International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 17, No. 1, March 2026, pp. 140∼154
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v17.i1.pp140-154

Modified firefly-optimized PI controller for BLDC motor performance under New European Driving Cycle conditions

Dibyadeep Bhattacharya[1,2], Raja Gandhi[3], Chandan Kumar[4], Karma Sonam Sherpa[2], Rakesh Roy[1]
[1]Department of Electrical Engineering, NIT Meghalaya, Shillong, India
[2]Electrical and Electronics Engineering, Sikkim Manipal Institute of Technology, Sikkim Manipal University (SMU), Majitar, India
[3]Department of Electrical Engineering, SCE Supaul, Bihar, India
[4]Department of Electronic and communication Engineering, SCE Supaul, Bihar, India

Article Info

Article history:

Received May 23, 2025
Revised Dec 26, 2025
Accepted Jan 23, 2026

Keywords:

Brushless direct current
Modified firefly algorithm
New European Driving Cycle
Particle swarm optimization
PI tuning

ABSTRACT

This paper presents the application of a modified firefly algorithm (MFA) for tuning the proportional-integral (PI) speed controller of a brushless direct current (BLDC) motor drive, targeting improved overall dynamic performance of the motor drive system for electric vehicle (EV) applications. The controller's effectiveness is evaluated under two variants of the New European Driving Cycle (NEDC) to replicate real-world driving scenarios. To validate the effectiveness of the proposed approach, a comparative study is carried out with two widely used optimization techniques, such as the standard firefly algorithm (FA) and particle swarm optimization (PSO). Comparative analysis reveals that the MFA-tuned controller delivers superior speed tracking accuracy, with significantly reduced speed error, speed ripple, and copper losses, when compared to controllers optimized using the standard firefly algorithm (FA) and particle swarm optimization (PSO). These improvements enhance both the energy efficiency and operational stability of the motor drive. Furthermore, the result of the experiment shows that the proposed controller demonstrates strong adaptability under varying load and speed conditions, positioning it as a robust solution for both electric vehicles and industrial motor control applications.

Corresponding Author:

Raja Gandhi
Department of Electrical Engineering, SCE Supaul
Bihar 852131, India
Email: rajagandhi1991@gmail.com

1. INTRODUCTION

The growing global emphasis on reducing greenhouse gas emissions and transitioning towards sustainable transportation has accelerated the adoption of electric vehicles (EVs). At the heart of EV propulsion systems lies the electric motor, which directly influences vehicle performance, energy efficiency, and driving range. Among the various motor technologies, brushless direct current (BLDC) motors have gained prominence due to their high efficiency, compact size, low maintenance, and favourable torque-speed characteristics. These attributes make BLDC motors well-suited for a wide range of EV applications, from two-wheelers to passenger cars [1]-[3]. However, the performance of a BLDC motor is heavily dependent on the effectiveness of its control strategy. Conventional proportional-integral (PI) controllers are widely employed for current and speed regulation due to their simple structure and ease of implementation. Nonetheless, fixed-gain PI controllers

often struggle under dynamic load conditions and varying driving cycles, leading to suboptimal performance, particularly in terms of energy consumption and transient response [4]-[7]. In the literature, various techniques have been employed to tune PI controllers, with manual or conventional tuning methods being the most common. However, these approaches often fall short when applied to complex and dynamic systems such as electric vehicle (EV) motor drives. Their inability to adapt to varying load conditions and system nonlinearities can lead to suboptimal performance, reduced efficiency, and poor control accuracy [8]-[10]. For instance, in [11], a nonlinear model predictive control (NMPC) approach is introduced for BLDC motor control in EVs, targeting the reduction of over-excitation, which causes unnecessary energy loss. By incorporating a nonlinear motor-load model, the NMPC optimizes voltage inputs to minimize current and torque errors, enhancing efficiency and torque response under changing loads. A BLDC motor-based EV model using real-time driving cycles was developed to evaluate PID, intelligent, hybrid, and adaptive supervisory controllers [12]. The adaptive supervisory controller achieves the best results in energy efficiency and driving range, despite its dependence on training data.

Building on these advancements, numerous optimization techniques based on artificial intelligence (AI) have been implemented to further enhance controller performance. Techniques such as fuzzy logic, neural networks, genetic algorithms (GA), particle swarm optimization (PSO), whale optimization, grey wolf optimization, ant optimization, and the BAT algorithm have shown promise in fine-tuning controller parameters and improving system stability and responsiveness [13]. However, each of these optimization techniques have their own pros and cons. As in [14], a comparative analysis of tunning proportional-integral (PI) controller for controlling the speed regulation of a brushless DC (BLDC) motor using particle swarm optimization (PSO) and whale optimization algorithm (WOA) has been presented. The objective is to reduce transient response time and enable faster attainment of the desired speed in a closed-loop control system. A mathematical model of the BLDC motor has been developed and simulated in Simulink, with the controller parameters (Kp and Ki) optimized using MATLAB implementations of both PSO and WOA. The results indicate that the PI controller tuned with WOA performed better than its PSO counterpart by achieving optimal gain values in a very few iterations and with reduced computation time. Though WOA is known for its straightforward yet effective search methods that enable fast identification of optimal solutions, WOA, like many swarm intelligence techniques, struggles with issues such as premature convergence, limited population diversity, and the risk of getting trapped in local optima [15]. Another study [16] presents the design of a buck-boost converter-fed BLDC motor drive and evaluates the motor's behavior under PI, fuzzy logic, hybrid PI-fuzzy, and grey wolf optimization (GWO)-based controllers. Key parameters such as speed, torque, overshoot, settling time, rise time, and steady-state error are analyzed, along with power quality factors like total harmonic distortion (THD), crest factor, and power factor. Simulation and modeling are done using MATLAB/Simulink and results have shown that GWO offers better overall performance with simpler computation compared to traditional techniques. GWO has shown strong performance in handling free optimization problems. However, when applied to constrained optimization problems, where the search landscape becomes more complex and irregular, this leader-driven strategy may lead to premature convergence. To counter this, introducing randomness into the search process can help maintain population diversity and avoid stagnation [17].

The result in [18] has applied neural approximation, friction compensation, and eccentricity control using simple activation functions in designing a novel nonlinear PI control method for nonlinear systems with unknown parameters and disturbances. Unlike traditional adaptive control, it ensures stability by driving error dynamics to a root of a perturbation function. The approach handles matched uncertainties and unknown control directions without high-gain feedback. Among the other evolving optimization techniques, the firefly algorithm (FA), introduced by Yang [19] in 2008, has gained attention for its effectiveness in solving complex optimization problems. Inspired by the natural flashing behavior of fireflies, FA relies on global communication among agents to efficiently explore the search space and identify both local and global optimal solutions. Due to its simplicity, robustness, and high convergence speed, FA has been successfully applied in various fields such as pattern recognition, neural network training, and clustering [20]-[22]. In the context of motor drive systems, FA has been employed to optimize the parameters of PI controllers used in controlling induction motors, permanent magnet synchronous machines, and BLDC motors. As BLDC motors are increasingly being used in automotive, aerospace, and industrial applications, achieving accurate and efficient control under fluctuating loads is essential [23], [24]. Despite its notable advantages, the traditional FA is not without limitations. One of the primary drawbacks is its relatively high computational demand, as it often requires a large number of iterations to achieve convergence. This extended computation time can be particularly problematic in real-time

or resource-constrained applications. Moreover, FA may suffer from premature convergence or get trapped in local optima, especially in high-dimensional or complex search spaces, which can limit its effectiveness in finding truly optimal solutions for PI controller tuning in motor drive systems [25]. These challenges have driven researchers to explore improved variants of FA or entirely new metaheuristic algorithms that can offer better convergence speed, enhanced global search capability, and reduced computational overhead.

