Research Article

Fuzzy PI Control of Trapezoidal Back EMF Brushless DC Motor Drive Based on the Position Control Optimization Technique

D. Magdalin Mary, 1 C. Kumar, 2 Felix Joseph Xavier, 3 Sara A. Rashad, 4 Hady H. Fayek 5, Naganthini Ravichandran, 6 and Sourav Barua 7

1 Sri Krishna College of Technology, Coimbatore, India
2 M.Kumarasamy College of Engineering, Karur, India
3 Bethlahem Institute of Engineering, Kanyakumari, Tamilnadu, India
4 Heliopolis University, Cairo, Egypt
5 Heliopolis University, Cairo, Egypt
6 University of Naples Federico II, Naples, Italy
7 Green University of Bangladesh, Dhaka, Bangladesh

Correspondence should be addressed to Sourav Barua; barua@eee.green.edu.bd

Received 17 February 2022; Revised 5 May 2022; Accepted 10 May 2022; Published 20 July 2022

Academic Editor: Jie Hu

Copyright © 2022 D. Magdalin Mary et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this article, a novel methodology has been created based on optimized fuzzy PI controller using position control optimization (PCO). This proposed PCO algorithm ensures better control of brushless direct current motor (BLDCM) in order to reduce steady-state error and oscillation as well as improve the system response by optimizing the speed and position of rotor. The suitability of this technique has been simulated in different stages, so as to infer the optimization in results. The main motivation of this paper is to get better efficiency and optimum response from BLDCM, by comparing its performance characteristics with existing methods. The detailed investigation research through simulation is carried out using MATLAB/Simulink. However, in order to obtain well-defined results, the PCO technique was applied and the performance obtained was highly optimized than previously applied techniques. Through the simulations, the proposed approach can outperform the existing approaches such as PSO and moth-flame algorithm (MFA).

1. Introduction

Innumerable research works have been conducted to develop stochastic and heuristic algorithms in the last few years to find the optimal solution. Metaheuristic is a high-level design algorithm used to find, create, or select an experimental data that could provide a satisfying solution for an optimization problem, with incomplete or inexact data. A subset of solutions from a large group will be entirely enumerated by metaheuristics. In comparison with optimization algorithms and other iterative analysis, metaheuristics do not assure an ideal solution for classes of problems. Various metaheuristics implementations are in the form of stochastic optimization, and the response of the solution found is relay on random variables. Through various studies conducted in the field of optimization, it has been observed that modern approaches suffer from different limitations in providing a suitable solution. At the same time, algorithms such as genetic, evolutionary, and ant colony have been applied successfully to different optimal problems. Since in this work, the emphasis has been laid on optimizing the performance of BLDC motor, it becomes contrary to other brushed motors, which feature higher efficacy and reliability at a lower cost. The BLDCM is actually a permanent magnet (PM) synchronous motor, featuring a trapezoidal back EMF waveform shape and having numerous advantages such as a different design higher power density and better controllability. The PM synchronous
motor featuring a sinusoidal back EMF and BLDCM with trapezoidal back EMF is exploited to a larger extent in many industrial applications. Additionally, in contrary to the PM synchronous motor, the BLDCM is more affordable because of the concentrated windings, which reduce the length of end windings. Power semiconductor switches are employed to run the BLDC motor, whereas feedback signal is sensed by an encoder to know the position of the rotor.

1.1. Literature Review. From the literature, different papers have been carried out to get an optimal performance. The BLDC motor exhibits an extensive range of speed based on control. While considering the speed controller performance, it is necessary that parameters such as rise time, starting torque, and current should also be observed more carefully. Additionally, speed control also ensures a controlled torque value. Primarily, two approaches have been applied for controlling the speed. Performances of PI controllers and fuzzy PI controllers have been discussed precisely [1, 2]. The classical PID controller is employed for the majority of industrial applications. But the position response offered for nonlinear systems is not optimum [3].

