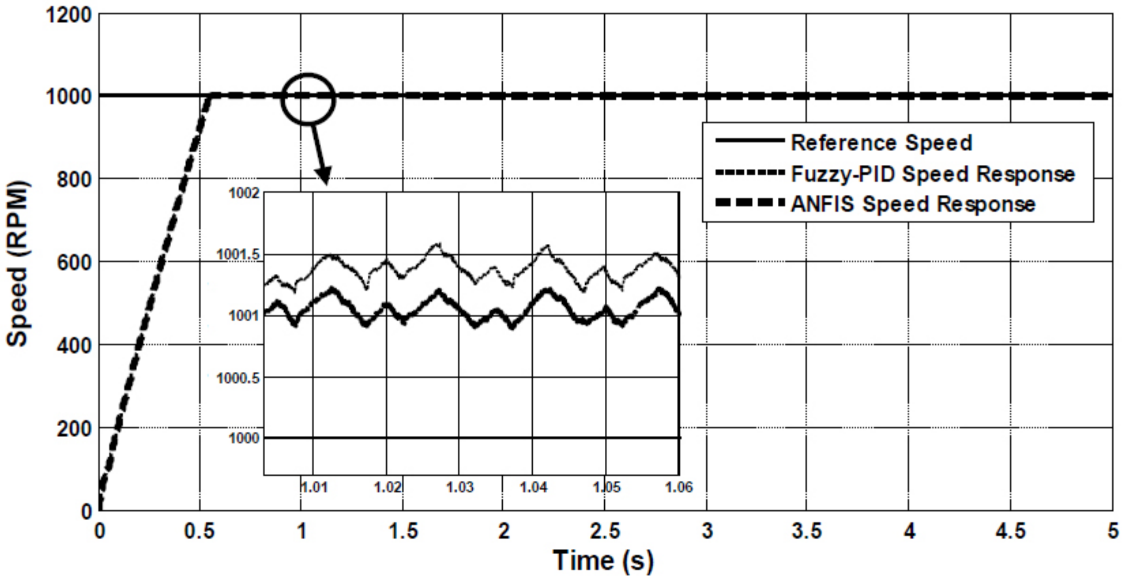

FIGURE 9. Speed response with fixed reference under no load condition

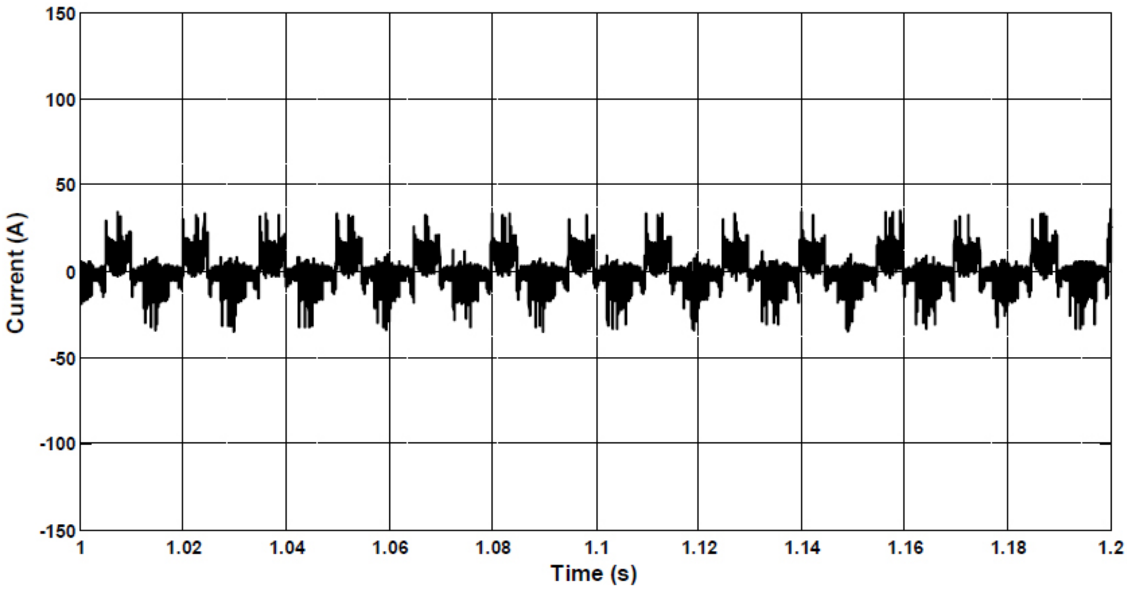

FIGURE 10. Current waveform in phase a (Ia) at 1000 rpm

reaches 33.45 A with an rms value of 10.5 A. Steady state conditions are achieved at 0.5 s with a rise time of 0.415 s. DTC based speed control systems have large torque ripples. The torque at the start motor is limited to 11.1 N·m. Figure 11 shows the estimated torque comparison with actual torque. Based on the comparison that has been made, the torque can be well estimated. However, at the starting condition, the estimated torque is very bad. This is because during starting condition, BEMF voltage cannot be detected properly, causing the estimation process does not work well.

4.2.2. *Simulation with varying speed reference and no loads.* Simulation with varying speed reference is done in order to know the response of control system which has been designed to follow a given reference value. The speed reference is set from 0 to 3500 rpm. Figure 12 shows the response of speed with changing speed reference.