In response to these challenges, this paper proposes a modified version of the firefly algorithm aimed at reducing computation time while maintaining effective performance. The key enhancement involves restricting the search space of the fireflies within a defined range, which helps to focus the optimization process and achieve faster convergence. This modified FA is tested on a BLDC motor drive system, where it is used to tune the PI controller parameters more efficiently. To evaluate the effectiveness of the proposed approach, the motor's performance is tested using the New European Driving Cycle (NEDC) [26], [27]. Two distinct scenarios are considered to assess the algorithm under varying operating conditions. The results are then compared against those obtained using two well-established optimization techniques, such as the standard firefly algorithm (FA) and particle swarm optimization (PSO), to demonstrate the superiority of the proposed method in terms of accuracy, response time, computational efficiency, and copper loss.

The structure of the paper is divided into seven key sections. The following section provides an overview of the BLDC motor control system. Section three introduces the mathematical formulation of the modified firefly algorithm. Section four outlines the algorithm flowchart in detail. Section five explains the New European Driving Cycle (NEDC), which is used to evaluate the controller's performance. Section six presents the analysis of the results obtained, and the final remarks and conclusions are also discussed in Section seven.

2. BLDC MOTOR CONTROL DRIVE

A vector control BLDC drives enables precise management of motor torque and speed by independently regulating the magnetic flux and the torque-producing currents. In this approach, a 1 kW BLDC motor is controlled using a speed controller driver circuit, specifically employing a PI-based controller as shown in Figure 1. This controller adjusts motor speed to match a reference command by monitoring and fine-tuning the current magnitudes. The difference between speed controller output and the actual current, measured via the current measurement block, is fed to the PI controller. The PI controller processes this error and generates a control signal that is then used to produce the inverter gate pulses required to effectively regulate the motor speed. The ability to achieve such improvements highlights the crucial role of precise tuning of the PI parameters in ensuring efficient and effective BLDC motor control, thus making utilization of optimization techniques in such systems quiet and offering benefits like increased efficiency, rapid dynamic response, and reduced copper loss.

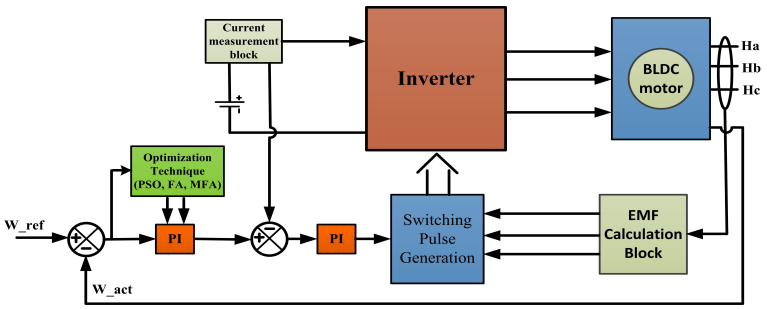


Figure 1. Block diagram for firefly algorithm-based vector control of BLDC motor drive

3. MATHEMATICAL FORMULATION OF MODIFIED FIREFLY OPTIMIZATION TECHNIQUES

3.1. Initial points calculation

The direct torque control of the BLDC motor drive needs appropriate parameters of the PI controller for proper operation of the motor. The proportional (K_p) and integral (K_i) gains must be within a limit for

running the motor in stable conditions. The lower and upper bound of K_p, K_i gain are termed as K_{p_lb}, K_{p_ub}, K_{i_lb} and K_{i_ub} respectively. From these upper and lower bound the middle points are calculated as given in (1).

$$\begin{aligned} K_{p_m} &= 0.5\,(K_{p_{ub}} + K_{p_{lb}}) \\ K_{i_m} &= 0.5\,(K_{i_{ub}} + K_{i_{lb}}) \end{aligned} \tag{1}$$

Using the upper, lower bound, and middle point of K_p, K_i, eight points are chosen at the boundaries of K_p and K_i as given in (2) which is shown in Figure 2. The optimized value of K_p and K_i will be in between the region of these boundaries. All these eight points are considered as the pop of the firefly optimization technique.

$$\begin{aligned} &\text{Point 1} = (K_{p_{lb}}, K_{i_{lb}}) \\ &\text{Point 2} = (K_{p_{lb}}, K_{i_{ub}}) \\ &\text{Point 3} = (K_{p_{ub}}, K_{i_{lb}}) \\ &\text{Point 4} = (K_{p_{ub}}, K_{i_{ub}}) \\ &\text{Point 5} = (K_{p_m}, K_{i_{lb}}) \\ &\text{Point 6} = (K_{p_m}, K_{i_{ub}}) \\ &\text{Point 7} = (K_{p_{lb}}, K_{i_m}) \\ &\text{Point 8} = (K_{p_{ub}}, K_{i_m}) \end{aligned} \tag{2}$$

The upper and lower bounds of each particle (K_p and K_i) can be estimated using (3) and (4) [20].

$$K_p = \frac{\sigma J}{1.5 P \phi_f} \tag{3}$$

$$K_i = \sigma K_p \tag{4}$$

Where, σ tuning coefficient, J is the rotor inertia, P is the number of poles and ϕ_f is flux linkage.

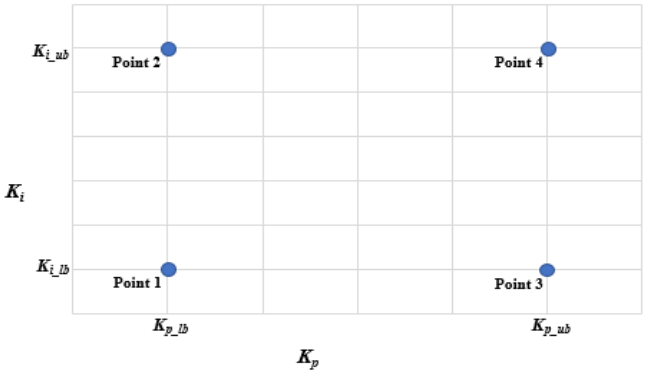


Figure 2. Boundaries of K_p and K_i

3.2. Determination of new K_p and K_i

Initially, the objective functions are calculated for first two pops (point 1 and point 2). These two points are marked as $Pointi$ [$K_p(i)$, $K_i(i)$] and $Pointj$ [$K_p(j)$, $K_i(j)$]. By comparing these two objective functions, the pop for higher objective function is modified by following (5)-(15). In the process, four parameters (α, $\beta 0$, θ and γ) are considered as constant. Here α is randomness strength which varies from 0 to 1 randomly. it controls the degree of randomness in firefly movement. Higher alpha (closer to 1) leads to more exploration of the search space while a lower alpha (closer to zero) leads to more exploitation (focusing on promising areas already discovered). $\beta 0$ is attractiveness constant. it determines the influence of a firefly attractiveness on the other fireflies. higher $\beta 0$ leads to stronger attraction, guiding it to a better result while lower $\beta 0$ leads

to weaker attraction, allowing more diverse exploration. γ is absorption coefficient. it controls the intensity of light absorption by other fireflies. Higher gamma means firefly are more influenced by other brighter fireflies leading to faster convergence, while lower gamma allows for more free movement in the search space. θ is randomness reduction factor. It governs how much the randomness $(alpha)$ is reduced over iterations. Higher theta leads to decrease in randomness allowing exploration in more promising area. Lower theta means higher degree of randomness throughout, which indicates continuation of the exploration. The $\alpha 1$ is calculated from alpha and theta as given in (5). Using the K_p, K_i value for $Pointi$ and $Pointj$, the xK_p and $x_K i$ are calculated using (6) and (7). The xK_p and $x_K i$ are the components of new K_p and K_i. From the upper and lower bounds, the $scaleK_p$ and $scaleK_i$ are calculated as given in (8) and (9).