For example, in the case of motor speed control, employing a PID controller increases noise and consequently impacts its application in industries. To alleviate this problem of increased noise generation and enhancing the stability of the system, in most of the industrial applications, PI controllers are used [4] [5]. Through PI controllers, tuning of nonlinear systems is a highly complex task. The aforementioned discussion clearly exhibits that the PI controller alone does not fulfill the requirements of nonlinear systems. Hence, fuzzy PI controllers are preferred in nonlinear systems, and it is necessary to have a wide range of acquaintance to tune the system parameters [6]. The fuzzy control approach has higher stability, simpler configuration, and minimum fluctuation due to dynamic load conditions and considerably lower cost [7]. The application of fuzzy logic controllers is generally considered for nonlinear complex systems, and it has been applied for different applications for a longer time efficiently [8]. Even though the response of the oscillation is smaller in fuzzy logic controllers when compared to PI controller, the system takes much time to settle down [9]. Various studies have been carried out for the performance of brushless DC motor. The response of the system using PI controller has the initial rise time of 0.008 seconds (fuzzy PI—0.002 sec, MFA—0.002 sec), 13% of peak overshoot (fuzzy PI—7%, MFA—1%), and the time taken for the system to settle down at 0.1 seconds (fuzzy PI—0.1 sec, MFA—0.8 sec). In fuzzy PI controller, the response of the system has high ripples and quick settling time. This limitation is overcome by optimizing the control parameters with PCO techniques.

1.2. Research Gap and Motivation. From the literature survey, the performance of the system for various classical controllers has been studied and the characteristic responses were analyzed. In early stages, a PI controller was employed, which gave better results while using classical approaches. For tuning of nonlinear system, fuzzy logic controller has been used but the system takes much times to settle down. Further, a fuzzy PI controller is implemented so as to improve the results obtained previously. However, in order to obtain well-defined results, the PCO technique was applied and the performance obtained was highly optimized, than those previously applied techniques. This proposed approach, in view of Newton’s second law of motion, continuously updates the speed and position, resulting in higher accuracy. The proposed approach has been analyzed through different benchmark functions. Through the simulations, the proposed approach can outperform the existing approaches such as particle swarm optimization (PSO) and moth–flame algorithm (MFA). It can be suitable for both linear and nonlinear systems.

The main motivation of this paper is to get better efficiency! and optimum response from BLDCM, by comparing its performance characteristics with existing methods. The detailed investigation is done with the help of MATLAB/ Simulink.

1.3. Contribution and Paper Organization. The proposed research technique influences the fuzzy PI control approach to get an optimal solution of BLDC motor. Considering the limitations mentioned above and the significance of PI controllers, the fuzzy logic concept is applied in conjunction with PCO, which fulfills the objective of optimizing the gain values of the controller. Some significant characteristics of fuzzy control are improved stability, cost-effectiveness, simple control configuration, and less sensitivity to load dynamics. The proposed approach exhibits an optimal performance of the fuzzy PI controller. The power factor correction (PFC) using bridgeless buck-boost converter-fed BLDC drive, using PI controller and fuzzy logic controller, is compared with their independent performance of speed control [10]. Similarly, the authors have discussed BLDC motor’s superiority over the classical DC motor [11]. They have asserted that though PI controller is widely exploited in industrial applications due to the larger settling time, an adaptive PID fuzzy controller can be more efficient than PI controller. Contrary to the proposed work, using an adaptive fuzzy PI control approach to control different BLDC motor parameters is a more efficient method [11, 12]. Fuzzy proportional-integral controller has many advantages over conventional PI controllers. A fuzzy controller gives optimum speed response for initial condition. A PI controller has good response over different load torques, but it takes longer time to settle down. To overcome this limitation, fuzzy PI controller has an incorporating advantage over fuzzy and PI controllers [13, 14]. They are aimed at optimizing the performance of the controller in terms of different variations in the parameters. To ensure optimal results, they have simulated proposed models for different speeds and loads.

The system response of each optimization technique PSO and MFA is compared with present approach, and the results are analyzed ([15]. PSO is a type of computational algorithm. The algorithm is constructed using natural
phenomena such that every particle is considered to be a possible solution. The suitable regions from complex search spaces are identified by observing interaction of particles in a group. The particles are initialized randomly, and they travel through multidimensional space. During this period of their flight, the particles update their position and velocity based on their individual analysis and the analysis of the entire group. This process of updating the position and velocity will facilitate the individual particles to accumulate around the location having the best fitness functions. When compared with other metaheuristic algorithms, PCO techniques give better responses than other techniques. If the parameters are too large, the particle swarm will fail to identify the optimal solution and it is unable to converge, since the entire particles move toward optimal solution. During convergence, PSO algorithm does not continue to optimize. Limitation of PSO is easy to jump into local optima in high-dimensional interplanetary and has a low convergence rate toward it. MFA algorithm is developed based on a navigation mechanism, termed as transverse orientation used by moths [16]. It will permit the moths to fly in a straight line. The moths are able to maintain long-distance flights with the help of the moon. However, if they come across any artificial or man-made light, they are distracted and exhibit spiral behavior. This moth behavior has been modeled mathematically, and the moth-flame algorithm is proposed [17].