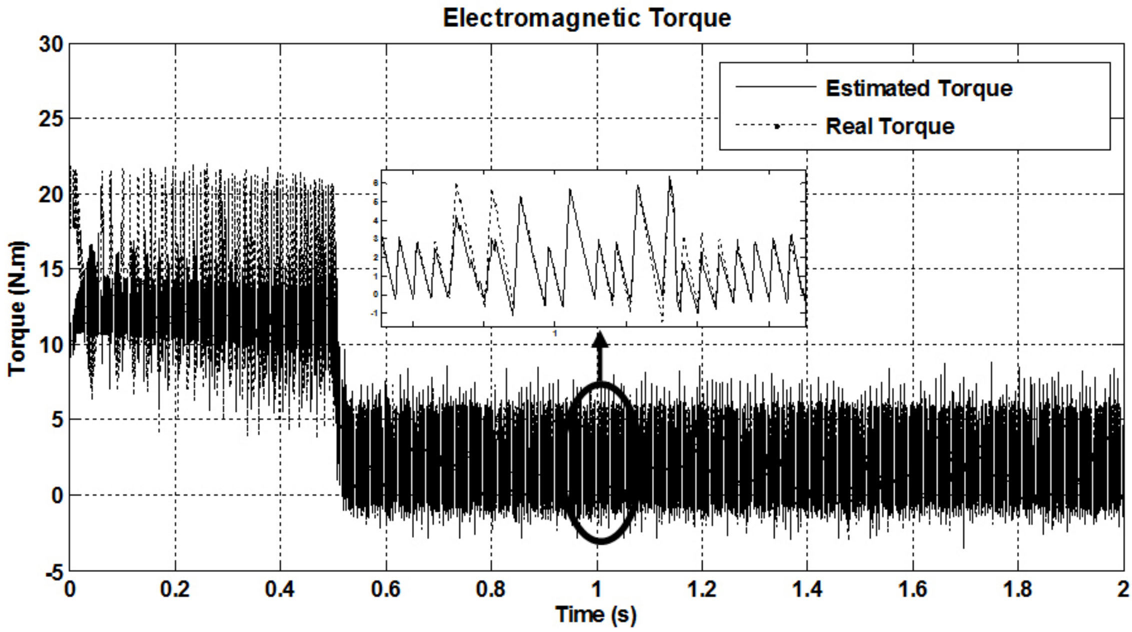

FIGURE 11. Comparison of estimated torque vs real torque

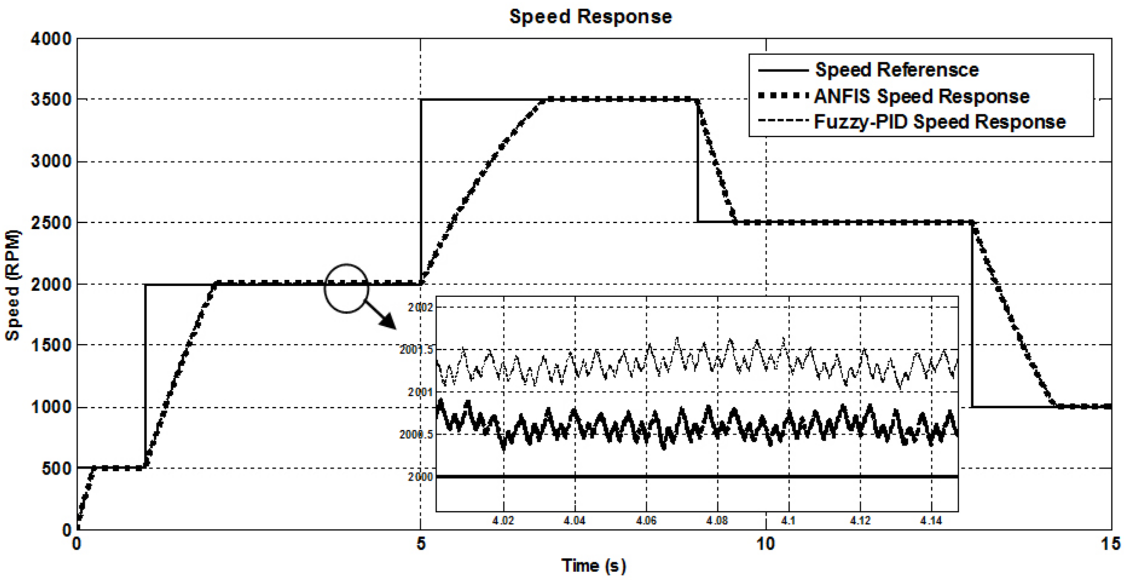

FIGURE 12. Speed response with varying reference

From Figure 12, it can be observed that ANFIS controller is able to follow the track according to the given speed reference. If it is zoomed in detailed, the steady state error of the ANFIS controller has a smaller value than Fuzzy-PID.

Current conditions and electromagnetic torque when the speed reference changes are shown in Figures 13 and 14. It can be seen that based on the simulation, there is a surge of current on the stator, when there is a change in speed, both in the condition of increasing and decreasing speed. This is because at the time of the acceleration, greater torque is required, so the voltage source V_{dc} enlarges the current supplied to the stator. When the vehicle tries to change the speed below the normal speed, in this condition, regenerative braking occurs, where the current stored on the stator winding, flows back to the system, this is indicated by the torque of negative value as shown in Figure 14.