β is calculated from $\beta 0$, γ and r which is given in (10), where r is obtained from $pointi$ and $pointj$ using (11).

$$\alpha 1 = \alpha\theta \tag{5}$$

$$xK_p = 0.5(K_p(i) + K_p(j)) \tag{6}$$

$$xK_i = 0.5(K_i(i) + K_i(j)) \tag{7}$$

$$scaleK_p = |K_{p\_ub} - K_{p\_lb}| \tag{8}$$

$$scaleK_i = |K_{i\_ub} - K_{i\_lb}| \tag{9}$$

$$\beta = \beta_0 e^{-\gamma r^2} \tag{10}$$

$$r = \sqrt{[K_p(i) - K_p(j)]^2 + [K_i(i) - K_i(j)]^2} \tag{11}$$

Finally, the new K_p and K_i are calculated using (14) and (15).

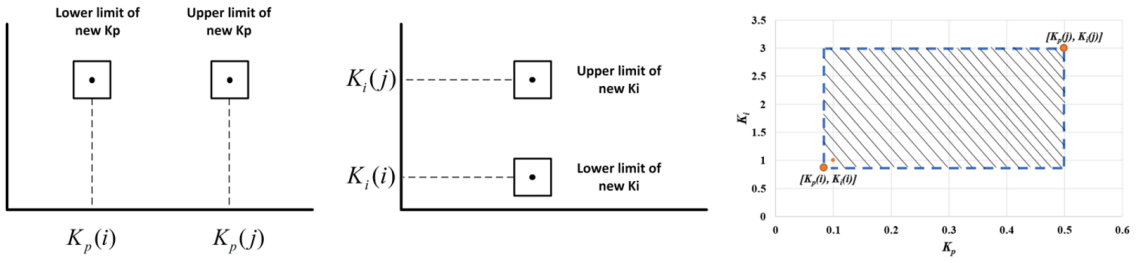
$$K_{p(new)} = K_p(i) + \beta[K_p(j) - K_p(i)] + stepsK_p \tag{12}$$

$$K_{i(new)} = K_i(i) + \beta[K_i(j) - K_i(i)] + stepsK_i \tag{13}$$

Here the $stepsK_p$ and $stepsKi$ of (12) and (13) are calculated using (14) and (15), where $randK_p$ and $randK_i$ are very important parameters. In conventional firefly algorithm, the $randK_p$ and $randK_i$ are considered any random values. So the selection of the new K_p and K_i has no selective area.

$$stepsK_p = \alpha 1(randK_p - xK_p)scaleK_p \tag{14}$$

$$stepsK_i = \alpha 1(randK_i - xK_i)scaleK_i \tag{15}$$



(a) (b)

Figure 3. Illustration of the proposed tuning process for the PI controller parameters: (a) determination of the upper and lower limits of new of K_p and K_i (b) selective search region in the K_p and K_i plane

In the modified firefly algorithm, the $randK_p$ and $randK_i$ are chosen within a range so that the new K_p and k_i must be within a selective area. For that limits of $randK_p$ and $randK_i$ are obtained by substituting

$K_{p(new)} = K_p(i)$, $K_{p(new)} = K_p(j)$ and $K_{i(new)} = K_i(i)$, $K_{i(new)} = K_i(j)$ into the update equations (12) and (13) respectively. The values obtained from these substitutions are then compared with (14) and (15), which provide the lower and upper limits of the gains. As a result, the bounds of $randK_p$ re obtained in (16) and (17), while the bounds of $randK_i$ are given in equations (18) and (19) as illustrated in Figure 3(a). The shaded region in the illustrative diagram Figure 3(b) represents the selective search space where the new parameters are chosen, ensuring a focused and efficient exploration in modified firefly algorithm (MFA).

$$randK_pl = \frac{\beta[K_p(i) - K_p(j)]}{\alpha 1 scale K_p} + \frac{[K_p(i) + K_p(j)]}{2} \tag{16}$$

$$randK_pu = \frac{[1-\beta][K_p(j) - K_p(i)]}{\alpha 1 Scale K_p} + \frac{[K_p(i) + K_p(j)]}{2} \tag{17}$$

$$randK_il = \frac{[1-\beta][K_i(j) - K_i(i)]}{\alpha 1 Scale K_p} + \frac{[K_i(i) + K_i(j)]}{2} \tag{18}$$

$$randK_iu = \frac{[1-\beta][K_i(j) - K_i(i)]}{\alpha 1 Scale K_i} + \frac{[K_i(i) + K_i(j)]}{2} \tag{19}$$

In conventional methods, $randK_p$ and $randK_i$ are chosen any random values. However by choosing the random values in a suitable manner will give the more optimized values. The random choosing of these values will search in the area of K_p and K_i randomly. However, by shrinking the searching area towards the best optimized value will give the better results in the performance of the motor drive.

4. FLOWCHART OF THE MODIFIED FIREFLY OPTIMIZATION TECHNIQUE

The flowchart of the modified firefly optimization technique is shown in Figure 4 and is explained below in step wise.

- Step 1 : All the parameters are initialized in first step. Here the α, β, θ and γ are selected as some constant values within their range. Eight bounce points are calculated from the upper and lower bound of K_p and K_i using (2). i, j and number of iteration (iter) are also initialized as 1.
- Step 2 : In this step, objective functions are calculated for point i and point j and they are termed as $F_x(p)$ and $F_x(q)$ respectively.
- Step 3 : Here $F_x(p)$ and $F_x(q)$ are compared. If $F_x(p)$ value is higher than $F_x(q)$, point i will be modified in step 4. If $F_x(q)$ value is higher than $F_x(p)$, no modification will take place and it will jump to step 6.
- Step 4 : This step gives the calculation of new K_p, K_i using (5) to (19) and the new objective function $F_x(r)$ is determined.
- Step 5 : The new objective function $F_x(r)$ is compared with $F_x(p)$ in this step. If $F_x(r)$ is less than $F_x(p)$, $F_x(p)$ is upgraded to $F_x(r)$ and the pop_i is upgraded with new K_p and K_i. In case $F_x(p)$ is less than $F_x(r)$, no modification will be done in $popi$ and it will go to step 6.
- Step 6 : In this step, j is upgraded to next value if it is not reached to maximum number of operands ($MAXPOP$) and it will go to step 2 again. If j reaches to maximum value, it will go to step 7.
- Step 7 : Here, i is upgraded to next value if it is not reached to maximum number of operands ($MAXPOP$). j is also initialized to 1 and again jump to step 2. In case i reaches to maximum value, it will go to step 8.
- Step 8 : Here number of iteration is checked whether it has reached to maximum value or not. If it does not reaches the maximum value, it will be increased to next number and jump to step 2. When it reaches the maximum value it will go to step 9.
- Step 9 : When all the iterations are completed, the best value (lowest objective function) is stored and the corresponding K_p and K_i are considered as optimized K_p and K_i.

In the first step initialization of the required parameters for the modified firefly algorithm is done, which is customized specifically for the objective of speed control in the BLDC motors. The problem dimensions of the problem search space which is denoted as D is set to 2, it reflects that two control parameters are essential for the speed regulation, which are proportional gain (Kp) and the integral gain (Ki). These parameters are chosen based on the BLDC motor characteristics to ensure that the problem objective are met. Nop denotes the number of population of the fireflies which is set to 8. This helps to create a balance between

computational complexity and exploration capability. The upper bounds (ub) and the lower bounds (lb) for the Kp and Ki are determined according to the motor parameters. This bounds helps the algorithm to find optimal value in which the motor can operate with effective speed control. To terminate the algorithm and find a suitable convergence point the maximum number of iteration (maxIter) is set to 2. There are some control parameters such as alpha, which are randomness strength which varies from 0 to 1 and it is highly random, beta0 which is attractiveness coefficient, gamma which is absorption coefficient and theta which is randomness reduction factor, these parameters are configured to influence the behavior of the fireflies, there values are selected such that the exploration and exploitation capabilities increases. Furthermore, the initial position of the firefly particles which is denoted as pop are such that it is under a defined bound, with 8 or 4 values at every corner such that all the points in a search space is covered, so that the algorithm does not always search in one place. The iteration count (iter) begins from 1 with loop indices as i and a counter j both their values starts at 1. With these the iterative process begins for optimizing the control parameters which is Kp and Ki.