2. Mathematical Modeling

The modeling of the BLDC motor has been performed using MATLAB/Simulink.

2.1. BLDC Motor Modeling. The differential equations for terminal voltages of the BLDC motor is shown in (1), and the simulation is performed in MATLAB toolbox [18].

\[ V_a = R_a i_a + L_a \frac{d i_a}{dt} + M_{ac} \frac{d i_b}{dt} + M_{bc} \frac{d i_c}{dt} + e_a, \]  
\[ V_b = R_b i_b + L_b \frac{d i_b}{dt} + M_{ac} \frac{d i_a}{dt} + M_{bc} \frac{d i_c}{dt} + e_b, \]  
\[ V_c = R_c i_c + L_c \frac{d i_c}{dt} + M_{ac} \frac{d i_a}{dt} + M_{bc} \frac{d i_b}{dt} + e_c. \]  

In the previous equations, \( R_{a-b} \) refers to the resistance in each phase, \( L_{a-b} \) denotes the inductance of each phase, and \( M_{ac} \) and \( M_{bc} \) refer to mutual inductances. Similarly, \( i_a, i_b, \) and \( i_c \) are the stator current in each phase and \( V_a, V_b, \) and \( V_c \) are the phase voltages [19]. Figure 1 illustrates the circuit diagram and schematic diagram of the motor [20].

Equation (2) indicates the calculation of other parameters of the BLDC motor, such as torque and EMF. These parameters are trapezoidal in nature [21].

\[ e_a = f_a(\theta)K_e \omega, \]  
\[ e_b = f_b(\theta)K_e \omega, \]  
\[ e_c = f_c(\theta)K_e \omega. \]  

Figure 2 indicates the control structure of BLDC motor to get the gate signal based on the back EMF. The following equation gives the electromagnetic torque of the motor.

\[ T_e = \frac{1}{\omega_e} (e_a i_a + e_b i_b + e_c i_c). \]  

In the above equation, \( \omega_e \) indicates the rotational speed of the motor. The equation is given as

\[ \frac{d}{dt} \omega_e = \frac{T_e - T_L - B \omega_e}{J}. \]  

The relation between the mechanical \( (\omega_e) \) and electrical speed \( (\omega_c) \) is as follows:

\[ \omega_c = \left( \frac{P}{2} \right) \omega_e. \]  

The final output power of the motor is given as follows:

\[ p = T_e \times \omega. \]  

In the previous equation, \( p \) refers to the total output power and \( \omega \) denotes the angular velocity (radians per second) of the motor.

The physical parameters such as resistance, inductance, and back EMF are primarily responsible for ensuring optimal control response of BLDCM. The parameters in Table 1 cause disturbances such as maximum overshoot and unsteady state, which decreases the system’s time response.

To alleviate such disturbances, it is necessary that the gain parameters of the motor are controlled using a PI controller. For optimal response, a fuzzy PI controller is implemented, which optimizes the motor’s gain parameters.

3. Optimization Methods

Optimization can be defined as a process of exploring and discovering the most suitable solution for resolving a certain issue. There have been encounters of several complex problems in the last few years, resulting in the development of some new optimization approaches. Exploring the literature, it was found that several important research works are conducted toward optimization problems. Consequently, numerous algorithms with each one aiming to resolve a particular issue have been developed. Some of the key algorithms developed are genetic algorithm (GA) [22], particle swarm optimization (PSO) [23], ant colony optimization (ACO) [9], differential evolution algorithm (DEA) [24], firefly algorithm (FA) [25], bat algorithm (BA) [26], gravitational search algorithm (GSA) [27], flower pollination algorithm (FPA) [28], etc. A snippet of some algorithms that are considered in this work is given below. The optimization problem is consumed to improve the conduct of an electric motor by varying the specific parameters and is used to find the best solution from all possible solutions. In high-
dimensional space, PSO algorithm is unable to converge continuously. To limit the problem inward in PSO algorithm and the consequences faced in MFA algorithm due to transverse orientation, PCO is used to converge in all conditions and provide optimal solution.

3.1. Particle Swarm Optimization (PSO). PSO is primarily developed simulation in three-dimensional space [15]. The speed of individual particles of the group is represented by $v_x$ (velocity of $x$-axis), $v_y$ (velocity of $y$-axis), and $v_z$ (velocity of $z$-axis). Any change in the position of these particles can be estimated through the data of position and velocity [29].