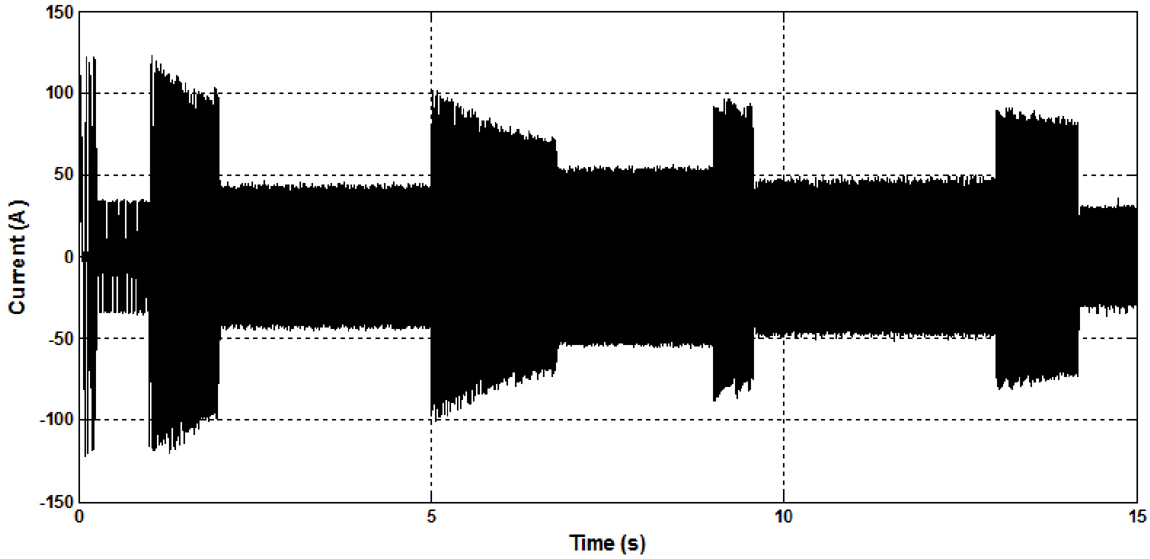

Figure 13. Current Ia with varying speed

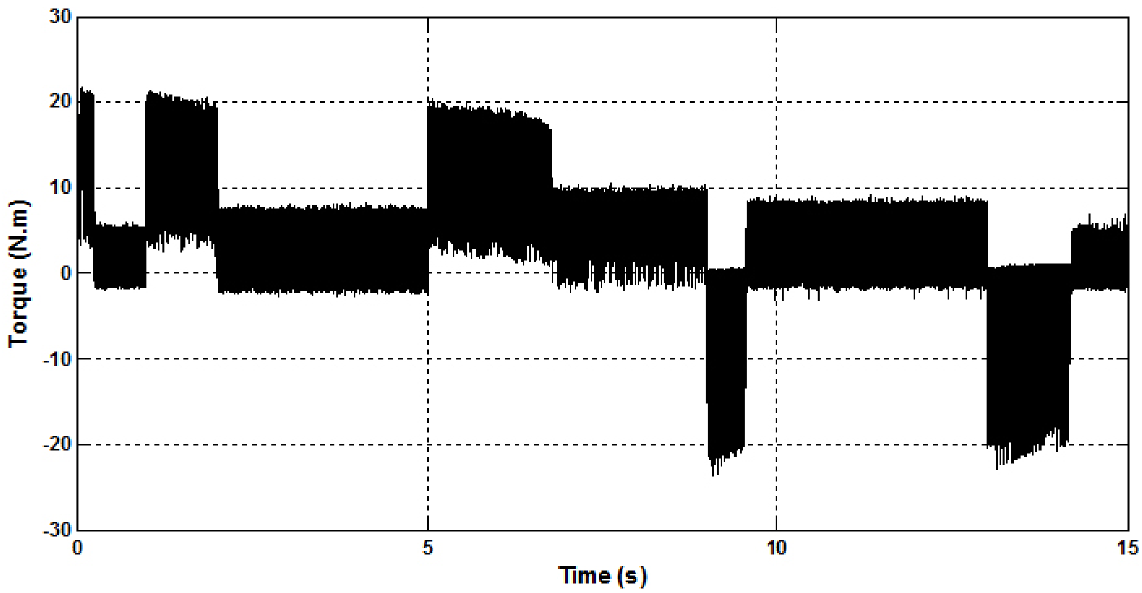

Figure 14. Electromagnetic torque with varying speed

4.2.3. *Simulation with fixed speed reference and varying loads.* In this case, simulations with fixed speed reference with load changes at any given time, represent dynamically load change in an electric vehicle. The load changes that occur in this simulation are shown in Figure 15. Furthermore, the control system is tested whether it is able to restore the condition of the system according to the given speed reference.