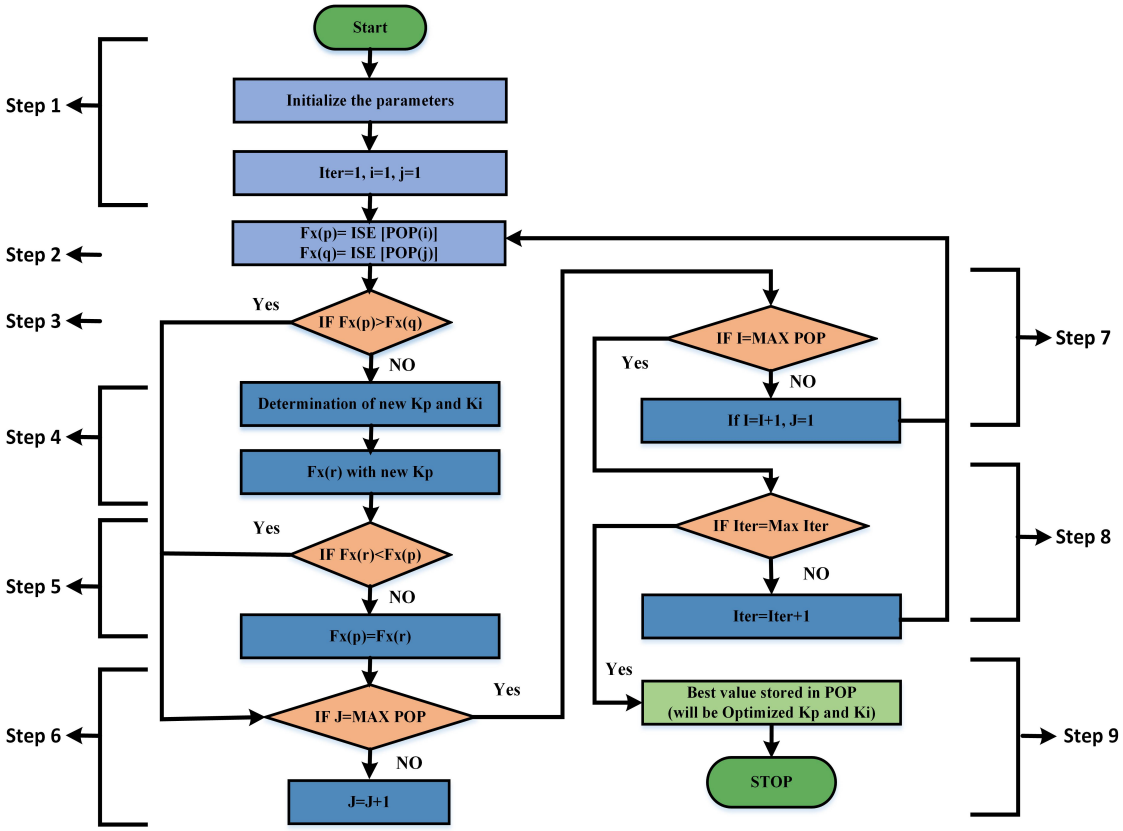


Figure 4. Flowchart of modified firefly optimization technique

4.1. Attractiveness and movement

The next step calculates the attractiveness and the movement of the firefly. It involves evaluating the objective function of the two positions of the fireflies at Fx(p) and Fx(q) for the generated population pop(i) and pop(j) respectively. The objective function Fx performs an integral time square error (ITSE) operation on each of the position of the firefly. The attractiveness of the firefly depends on the objective function values, the lower the ITSE value the higher will be the attractiveness. The firefly shifts its position to the more attractive firefly, this attractive get influenced by the randomness factor which is alpha. This step helps the firefly get attracted to the most promising value for effective speed control in BLDC motor.

4.2. Comparison between Fx(r) and Fx(p)

This step compares the fitness function of Fx(p) and Fx(q) for the current iteration (p) with the fitness function for the previous iteration (q). If the current fitness function is greater it means the firefly position

is good for speed performance of the BLDC motor, therefore the algorithm proceeds to step number 4. If the current fitness function is not greater (NO), it indicated the firefly position will not improve the speed performance therefore the algorithm moves to step 6.

4.3. Determination of new Kp and Ki

In this step, the new values of Kp and Ki are calculated based on the (5) to (19) and the new objective function $F_x(r)$ is determined.

4.4. Comparison between Fx(r) and Fx(p)

This step compares the fitness function of the new iteration (r) for the new values of Kp and Ki with the fitness function of the previous iteration (p). As the algorithm is defined to perform a minimization operation therefore if the new fitness function value is lower (YES) then the previous fitness function value it means the new Kp and Ki did improve the performance of the motor therefore it will assign the new value of kp and ki to the fitness function of Fx(p). If the new fitness function is higher, it indicates that the new values did not improve the performance so the algorithm shift to step 6.

4.5. Check whether the counter j reached the `max_pop` value

If the index j reaches its maximum value of pop it then checks goes to the next step to check the index i. if j is not equal to the maximum value of pop the algorithm increments the index j. If j is equal to maximum pop it indicates that all the possible values of Ki for the current Ki have been evaluated and explored, so the algorithm moves to step 7.

4.6. Check whether the counter i reached the `max_pop` value

If the index i reaches its maximum value of pop it then checks goes to the next step to check the index i. if i is not equal to the maximum value of pop the algorithm increments the index i. If i is equal to maximum pop it indicates that all the possible values of Kp for the current Kp have been evaluated and explored, so the algorithm moves to step 8.

4.7. Checking of algorithm stopping criteria

Check if the iteration counter has reached its maximum value i.e whether iter = Maxiter. If the iteration is equal to maximum iteration (YES) that means the algorithm has reached its limit so the algorithm moves to step 9 to stop the algorithm. If iteration is not equal to maximum iteration the algorithm increments the iteration and again starts the process from step 2.

4.8. End

This process ends the algorithm showing us the best values stored in pop, which will be the optimized Kp and ki values. At the end of the algorithm, the algorithm displays the convergence graph and also the cost function values.

5. NEW EUROPEAN DRIVING CYCLE (NEDC) FOR CONTROLLER PERFORMANCE EVALUATION

The New European Driving Cycle is a widely accepted standardized driving pattern developed to evaluate the energy efficiency and performance of vehicles under simulated conditions. It is designed to replicate real-world driving behavior, including both city traffic and highway scenarios. This cycle is particularly suitable for assessing electric vehicle (EV) performance, making it an ideal benchmark for validating motor controllers such as those used in BLDC motor drive systems.

The NEDC consists of two distinct parts: an urban driving segment and an extra-urban segment. The urban phase, known as the urban driving cycle (UDC), simulates low-speed city driving with frequent stops, starts, and idle periods. Covers a distance of approximately 4.052 kilometers over a duration of 195 seconds. This phase includes multiple acceleration and deceleration patterns to mimic congested traffic conditions, as shown in Figure 5(a). The extra-urban phase, called the Extra-Urban Driving Cycle (EUDC), represents high-speed driving typically seen on highways or open roads. It covers around 6.955 kilometers in 400 seconds, with speeds ranging up to 120 kilometers per hour which is nearly 2000 rpm as shown in Figure 5(b).