Certain fitness functions are optimized by bird flocking. It is noteworthy that each particle in the group is now familiar with its optimal value (Po). Additionally, the particle is also familiar with its optimal value in the entire group (Go). This can be helpful for the particles to know about the performance of other particles in the group. Thus, each particle in the group aims at optimizing its position by knowing about the current position ($x$, $y$, $z$), current velocities ($v_x$, $v_y$, $v_z$), the gap between the present
position and Po, and the gap between the current situation and Go.

This can be explained through the given equation [27].

\[ \psi_{i}^{k+1} = \omega \psi_{i}^{k} + C_{1} \text{rand}_{1} \times \left( \text{Po} - \psi_{i}^{k} \right) + C_{2} \text{rand}_{2} \times \left( \text{Go} - \psi_{i}^{k} \right), \]  

(11)

where \( \psi_{i}^{k} \) represents velocity of particle I at iteration k.

\[ \psi_{i}^{k+1} = \psi_{i}^{k} + \psi_{i}^{k+1}. \]  

(12)

3.2. Moth-Flame Algorithm (MFA). This algorithm is derived from nature, or in other words, it can be asserted that it is nature-based. This algorithm considers the particles as moths and the problem’s parameters as the position of moths. Contrary to the PSO algorithm, MFA enables particles to move in multidimensional space by modifying their position vectors. The matrix representation of it is given as follows:

\[ M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,d} \end{bmatrix}. \]  

(13)

In this matrix, \( n \) refers to a quantity of moths, and \( d \) indicates the dimension. Also, it is assumed there is a list for storing the values of respective fitness functions for all the moths. The list can be represented as follows:

\[ PM = \begin{bmatrix} PM_{1} \\ PM_{2} \\ \vdots \\ PM_{n} \end{bmatrix}. \]  

(14)

The first row of the matrix \( M \) represents the position vector for the moth, which is input for fitness function. The equivalent of fitness function is provided to consistent moth.

Besides moths, the flame is another significant part of this algorithm. It can be represented as follows:

\[ F = \begin{bmatrix} F_{1,1} & F_{1,2} & \cdots & F_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n,1} & F_{n,2} & \cdots & F_{n,d} \end{bmatrix}. \]  

(15)

Similar to the moths, there is also a list considered for flames for storing the fitness function. It is given as

\[ PF = \begin{bmatrix} PF_{1} \\ PF_{2} \\ \vdots \\ PF_{n} \end{bmatrix}. \]  

(16)

It is noteworthy here that moths and flames are both considered to be the solution [30]. They differ in the way they are considered and updated for each iteration. Moths are actually the search particles that travel in the search space, while flames are their optimal positions. Thus, each moth finds an optimal position and updates it once a better position is found. In this case, a moth is always at its optimal position.

3.3. Position Control Optimization (PCO). Nature has always been an inspiration for humans as it provides solutions to many problems. There have been developed several natural heuristic algorithms in the past for resolving real-world issues. However, since some heuristic algorithms have not been able to deliver satisfactory results, they are combined with other algorithms, in order to optimize the performance. Developing such hybrid algorithms has been accepted globally and is applied to solve several problems. PSO algorithm has been applied to recurrent neural/fuzzy network.

Related to other algorithms, PCO is also a population-based iterative process that is derived from the atmosphere. It is based on the wind blowing phenomena in the atmosphere for balancing the pressure such that wind blows at a particular speed from high-pressure area to low-pressure area [31, 32]. This technique provides precise outcome for estimating the atmospheric motion in the Lagrangian description [33, 34].

\[ \rho \ddot{\mathbf{a}} = \nabla \mathbf{P}. \]  

(17)

In equation, \( \ddot{\mathbf{a}} \) refers to the acceleration, \( \rho \) denotes the density of air for a particle, and \( \mathbf{F} \) refers to the forces that act on the air parcel. Ideal gas equation between air pressure \( P \), density \( \rho \), universal gas constant \( R \), and temperature \( T \) is

\[ P = \rho RT. \]  

(18)

It is imperative to note here that the movement of air occurs because different forces are gravitational force \( \mathbf{F}_{G} \), Coriolis force \( \mathbf{F}_{C} \), friction force \( \mathbf{F}_{F} \), and pressure gradient force \( \mathbf{F}_{PG} \).