Figure 16 shows the speed response during load changes. At the 5 second, the system is loaded at 5 N·m, and at 8 second the load is raised to 8 N·m. From both cases it is seen that when the load is increased, the speed response decreases; however, the speed decrease is still within the tolerance of the error value, with an error value of 0.1%. The decrease in speed at the time of the load changes, does not change the value of the speed significantly, this is because the control system designed, considering the value of speed, also considers the torque required by the system. The current and the electromagnetic torque under load change conditions, each phenomenon is shown by Figures 17 and 18.

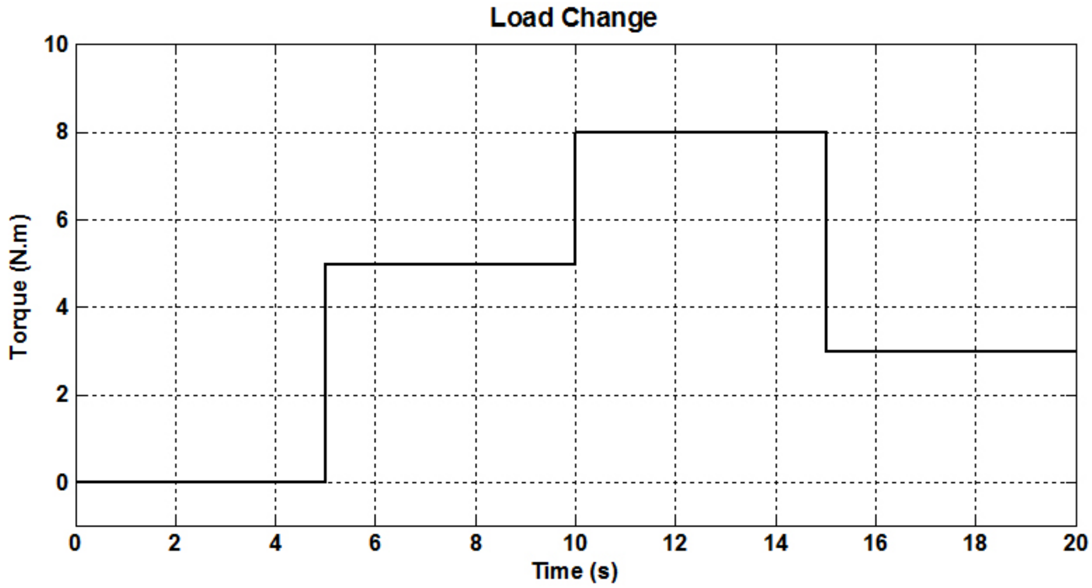

FIGURE 15. Load profile

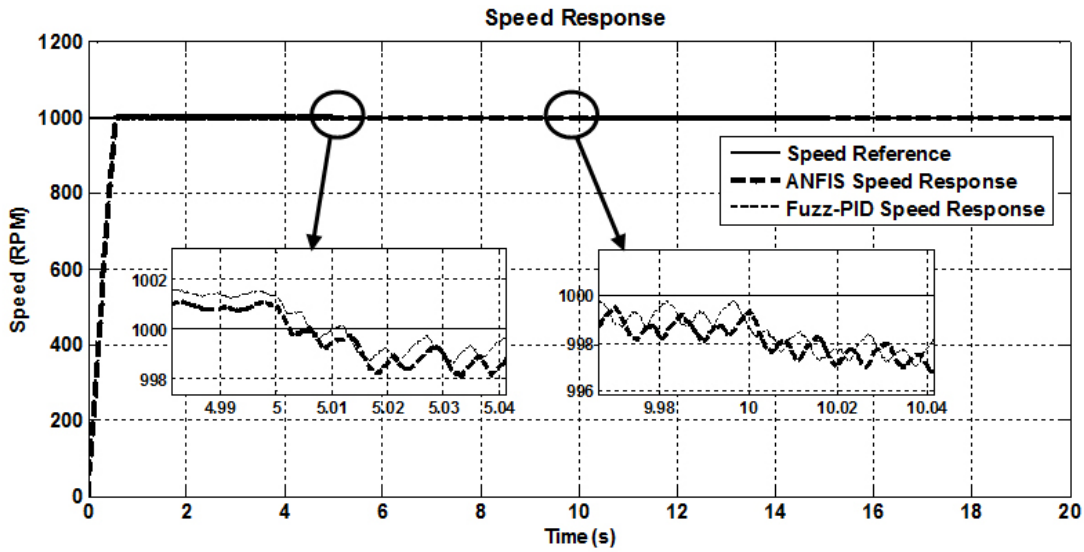

FIGURE 16. Speed response with varying load