The NEDC is used as a dynamic input to evaluate the performance of an optimized BLDC motor controller. The controller parameters are tuned using the firefly algorithm to enhance the motor's efficiency, dynamic response, and minimize the copper losses. Applying the NEDC as the speed reference profile in

simulations enables the testing of the controller under varying operational conditions. This includes assessing its behavior during acceleration, deceleration, and steady-state phases, which are all integral parts of the NEDC.

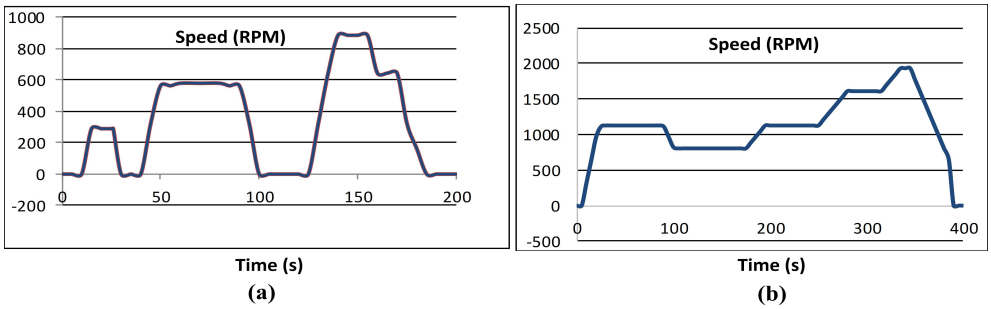


Figure 5. Vehicle speed profiles under New European Driving Cycle (NEDC): (a) low-speed urban driving segment and (b) high-speed highway segment

6. RESULTS AND DISCUSSION

The modified firefly algorithm (MFA) is implemented in MATLAB/Simulink as shown in Figure 6 with optimize PI speed control parameters for BLDC drive. The motor specifications used for the experiments are listed in Table 1. To evaluate the effectiveness of the MFA-based PI controller, its performance was compared with two other commonly used tuning methods such as standard firefly algorithm and particle swarm optimization (PSO) technique. To evaluate the performance of these optimize controller, the two key scenarios of New European Driving Cycle are considered and tested such as low-speed operation with frequent start-stop and high-speed driving with acceleration. In each case, the motor's speed response and copper loss are closely monitored and analyzed. In motor response the key parameter are speed ripple and speed error were measured to assess controller performance. The specifications and parameter settings of all the algorithms used in this study are summarized in Table 2.

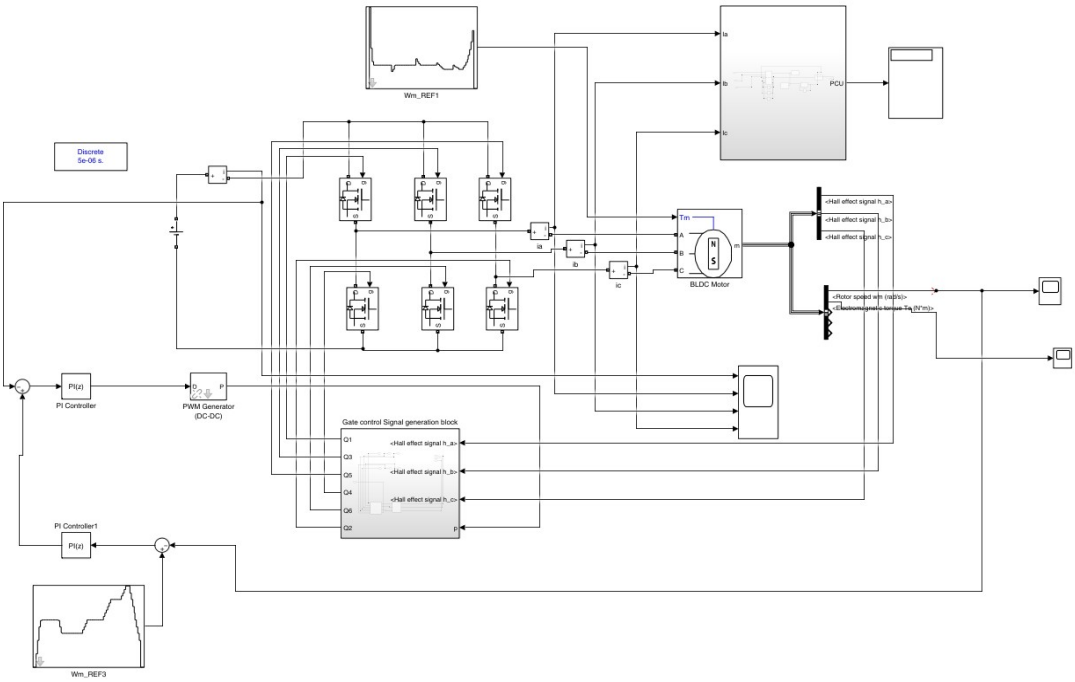


Figure 6. Simulation model of BLDC motor drive with optimized PI controller in MATLAB/Simulink

Table 1. BLDC motor specification

Parameters	Value	Unit
Rated power	1	kW
Rated voltage	48	Volt
Rated current	3	Amp
Rated speed	3000	rpm
Inductance	0.309	mH
Torque constant	0.33	Nm/A
Resistance	0.75	Ω

Table 2. Initial and additional parameters

Initial Parameters		Additional Parameters			
FA, PSO & MFA	Value	MFA	Value	PSO	Value
Particle size (n_{max})	15	γ	0.1–1	ω_{max}	0.9
Iteration (m_{max})	25	β_0	0.5–1	ω_{min}	0.4
K_p range	0–10	α	0–1	c_1, c_2	1.5
K_i range	0–8				

6.1. Case 1: Low-speed urban driving (stop-and-go conditions)

During the initial phase of the NEDC cycle (0–200 seconds), the vehicle operates at low speeds with frequent accelerations and decelerations, resembling typical city traffic conditions with variable input load as shown in Figures 7 and 8. This segment is critical for assessing how effectively the controller manages abrupt changes in speed and load. Three optimization techniques are applied to tune the speed controller in the NEDC cycle: PSO-based PI tuning, the firefly algorithm, and modified firefly algorithm-based PI tuning. The results of these three tuning methods, that is, speed and current response, are depicted in Figures 8(a)-8(c), respectively.

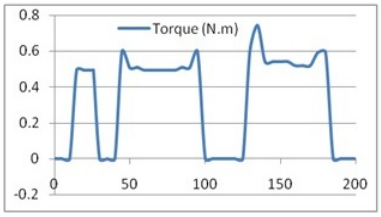


Figure 7. Variable load torque applied to low-speed urban drive

All three controllers track the reference speed accurately. However, from the observations, it is evident that the modified firefly algorithm provides superior performance compared to the other two conventional optimization techniques. The modified firefly-optimized BLDC controller exhibits quick responsiveness to frequent accelerations and braking, ensuring precise speed control throughout. Even with constant stops and starts, the motor runs smoothly without any noticeable jerks or delays. This demonstrates that the controller is highly suited for urban driving, offering enhanced control and a more comfortable driving experience in heavy traffic compared to the other two tuning methods. Table 3 presents additional parameters for comparison, where performance is evaluated based on speed ripple, speed error, and copper loss. These losses are calculated using (20)-(22). The results indicate that the modified firefly algorithm delivers the best overall performance.

$$\text{Speed Error} = \frac{|\omega_{ref} - \omega_{act}|}{\omega_{ref}} * 100 \tag{20}$$

$$\text{Speed Ripple} = \frac{|\omega_{max} - \omega_{min}|}{\omega_{avg}} * 100 \tag{21}$$

$$\text{Copper Loss} = (I_a^2 + I_b^2 + I_c^2) * R * Time \tag{22}$$

Where, ω_{max} and ω_{min} are maximum and minimum speed value during steady state speed.