\[ \mathbf{F}_{G} = \rho \delta V \mathbf{g}, \]

\[ \mathbf{F}_{PG} = -\nabla P \delta V, \]

\[ \mathbf{F}_{C} = -2\Omega \times \mathbf{u}, \]

\[ \mathbf{F}_{F} = -\rho \alpha \mathbf{u}. \]  

(19)

In the above equations, \( \delta V \) refers to the volume of air, \( \mathbf{g} \) is the gravitational acceleration, \( \Omega \) indicates the Earth’s rotation, \( \nabla P \) is the pressure gradient, \( \alpha \) denotes the coefficient of friction, and \( \mathbf{u} \) indicates the velocity of air.
These forces can be summed and described through the following equation:

\[
\rho \frac{\Delta \vec{u}}{\Delta t} = (\rho \Delta \vec{g}) + (-\nabla P \delta V) + (-2\Omega \times \vec{u}) + (-\rho a \vec{u}). \tag{20}
\]

Simplifying the equation, \(\Delta \vec{u}\) is rewritten as \((\Delta u/l)\) where \(\Delta t = 1\) and \(\delta V = 1\). Substituting these values in the equation, we get

\[
\rho \Delta \vec{u} = (\rho \Delta \vec{g}) + (-\nabla P) + (-2\Omega \times \vec{u}) + (-\rho a \Delta \vec{u}). \tag{21}
\]

The equation for density can be simplified and written as

\[
\Delta \vec{u} = \Delta \vec{g} + \left(-\nabla P \frac{RT}{P_{\text{pres}}}\right) + \left(-2\Omega \times \vec{u} \frac{RT}{P_{\text{pres}}}\right) + (-a \Delta \vec{u}). \tag{22}
\]

In the previous equation, \(P_{\text{pres}}\) refers to the present position. In the PCO technique, speed and position are considered to be altering at every iteration. Hence, \(\Delta \vec{u}\) can be assumed to be as \(\Delta \vec{u} = \vec{u}_{\text{new}} - \vec{u}_{\text{pres}}\).

\(\vec{u}_{\text{new}}\) refers to the new velocity or the velocity observed in the next iteration, while \(\vec{u}_{\text{pres}}\) refers to the velocity in the present iteration.

The equation is further simplified as follows:

\[
\vec{u}_{\text{new}} = (1 - \alpha) \vec{u}_{\text{pres}} - g\vec{l}_{\text{pres}}
\]  
\[
+ \left(\frac{RT}{P_{\text{pres}}}\right) (P_{\text{opt}} - P_{\text{pres}}) (x_{\text{opt}} - x_{\text{pres}}) \tag{23}
\]  
\[
+ \left(-2\Omega \times \vec{u} \frac{RT}{P_{\text{pres}}}\right).
\]

Substitute the Coriolis force \((\Omega \times \vec{u})\) with speed influence of other dimension \(\left(\vec{u}_{\text{pres}}\right)\), and represent all the coefficients with \(\gamma = -2RT\). S also observed that at certain points the pressure is intensely higher, making the updated values of speed useless and reducing the efficacy of the PCO algorithm. To alleviate this situation, the values of pressure are substituted by rank for all the particles. The final equations obtained for position and velocity are as follows. Figure 3 explains the concept of position control optimization for variable pressure.

\[
\vec{u}_{\text{new}} = (1 - \alpha) \vec{u}_{\text{pres}} - g\vec{l}_{\text{pres}} + \left(\frac{RT}{P_{\text{pres}}}\right) (1 - \frac{1}{n})(x_{\text{opt}} - x_{\text{pres}}) \tag{24}
\]  
\[
+ \left(\frac{\mu_{\text{other dim}}}{P_{\text{pres}}}\right).
\]

\[
\vec{l}_{\text{new}} = \vec{l}_{\text{pres}} + (\vec{u}_{\text{new}} \times \Delta t). \tag{25}
\]

In these equations, I refers to the rank and \(\vec{l}_{\text{new}}\) indicates to the new position.

Contrary to other algorithms, PCO requires less number of parameters for controlling the system.

The superiority of PCO to PSO and MFA, simulation has been executed using benchmark functions. These functions are expressed as equations. The configurations of the following benchmark functions are given in Table 2.