Based on the simulation results shown in Figures 17 and 18, when the load changes occur, the system can directly respond to the need for torque well. This is seen in the changes of the electromagnetic torque and current on the stator in the event of a load change.

5. **Conclusion.** Hybrid PSO and RLSE methods can be used for training ANFIS multi-output systems. From the results obtained in this study, by using Fuzzy-PID as a learning supervisor, the ANFIS method based on PSO-RLSE is able to train the pattern that has been determined with RMSE of 0.05364 with a value of $\lambda = 1$. The value of λ greatly affects the learning process in ANFIS. The smaller value of λ will be able to make the learning process not convergent.

The BLDC motor speed control system using ANFIS has been well designed. The results show that the speed control system using ANFIS based on DTC (Direct Torque Control) is able to maintain the speed value by considering the torque required by the system. The

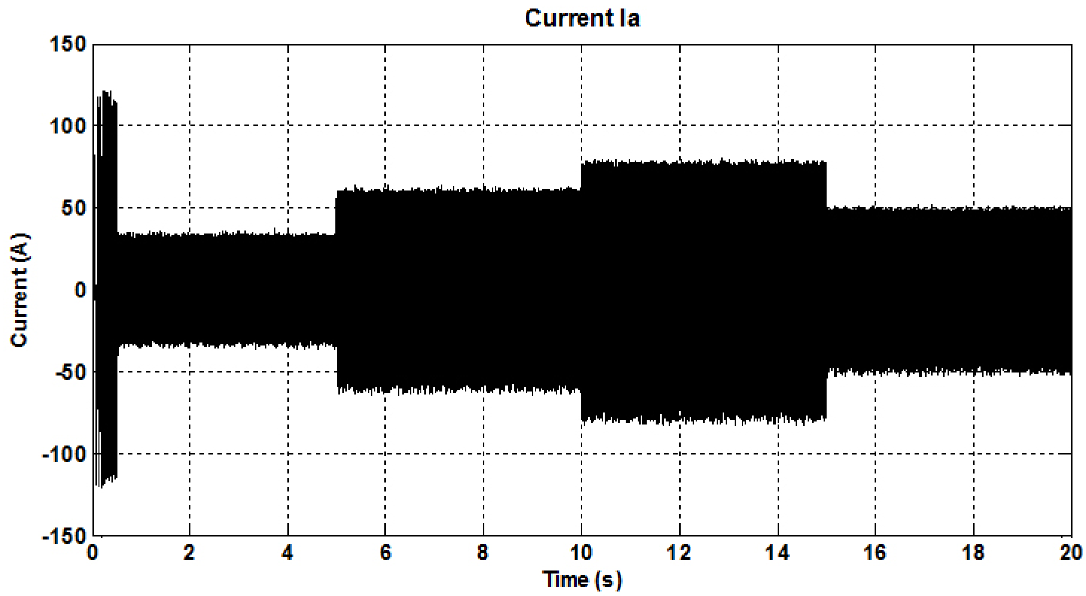

FIGURE 17. Current waveform in phase a (Ia)