Table 3. Comparison of parameters at stop-and-go conditions

Parameter	Experimental Result		
	PSO	FA	MFA
Speed error (%)	0.018	0.038	0.012
Speed ripple (%)	12.2	16.63	10.1
Copper loss (kWh)	0.281	0.273	0.265

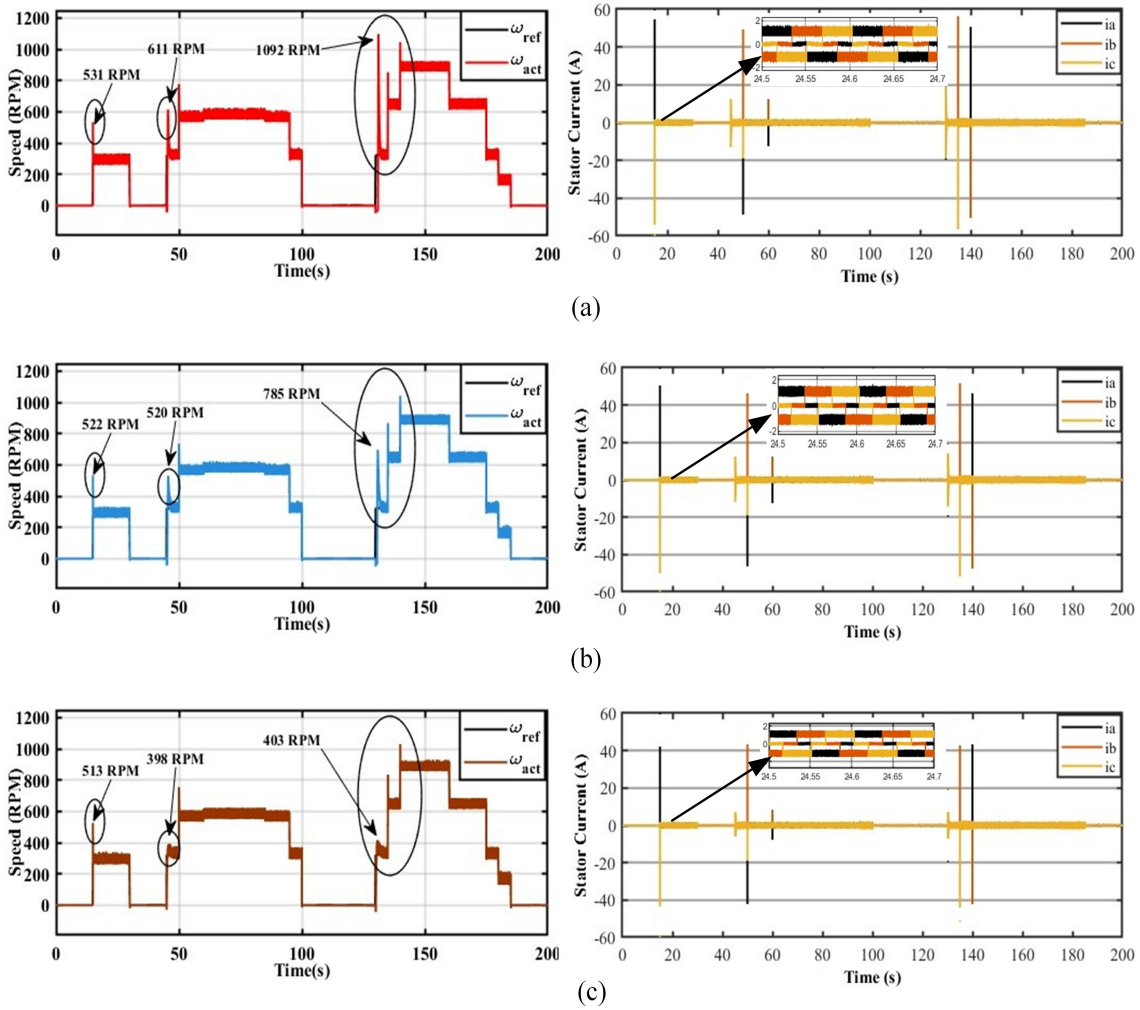


Figure 8. Speed and stator current response of the motor under the New European Driving Cycle using (a) PSO-optimized PI controller, (b) firefly optimized PI controller, and (c) modified firefly-optimized PI controller

6.2. Case 2: High-speed driving with acceleration (highway segment)

In the later stages of the NEDC cycle, the vehicle experiences higher speeds, sharp accelerations, and steady cruising, closely resembling highway or extra-urban driving conditions where load variations are continuous, as illustrated in Figure 9. This phase tests the controller's capability to handle high-speed operations and rapid changes in torque demand. The experimental results presented in Figures 10(a)-10(c) show that all the controllers accurately track the reference speed even under such dynamically changing load conditions.

Upon analyzing the graphs, it becomes clear that the controller optimized with MFA performs better than those tuned with PSO and FA techniques. Specifically, the MFA-based controller shows improvements in speed error, speed ripple, and copper losses, as summarized in Table 4. The reduction in copper loss significantly increases the efficiency of electric vehicles, extending the battery range and improving overall performance. In addition, the operation of the cooler motor improves reliability, prolongs the life of the components, and reduces the need for heavy thermal management. Overall, the controller demonstrates efficient performance during rapid acceleration phases and maintains a stable speed with minimal fluctuations. Its ability to quickly adapt to changing speed requirements ensures consistent and smooth operation, making it highly reliable for highway driving, where performance at higher speeds is crucial for both driver comfort and vehicle endurance.

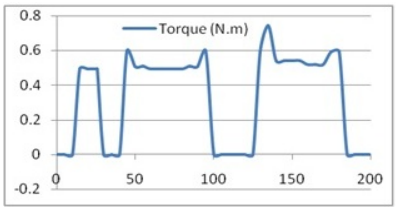
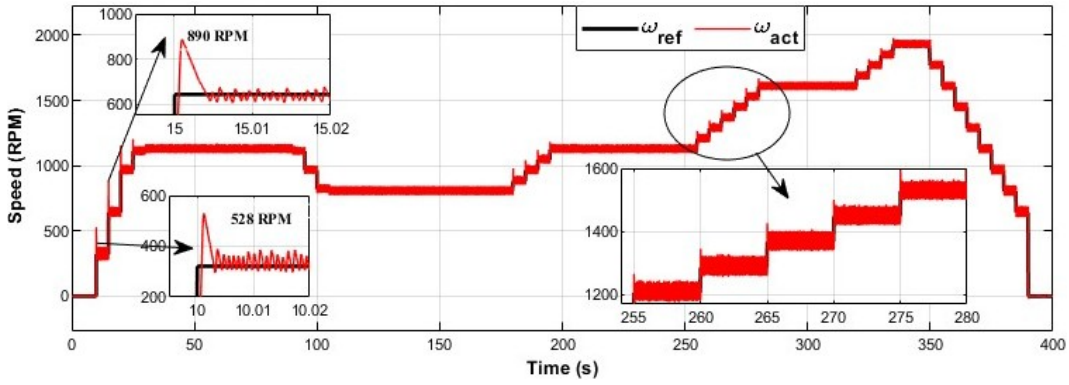
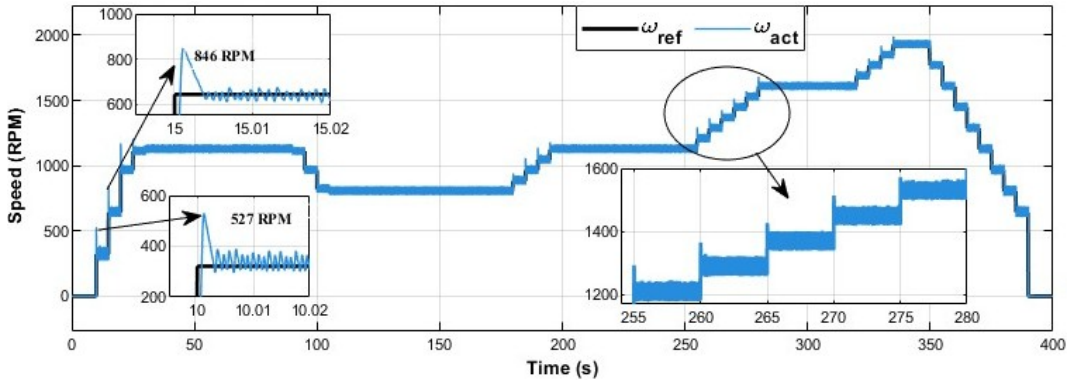