**Figure 3: Flowchart for PCO technique.**

<table>
<thead>
<tr>
<th>Table 2: Benchmark function.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
</tr>
<tr>
<td>(f_a)</td>
</tr>
<tr>
<td>(f_b)</td>
</tr>
<tr>
<td>(f_c)</td>
</tr>
<tr>
<td>(f_d)</td>
</tr>
</tbody>
</table>

\[
f_a(x) = \left(\sum_{i=1}^{m} \left(\sum_{j=1}^{i} x_{kj}\right)^2\right)^{\frac{1}{2}},
\]

\[
f_b(x) = -\sum_{i=1}^{8} \left((k - a_i)(k - a_i^T) + Ci\right) - 1,
\]

\[
f_c(x) = \sum_{i=1}^{7} \left[\frac{a_i - k_i(b_i^2 + b_jk_j)}{b_i + b_jk_j + k_i^2}\right]^2,
\]

\[
f_d(x) = -\sum_{i=1}^{4} C_i \exp \left(\left[-\sum_{j=1}^{3} a_{ij}(k_j - p_{ij})\right]^2\right).
\]
4. Fuzzy PI Controller Design

The approach proposed for this system has been implemented here in a step-by-step manner so as to ensure optimal efficacy and control results. In order to clearly explain each step of the controller design’s significance, the simulation and results have also been presented in Figure 4. The PI controller has been analyzed in the initial step and then followed by a fuzzy PI controller. Then, PCO technique has been used, followed by a brief discussion of the proposed approach. A snippet of the approach developed for this work is given in subsequent sections.

4.1. Conventional PI Controller. Any controller’s main task is to reduce the steady-state error, indicated by the following equation:

\[ e(t) = \omega SP(t) - \omega PV(t). \]  \hspace{1cm} (27)

![Figure 4: Schematic diagram of the PI controller.](image)

![Figure 5: Speed vs time.](image)

![Figure 6: Electromagnetic torque vs time.](image)

![Figure 7: Torque vs time.](image)
In the previous equation, $\omega_{SP}(t)$ refers to the reference speed, which is a function of time and $\omega_{PV}(t)$ is the real speed. The term $e(t)$ denotes the error function.

4.1.1. Simulation Results of the Conventional PI Controller.
The simulation results obtained for different parameters of a BLDC motor are given here. From Figure 5, it can be clearly observed that the time consumed by a PI controller to reach the desired speed value is higher, and also, the desired output is not steady or constant. A similar variation is observed for other parameters as well and is illustrated through the graphs given in Figures 6 and 7.

4.2. Fuzzy PI Controller. Fuzzy controller is actually a controller that is based on fuzzy logic, i.e., certain rules and conditions on the basis of which output is given. The rules are described in simple way; it can be easily understood by humans, helping them design a system easily if they have the proper knowledge. Figure 8 illustrates the fuzzy PI controller, and the inputs or terms used for controlling a fuzzy system are mapped with certain values [35]. These values are known as fuzzy sets. Majority of the fuzzy controllers use two or more than two fuzzy sets. It is noteworthy that higher the number of fuzzy sets used in a controller, higher are the stability and consequently optimal performance. However, with the increase in the number of fuzzy sets, the system developed also becomes highly problematic.

The gain parameters influence the characteristics of the system in a better way such that there is reduction in the rise time and increase in the overshoot and settling time.

4.2.1. Simulation Results of the Fuzzy PI Controller. Figure 9 shows the performance of the BLDCM using a fuzzy PI controller. It is contrary to the classical approach that fuzzy PI controller produces a better response. The rise time has decreased to a larger extent, contrary to the conventional approach. Also, there can be observed significantly lesser oscillations, or in other words, the range of oscillations is smaller. The figure clearly illustrates that using a fuzzy PI controller can maintain the speed at a constant value even under dynamic load conditions.

Figure 10 shows the variation in motor voltage through the various controllers. This voltage is the output obtained when an error in speed and an error in the change in speed are processed. It is noteworthy here that with the increase in speed, the voltage also increases. It can also be observed that the voltage is uniform with fewer ripples and oscillations.

Apart from voltage, here, the variation in torque has also been estimated. It can be observed from Figure 11 that torque is significantly lower. When the voltage is near stability with lesser oscillations, any increase in the torque can cause the speed to decrease. Thus analyzing these conditions, it can be asserted that the fuzzy PI controller optimizes the torque, tuning it to a nominal value.

The fuzzy PI controller proposed in this section uses different sets, NL, NM, NS, Z, PS, PM, and PL. These sets are given in Tables 3 and 4.

As shown from the table, there are two inputs $E$ and $E_C$, which can be described by the following equation:

$$E(N) = \omega_{SP}(N) - \omega_{PV}(N),$$

where $N = 1, 2, 3, \ldots, \text{etc.}$.
In the previous equation, $E(N)$ refers error at sample $N$. Similarly, $\omega PV(N)$ refers to the real speed of motor and $\omega SP(N)$ refers to the reference speed.