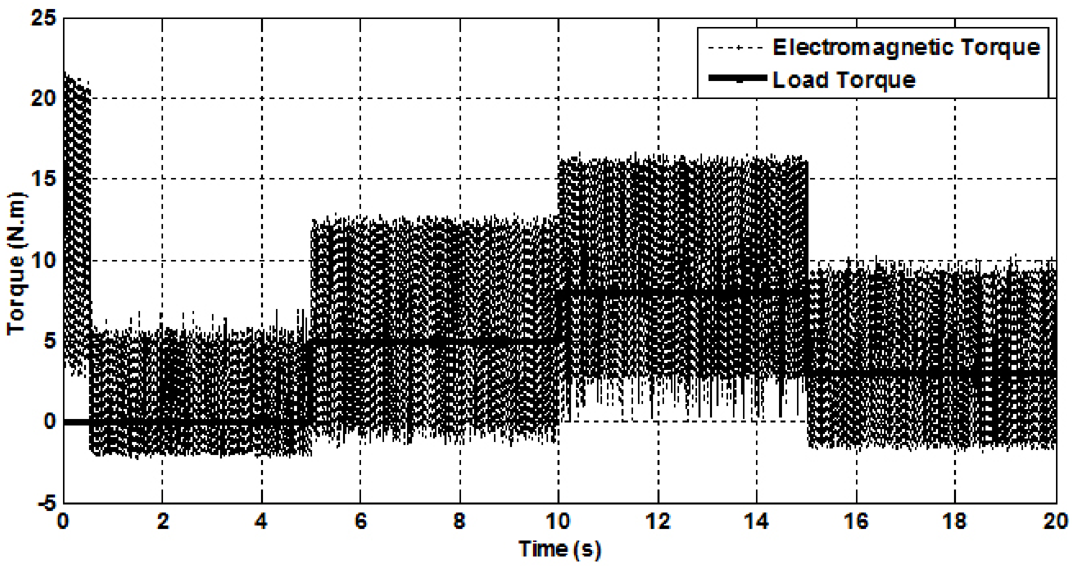

FIGURE 18. Electromagnetic torque under varying load condition

average steady state error achieved using ANFIS is 0.1% with a rise time of 2.7437 s for a reference speed of 4000 rpm. In the event of increasing in load, the speed performance will decrease; however, this decrease value is not too significant and can still be tolerated using the designed speed control system.

Based on the simulation, the control system using ANFIS is much better than Fuzzy-PID. With the same performance, ANFIS has the advantage of fewer variables than Fuzzy-PID, so that the execution process on the microcontroller becomes faster and requires less storage memory. Thus, ANFIS can be implemented in a real-time system where the execution process becomes a priority choice.

REFERENCES

[1] X. D. Xue, K. W. E. Cheng and N. C. Cheung, Selection of electric motor drives for electric vehicles, *IEEE Australasian Universities Power Engineering Conference (AUPEC'08)*, Sydney, 2008.