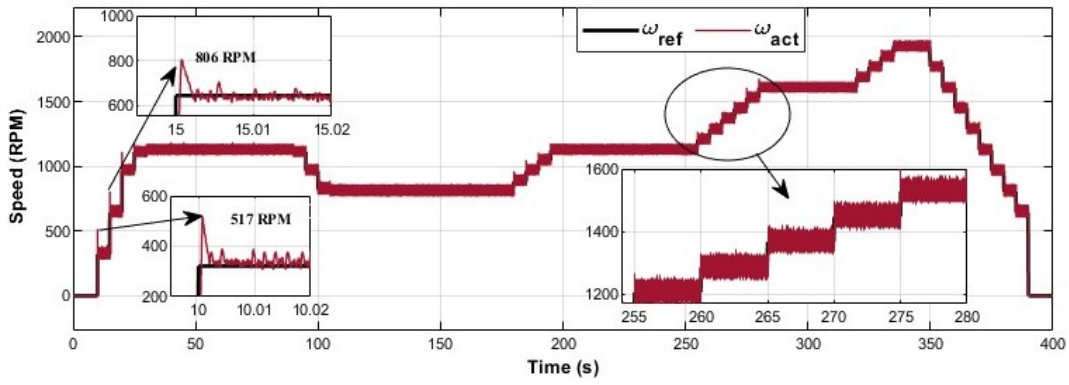
Figure 9. Variable load torque applied to low speed urban drive



(a)



(b)



(c)

Figure 10. Speed response under highway segment condition using (a) PSO-PI, (b) FA-based PI, and (c) modified FA-based PI

Table 4. Comparison of parameters at highway segment conditions

Parameter	Experimental result		
	PSO	FA	MFA
Speed error (%)	0.0038	0.0043	0.0014
Speed ripple (%)	3.91	4.24	3.76
Copper loss (kWh)	1.811	1.978	1.68

7. CONCLUSION

This paper presents the use of a modified firefly algorithm (MFA) to tune the PI speed controller of a BLDC motor drive. The controller's performance was evaluated under two different NEDC driving conditions to reflect real-world scenarios. The results clearly show that the MFA-tuned controller outperforms those optimized using standard firefly algorithm (FA) and particle swarm optimization (PSO). It achieves better speed tracking with noticeably lower speed error and ripple, along with reduced copper losses. These improvements not only enhance overall efficiency but also contribute to more stable and reliable motor operation. Given these strengths, the proposed controller is well suited for electric vehicle applications where smooth and energy-efficient performance is essential.Its ability to adapt to varying load and speed conditions also makes it suitable for broader industrial motor drive systems. However, the current study is limited to simulation-based validation. Future work will include experimental verification through hardware implementation, possibly using real-time platforms such as dSPACE or DSP controllers. Further, incorporating advanced hybrid or adaptive control strategies could offer additional performance gains under dynamic operating conditions.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Dibyadeep Bhattacharya	✓	✓	✓		✓				✓	✓				✓
Raja Gandhi		✓		✓			✓		✓	✓	✓	✓	✓	
Chandan Kumar	✓		✓				✓				✓		✓	
Karma Sonam Sherpa				✓			✓			✓		✓		
Rakesh Roy				✓	✓	✓	✓			✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

This published article contains all of the data created or examined during this investigation.

REFERENCES

[1] A. Zaineb, J. Naresh, S. A. Sydu, K. Chenchireddy, V. Jagan, and L. B. Ganesh, "Advancements in BLDC motor technology for sustainable electric mobility," in *2025 International Conference on Electronics and Renewable Systems (ICEARS),* IEEE, Feb. 2025, pp. 263–267, doi: 10.1109/ICEARS64219.2025.10941599.