Change in error is given as

$$EC(N) = E(N) - E(N-1). \quad (29)$$

In equation (5), $E(N-1)$ refers error at the sample $(N-1)$ and $E_{C}(N)$ refers to change in error.

The membership function transforms the degree of fuzziness into a normalized interval $(0, 1)$. The triangle membership functions are used, since they are easier to interpret the concept. These membership functions enable the graphical representation of the fuzzy set. The values of the membership function are used to implement the mentioned surface more appropriate with the behavior of system shown in Figure 12.

Figures 13 and 14 show $E$ and $EC$ take the values between $-5000$ and $5000$ and it is measured in seconds for evaluating the system performance compared with existing one.

<table>
<thead>
<tr>
<th>Table 3: Rule base-$K_p$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$E$</th>
<th>NL</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>PL</td>
<td>PL</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>NM</td>
<td>PL</td>
<td>PL</td>
<td>PM</td>
<td>PS</td>
<td>PS</td>
<td>Z</td>
<td>NS</td>
</tr>
<tr>
<td>NS</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>Z</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>Z</td>
<td>PM</td>
<td>PM</td>
<td>PS</td>
<td>Z</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
<td>Z</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
</tr>
<tr>
<td>PM</td>
<td>PS</td>
<td>Z</td>
<td>NS</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NL</td>
</tr>
<tr>
<td>PL</td>
<td>Z</td>
<td>Z</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NB</td>
<td>NL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4: Rule base-$K_i$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$E$</th>
<th>NL</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>NM</td>
<td>NL</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>NS</td>
</tr>
<tr>
<td>NS</td>
<td>NL</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td>Z</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>PS</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td>PM</td>
<td>Z</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PL</td>
<td>PL</td>
</tr>
<tr>
<td>PL</td>
<td>Z</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

$E_{C}(N) = E(N) - E(N-1).$
E and EC are represented as input parameters. The time response of the error signals is used to depict the necessary knowledge associated with the system’s output parameters. In various cases, the parameters are related to fuzzy membership functions and are defined in random manner. Based on the performance measure, membership function parameters modify the behavior of the controller. These parameters help to get an optimum performance of the system.

The closer the error signals are to zero, the more the output signals are close to reference values. Thus, the error signals are considered sources of information, for developing the rule base for a fuzzy logic controller. The output obtained from this controller is the modification that needs to be made to increase or decrease the system’s overall control action. Fuzzy membership for the $K_p$ and $K_i$ takes the values between zero to 20 mapped along the width and degrees of membership functions based on the results. Use of narrower membership functions performs better response, larger overshoot, and quick settling time.

The width and degrees of membership functions are based on the results. Use of narrower membership functions results in a faster response (smaller response time). Larger oscillation, overshoot, and settling time appear when narrower membership functions are used. In various cases, the parameters are related to fuzzy membership functions defined in random manner. Based on performance measure, the membership function parameters modify the behavior of the controller. These parameters are led to get an optimum performance of the system. The respective rule viewer for these membership functions is shown in Figures 15 and 16 to know about the system’s stability.

Figure 13: Membership function $E$ ($-5000$ to $5000$).

Figure 14: Membership function $E_C$ ($-5000$ to $5000$).

Figure 15: Membership function $K_p$ (0–20).
5. Position Control Optimization (PCO) Technique

A brief description of PCO technique has been discussed in Section 3. From final iterative equation (24), PCO required less number of parameters to control the system. Due to its advantages over other classical approaches, it is widely exploited by the academic community. The system responses have been characterized with following parameters.

5.1. Simulation Results of Position Control Optimization

Performance of the motor using PCO is depicted through Figure 17. It is noticed that in comparison with previously
mentioned techniques, PCO offers highly optimized speed control. It can also be seen that the ripples are very low, exhibiting higher stability.

Similar to the highly efficient speed control, this technique also offers exceptional voltage control. As mentioned previously, the variation in speed is proportional to supply voltage. Hence, better the voltage control, better is the speed control. It can be observed from Figure 18 that the voltage is continuous, and the ripples are highly reduced.

Figure 19 shows the torque control exhibited by PCO. The oscillations are highly reduced, and the motor’s maximum torque maintains a constant pace, which enhances the rotor’s performance.

5.2. Parameter Setting. The parameter setting for the PCO algorithm can be represented in Table 5. The authors have conducted extensive research for the parameters of PCO, and the parameters of the optimization algorithms are selected as follows.