[2] M. Yildirim, M. Polat and H. Kürüm, A survey on comparison of electric motor types and drives used for electric vehicles, *IEEE the 16th International Power Electronics and Motion Control Conference and Exposition*, Turkey, 2014.
[3] B. K. Lee and M. Ehsani, Advanced BLDC motor drive for low cost and high performance propulsion system in electric and hybrid vehicles, *IEEE International Electric Machines and Drives Conference (IEMDC)*, 2001.
[4] B. N. Kommula and V. R. Kota, Performance evaluation of hybrid fuzzy PI speed controller for brushless DC motor for electric vehicle application, *IEEE Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG)*, 2015.
[5] M. Singh and A. Garg, Performance evaluation of BLDC motor with conventional PI and fuzzy speed controller, *IEEE the 5th India International Conference on Power Electronics (IICPE)*, 2012.
[6] R. Somanatham, P. V. N. Prasad and A. D. Rajkumar, Modeling and simulation of sensorless control of PMBLDC motor using zero-crossing back E.M.F detection, *IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, 2006.
[7] K. H. Ang, G. Chong and Y. Li, Patents, software, and hardware for PID control: An overview and analysis of the current art, *IEEE Control Systems Magazine*, 2006.
[8] K. H. Ang, G. Chong and Y. Li, PID control system analysis, design, and technology, *IEEE Trans. Control Systems Technology*, vol.13, no.4, 2005.
[9] S. Skoczowski, S. Domek, K. Pietrusewicz and B. Broel-Plater, A method for improving the robustness of PID control, *IEEE Trans. Industrial Electronics*, vol.52, no.6, 2005.
[10] B. M. Mohan and A. Sinha, Analytical structure and stability analysis of a fuzzy PID controller, *Applied Soft Computing*, vol.8, 2008.
[11] A. Rubaai, M. J. Castro-Sitiriche and A. R. Ofoli, Design and implementation of parallel fuzzy PID controller for high-performance brushless motor drives: An integrated environment for rapid control prototyping, *IEEE Trans. Industry Applications*, vol.44, no.4, 2008.
[12] R. Arulmozhiyal and R. Kandiban, Design of fuzzy PID controller for brushless DC motor, *IEEE International Conference on Computer Communication and Informatics (ICCCI)*, 2012.
[13] R. Arulmozhiyal, Salem, design and implementation of fuzzy PID controller for BLDC motor using FPGA, *IEEE International Conference on Power Electronics, Drives and Energy Systems*, 2012.
[14] Y. Wang, Y. Yu, G. Zhang and X. Sheng, Fuzzy auto-adjust PID controller design of brushless DC motor, *International Conference on Medical Physics and Biomedical Engineering (ICMPBE)*, vol.33, 2012.
[15] G. Cheng, Brushless DC motor speed control system based on fuzzy PID controller, *The 2nd International Conference on Network Computing and Information Security (NCIS)*, 2012.
[16] A. Rubaai and P. Young, Hardware/Software implementation of fuzzy-neural-network self-learning control methods for brushless DC motor drives, *IEEE Trans. Industry Applications*, vol.52, 2016.
[17] P. Liu, W. Leng and W. Fang, Training ANFIS model with an improved quantum-behaved particle swarm optimization algorithm, *Mathematical Problems in Engineering*, 2013.
[18] J. Kennedy, The particle swarm: Social adaptation of knowledge, *Proc. of IEEE Int. Conf. Evol. Comput.*, pp.303-308, 1997.
[19] H. E. A. Ibrahim, F. N. Hassan and A. O. Shomer, Optimal PID control of a brushless DC motor using PSO and BF techniques, *Ain Shams Engineering Journal*, vol.5, pp.391-398, 2014.
[20] M. Nasri, H. Nezamabadi-pour and M. Maghfoori, A PSO-based optimum design of PID controller for a linear brushless DC motor, *Proc. of World Academy of Science: Engineering and Technology*, vol.20, 2007.
[21] S. Badurson, *BLDC Motor Modelling and Control – A Matlab/Simulink Implementation*, Master Thesis, International Master Program in Electrical Power Engineering, Chalmer Tekniska Hogskola, 2005.
[22] C. Xia, *Permanent Magnet Brushless DC Motor Drives and Controls*, Science Press, 2012.
[23] B. Carlsson, *Recursive Identification*, Uppsala Universitet, Institutionen För Informationsteknologi, 2011.
[24] H. Suryoatmojo, N. R. Arsya, R. Mardiyanto, D. C. Riawan, S. Anam and M. Ashari, Design of electronic speed controller for BLDC motor based on single ended primary inductance converter (SEPIC), *International Seminar on Intelligent Technology and Its Application (ISITIA)*, Indonesia, 2017.
[25] G. Cho, K. Baik and H.-S. Kim, Study on advanced behavior analysis for towfish using particle-based simulation and real-time rendering, *ICIC Express Letters*, vol.12, no.3, pp.263-268, 2018.