[2] W. Cao, A. A. S. Bukhari, and L. Aarniovuori, "Review of electrical motor drives for electric vehicle applications," *Mehran University Research Journal of Engineering and Technology,* vol. 38, no. 3, pp. 525–540, 2019, doi: 10.22581/muet1982.1903.01.
[3] L. Alberti and D. Troncon, "Design of electric motors and power drive systems according to efficiency standards," *IEEE Transactions on Industrial Electronics,* vol. 68, no. 10, pp. 9287–9296, 2021, doi: 10.1109/TIE.2020.3020028.
[4] A. M. Jarkas and M. A. N. Doss, "Optimized PI controller tuning for improved performance in BLDC motor speed control using heuristic adaptive lyrebird optimization algorithm," *Electrical Engineering,* vol. 107, no. 7, pp. 9467–9487, 2025, doi: 10.1007/s00202-025-02984-1.
[5] Z. A. Al-Dabbagh and S. W. Shneen, "Design of a PID speed controller for BLDC motor with cascaded boost converter for high-efficiency industrial applications," *International Journal of Robotics and Control Systems,* vol. 5, no. 1, pp. 22–46, 2025, doi: 10.31763/ijrcs.v5i1.1601.
[6] B. M. Yilmaz, S. Unver, E. Selim, I. Saka, and E. Tatlicioglu, "Fuzzy logic based adaptive control of robot manipulators driven by BLDC motors," *Journal of the Franklin Institute,* vol. 362, no. 4, 2025, doi: 10.1016/j.jfranklin.2025.107528.
[7] A. Singh, S. Yadav, S. Sarangi, S. De, A. K. Singh, and R. K. Singh, "Optimizing BLDC motor performance: A study of PID, artificial neural network and fuzzy logic controllers," in *2025 International Conference on Innovation in Computing and Engineering (ICE),* IEEE, Feb. 2025, pp. 1–6, doi: 10.1109/ICE63309.2025.10984074.
[8] M. Ghasemi, S. kadkhoda Mohammadi, M. Zare, S. Mirjalili, M. Gil, and R. Hemmati, "A new firefly algorithm with improved global exploration and convergence with application to engineering optimization," *Decision Analytics Journal,* vol. 5, 2022, doi: 10.1016/j.dajour.2022.100125
[9] G. Shokri, E. Naderi, and M. Najafpour, "DTC based BLDC motor controlled centrifugal pump Fed by PI-BFO tuning strategy for buck-boost converter in solar PV array water pumping system," in *2019 10th International Power Electronics, Drive Systems and Technologies Conference (PEDSTC),* IEEE, Feb. 2019, pp. 769–774, doi: 10.1109/PEDSTC.2019.8697270..
[10] R. Jauhar, N. Ismail, and N. Sartika, "Design of torque controller based on field oriented control (FOC) method on BLDC motor," in *2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA),* IEEE, Oct. 2022, pp. 1–5, doi: 10.1109/TSSA56819.2022.10063889.
[11] P. Ubare and D. N. Sonawane, "Performance assessment of the BLDC motor in EV drives using nonlinear model predictive control," *Engineering, Technology and Applied Science Research,* vol. 12, no. 4, pp. 8901–8909, 2022, doi: 10.48084/etasr.4976
[12] P. Saiteja, B. Ashok, B. Mason, and P. S. Kumar, "Assessment of adaptive self-learning-based BLDC motor energy management controller in electric vehicles under real-world driving conditions for performance
[13] R. Gandhi, R. Wilson, A. Kumar, and R. Roy, "Comparative analysis of vector controlled PMSM drive with particle swarm optimization and ant colony optimization technique," in *2020 International Conference on Computational Performance Evaluation (ComPE),* IEEE, Jul. 2020, pp. 744–750, doi: 10.1109/ComPE49325.2020.9200184.
[14] S. Banerjee, S. S. Kumar, and A. Alam, "Whale optimization algorithm (WOA) based speed control of BLDC motor," in *2022 IEEE 2nd International Conference on Sustainable Energy and Future Electric Transportation (SeFeT),* IEEE, Aug. 2022, pp. 1–6, doi: 10.1109/SeFeT55524.2022.9909419.
[15] M. H. N. Shahraki, H. Zamani, Z. A. Varzaneh, and S. Mirjalili, "A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations," *Archives of Computational Methods in Engineering,* vol. 30, no. 7, pp. 4113–4159, Sep. 2023, doi: 10.1007/s11831-023-09928-7.
[16] B. M. Kumar and R. B. Ashok, "Soft computing using grey wolf optimization (GWO) for the performance improvement of high speed brushless DC motor," in *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR),* IEEE, Jul. 2018, pp. 1–6, doi: 10.1109/ICETIETR.2018.8529001.
[17] S. N. Makhadmeh *et al.,* "Recent advances in grey wolf optimizer, its versions and applications: review," *IEEE Access,* vol. 12, pp. 22991–23028, 2024, doi: 10.1109/ACCESS.2023.3304889.
[18] R. Ortega, A. Astolfi, and N. E. Barabanov, "Nonlinear PI control of uncertain systems: an alternative to parameter adaptation," *Systems and Control Letters,* vol. 47, no. 3, pp. 259–278, Oct. 2002, doi: 10.1016/S0167-6911(02)00212-8.
[19] D. Yang, Q. Chen, S. B. Su, P. Zhang, K. Kurosaka, R. R. Caspi, S. M. Michalek, H. F. Rosenberg, N. Zhang, and J. J. Oppenheim, "Eosinophil-derived neurotoxin acts as an alarmin to activate the TLR2–MyD88 signal pathway in dendritic cells and enhances Th2 immune responses," *Journal of Experimental Medicine,* vol. 205, no. 1, pp. 79–90, Jan. 2008, doi: 10.1084/jem.20062027.
[20] N. F. Johari, A. M. Zain, N. H. Mustaffa, and A. Udin, "Firefly algorithm for optimization problem," *Applied Mechanics and Materials,* vol. 421, pp. 512–517, 2013, doi: 10.4028/www.scientific.net/AMM.421.512.
[21] V. Kumar and D. Kumar, "A systematic review on firefly algorithm: past, present, and future," *Archives of Computational Methods in Engineering,* vol. 28, no. 4, pp. 3269–3291, 2021, doi: 10.1007/s11831-020-09498-y.
[22] S. Bazi, R. Benzid, Y. Bazi, and M. M. Al Rahhal, "A fast firefly algorithm for function optimization: Application to the control of BLDC motor," *Sensors,* vol. 21, no. 16, 2021, doi: 10.3390/s21165267.
[23] B. N. Kommula and V. R. Kota, "Direct instantaneous torque control of brushless DC motor using firefly algorithm based fractional order PID controller," *Journal of King Saud University - Engineering Sciences,* vol. 32, no. 2, pp. 133–140, 2020, doi: 10.1016/j.jksues.2018.04.007.
[24] I. Anshory, D. Hadidjaja, and R. B. Jakaria, "BLDC motor: modeling and optimization speed control using firefly algorithm," *Dinamik,* vol. 25, no. 2, pp. 51–58, 2020, doi: 10.35315/dinamik.v25i2.7851.
[25] E. Yilmaz, M. Artar, and M. Ergün, "Investigation of notch effect in the optimum weight design of steel truss towers via particle swarm optimization and firefly algorithm," *Frontiers of Structural and Civil Engineering,* vol. 19, no. 3, pp. 358–377, 2025, doi: 10.1007/s11709-025-1160-0.
[26] H. Chen, Z. Song, X. Zhao, T. Zhang, P. Pei, and C. Liang, "A review of durability test protocols of the proton exchange membrane fuel cells for vehicle," *Applied Energy,* vol. 224, pp. 289–299, 2018, doi: 10.1016/j.apenergy.2018.04.050.
[27] T. M. I. Mahlia, S. Tohno, and T. Tezuka, "A review on fuel economy test procedure for automobiles: Implementation possibilities in Malaysia and lessons for other countries," *Renewable and Sustainable Energy Reviews,* vol. 16, no. 6, pp. 4029–4046, 2012, doi: 10.1016/j.rser.2012.03.032.

BIOGRAPHIES OF AUTHORS

Dibyadeep Bhattacharya received the B.Tech. degree in electrical engineering from Durgapur Institute of Advanced Technology and Management in Rajbandh, West Bengal, India, in 2008, and the M.Tech. degree in power electronics and electrical drives specialization from the Sikkim Manipal Institute of Technology, SIKKIM, India, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Institute of Technology Meghalaya, Shillong, India. His research interests include electrical machine drives and power electronics. He can be contacted at email: p19ee002@nitm.ac.in.

Raja Gandhi has completed his B.Tech. in Electrical Engineering from NIT Sikkim in 2016, followed by an M.Tech. from NIT Meghalaya. He earned his Ph.D. from NIT Meghalaya, where his research focused on machine drives. Currently, he is an assistant professor in the Department of Electrical Engineering at Supaul College of Engineering, India. His primary research interests include the advancement and optimization of machine drives. He can be contacted at email: rajagandhi1991@gmail.com.

Chandan Kumar has earned a Ph.D. from IIT (BHU), Varanasi, a Postdoc from IIT Bombay, and a JSPS Fellow from NAIST, Japan. He is currently working as an assistant professor and Dean of Academics at Supaul College of Engineering, India. He has been a guest editor for Frontiers in Materials and is currently a member of the editorial board of IJPEDS. His current research interest includes flexible electronics, solar cells, photodetectors, sensors, and other electrical applications. He can be contacted at email: chandan.ece.mit@gmail.com or chandank.rs.ece14@iitbhu.ac.in.

Karma Sonam Sherpa has completed his B.E. (electrical engineering) from the MREC, Jaipur, and M.Tech. (power electronics and machine drives) from I.I.T., Kharagpur in the years 1996 and 2003, respectively. He holdsa a doctorate from Sikkim Manipal University. He has been serving Sikkim Manipal University since 1998 in various positions. Presently, he is the registrar at Sikkim Manipal University. He is a life member of ISTE, IEI, and the System Society of India. His areas of interest are electric power distribution systems, power electronics, and electrical drives. He has published various research papers in journals and conferences. He can be contacted at email: karma.sherpa@smit.smu.edu.in.

Rakesh Roy received the B.Tech. degree in electrical engineering from NIT Agartala, Tripura, India in 2009, the M.Tech. degree with specialization in power electronics and drives from NIT Agartala, Tripura, India, in 2011, and a Ph.D. degree from NIT Meghalaya, Shillong, India, in 2018. He is currently an assistant professor in the Electrical Engineering Department of NIT Meghalaya, Shillong, India. His research interests broadly include electrical machine drives and power electronics. He can be contacted at email: rakesh.roy@nitm.ac.in.