The approach proposed in this work employs a fuzzy PI controller that uses the PCO technique to optimize the performance and generate optimized results. It has been observed in the previous sections that a PI controller produces relevant results, but the efficacy demonstrated by fuzzy PI control was much better. Figure 20 shows the PCO technique performed more optimized results. Thus considering the results produced by each of these techniques, all the techniques are integrated and applied, i.e., fuzzy PCO-PI control. This technique delivers highly optimized performance and better results.

5.2.1. Simulation Results of Fuzzy PCO-PI Control. It can be noticed from Figure 21 using the approach developed in this work has optimized the results to a larger extent. It can be noticed that the ripples are almost negligible and also the rise time is very less. Contrary to the PCO approach used in the previous section, using the fuzzy PCO-PI controller approach has given better results.

The figure given below shows the position control of the BLDC motor for reference and the proposed approach.

6. Results and Discussion

In this section of the work, a comparative analysis has been made between the conventional approaches used for position control and the proposed approach. It can be observed from Figure 22 that the settling time for classical PI controller is more than the fuzzy PI control, PCO, and the proposed approach.

Also, it is noticed that the performance exhibited by the proposed approach is highly optimized than that of the
other approaches. The rise time in the case of the proposed approach is very small, toward stability and very small value of oscillations. Additionally, the proposed controller has almost negligible settling time with no overshoot at all. It can be asserted from the graphical illustrations that fuzzy PI-based PCO is by far the best and robust approach than contemporary approaches. Thus, it can be affirmed that it is an effective solution for different optimization solutions.

Different control techniques are characterized and carried out in Table 6 for concluding the performance of BLDCM. While using the PI controller, the response of the system has less ripple, high oscillation, and longer settling time. In fuzzy PI controller, the system responses are high ripples and quick settling time. In contrary to these techniques, the PCO algorithm gives same settling time as fuzzy PI controller but the system has less oscillation. From analysis, the response of the system using fuzzy PCO-PI controller gives more optimal results. Here, the red color characteristics curve has low ripples and takes lesser time to reach settling point. From the overall analysis, fuzzy PCO-PI control techniques yield optimum results than other conventional techniques. Results are compared with the published MFA results.

**Table 6: Performance analysis of different controllers.**

<table>
<thead>
<tr>
<th>Controllers</th>
<th>Rise time (sec)</th>
<th>Overshoot (%)</th>
<th>Peak</th>
<th>Settling time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCO</td>
<td>0.002</td>
<td>5</td>
<td>0.05</td>
<td>0.04</td>
</tr>
<tr>
<td>PI</td>
<td>0.008</td>
<td>13</td>
<td>0.13</td>
<td>0.1</td>
</tr>
<tr>
<td>Fuzzy PI</td>
<td>0.002</td>
<td>7</td>
<td>0.07</td>
<td>0.1</td>
</tr>
<tr>
<td>PCO-fuzzy PI</td>
<td>0.002</td>
<td>2</td>
<td>0.02</td>
<td>0.1</td>
</tr>
<tr>
<td>MFA</td>
<td>0.002</td>
<td>1</td>
<td>0.01</td>
<td>0.08</td>
</tr>
</tbody>
</table>

**Figure 22:** Comparative study of speed control.

7. Conclusion

In this article, an overview of the different speed control approaches for brushless DC motor drive has been discussed. With the motive of the proposed works, a review of the existing control techniques is analyzed with the proposed approach through different benchmark functions. The proposed algorithm (fuzzy PCO-PI) exhibits the ability to converge within the maximum number of iterations. From the results stated, the proposed model of fuzzy PCO-PI (position control optimization-proportional-integral) control techniques was compared with existing algorithms and proved their enhanced results. It affords better steady-state responses, reduced error, less oscillatory, and good speed regulation for various positions of rotor. The proposed algorithm is highly efficient in exploring optimal solutions, convergence speed, and accuracy which are significantly higher. Thus, the proposed approach is a highly effective and trustworthy, practical global optimization algorithm. The future scope is to analyze the performance of the brushless DC motor controller for variable DC input source using different battery models.

**Abbreviations**

BLDCM: Brushless DC motor  
PCO: Position control optimization  
PSO: Particle swarm optimization  
MFA: Moth-flame algorithm  
PFC: Power factor correction  
PI: Proportional-integral controller.

**Data Availability**

No data were used to support the study.

**Conflicts of Interest**

The authors declare that they have no conflicts of interest.

**References**


