IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 4, April, 2026, pp. 1125-1139
Journal homepage: http://iieta.org/journals/jesa

Optimization and Control of a BLDC Motor for Improving Electric Vehicle Efficiency Using the Giant Armadillo Optimization Algorithm

Arwa Amer Abdulkareem[1*], Hasan Wahhab S. Rabee[2], Zaid KH Sadane[3]

[1] Electrical Engineering Technical College, Middle Technical University, Baghdad 10001, Iraq
[2] Najaf Technical Institute, Al-Furat al-Awsat Technical University, Najaf 54001, Iraq
[3] Department of Prosthetics and Orthotics Techniques, Polytechnic College Mosul, Northern Technical University, Mosul 41001, Iraq

Corresponding Author Email: arwa.amer@mtu.edu.iq

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/jesa.590422

Received: 11 January 2026
Revised: 30 March 2026
Accepted: 14 April 2026
Available online: 30 April 2026

Keywords:
Giant Armadillo Optimization algorithm, brushless direct current control, electric vehicles, proportional-integral control, fractional-order, particle swarm optimization

ABSTRACT

Brushless direct current (BLDC) motors are widely utilized in industrial and traction applications, particularly in electric vehicles (EVs), due to their inherent advantages. In EVs, a very low steady-state error with rapid convergence responses is required for a smooth wheel acceleration. This article proposes a robust and fast speed controller for the BLDC-driven EV using optimized proportional-integral (PI) and fractional-order (FOPI) controllers tuned by a new bio-inspired optimization algorithm called the Giant Armadillo Optimization (GAO) algorithm. The main goal of this study is to improve the performance of the BLDC motor under different variations of the reference speed. The suggested system was designed and tested using the MATLAB/Simulink environment. In this study, three different case studies are applied, including no load, sudden decrease in speed, and random speed variations. Moreover, the achieved results are compared with the classical PI and particle swarm optimization (PSO-PI) controllers. The obtained results show that the suggested GAO-FOPI controllers significantly outperform the conventional controllers, obtaining a speed overshoot of 2.13%, compared to 8.4% with the classical PI method. Finally, the suggested controller decreases the rise time to 1.02 sec, achieving a 41% enhancement when compared with PSO-PI and PI techniques (1.64 sec and 1.72 sec).

1. INTRODUCTION

1.1 Background and motivation

The rapid electrification of transportation change has pushed the requirement for effective and steady motor drive systems for the new electric vehicles (EVs). EVs should be able to ensure good dynamic performance with the lowest power losses possible [1, 2]. Today, the brushless direct current (BLDC) motor has been used in EVs due to its high efficiency, small size, high torque-to-weight ratio, and low maintenance. However, BLDC drives are highly nonlinear and thus very sensitive to noise and parameter changes, besides the strong coupling between torque and speed, as well as the sensitivity to parameter variations and external load fluctuations [3, 4]. Hence, the problem of finding a control system that is effective and capable of optimal speed and torque responses even under these uncertainties remains a major challenge in electric vehicle control systems. Therefore, due to their simple structure and easy implementation, traditional proportional–integrative–derivative (PID) controllers are generally employed in the speed control of BLDC [5-7]. Accordingly, a classical PID controller with fixed gains cannot satisfactorily deal with the nonlinearities of the system, changes in parameters, and dynamic loads that are typical for the operations of ATVs. In order to increase the rigidity and extend the capabilities, a fractional-order PID (FOPID) controller was proposed by several authors [8, 9]. These settings provide the system more freedom, which improves the frequency response, makes it less likely to be disturbed, and makes it more accurate in the steady state. The five parameters of the FOPID controller also affect how well the control system works. This system may be very sensitive to these values, and the search area is nonlinear and multimodal, which means that small changes in the parameters can lead to significant variations in the system's performance and stability.

As a result, smart optimization techniques are used for defining the best FOPID parameters in BLDC motor control [10].

1.2 Key challenges

Recently, a large number of metaheuristic algorithms mimicking nature have been suggested for solving motor control issues. Particle swarm optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA) are the most recognized examples of such algorithms [11-15]. While these algorithms have been successful in achieving their goals, they are still

prone to the problem of premature convergence, low exploitation capability in subsequent iterations, and stagnation at local optima when dealing with complex dynamic models. Therefore, it is necessary to develop more robust and adaptive optimization methods that not only can maintain a balance between exploration and exploitation over the course of the search but also can carry out calculations with less computational effort. Because of these limits, this study offers a fast-speed FOPI controller optimized by the new Giant Armadillo Optimization (GAO) algorithm. GAO is a nature-inspired metaheuristic that models the feeding and defensive activities of giant armadillos and, by incorporating adaptive learning and cooperative features, depicts the global search in the initial stages and refined local search near the solution. This algorithm was proposed in the study [14]. This inherent equilibrium enables GAO to efficiently traverse high-dimensional search spaces, thereby making it a very appropriate tool for the nonlinear and coupled dynamics of BLDC control systems. Here, the GAO method is used to perform automatic tuning of the five FOPID variables as well as the three PID parameters to minimize a composite performance index consisting of the integral time absolute error (ITAE) and integral time square error (ITSE), thus ensuring an optimal trade-off between transient response and steady-state precision.

1.3 Previous work

In the past three years, several algorithms have been proposed to tune the parameters of the FOPID and PID controllers to enhance the performance of the BLDC control system. A hardware trans-conductance amplifier (OTA-C)-based fractional-order PID (FOPID) controller for BLDC motor speed regulation in a previous study [16]. A hardware-efficient analog structure capable of continuous-time control with adjustable fractional dynamics while still keeping high accuracy and low power consumption was the focus of the work. The paper has shown that the OTA-C implementation provides faster speed tracking, lower overshoot, and better dynamic stability as compared to conventional PID and digital FOPID control schemes. Nevertheless, the method is constrained by the limited performance of analog OTA components that are affected by temperature, are difficult to tune, and whose parameters drift with time.

To enhance steady-state accuracy and transient performance amidst fluctuating load conditions, an Equilibrium Optimizer (EO)-based FOPID controller was developed for the speed regulation of a BLDC motor [17]. Their goal was to use EO's balancing mechanism, which shows the dynamic balance between exploration and exploitation, to find the best values for the five FOPID parameters that would lead to faster convergence and fewer errors. The speed response was found to be excellent with very low settling time when comparing GA- and PSO-based techniques to the method proposed by the authors. However, the limitations of the method include its reliance on simulation environments with no hardware experiments and no robustness tests against parameter uncertainties and sensor noise. In addition, EO's computational cost and the possibility of premature convergence in a high-dimensional search space were not fully considered. In addition, a dual fuzzy logic–based PID controller optimized by the Harmony Search Algorithm (HSA) for BLDC speed control was proposed in a previous study [18]. The primary goal was to improve the system's adaptability and its ability to handle nonlinearity by employing two fuzzy inference subsystems: one for on-the-fly gain adjustment of the PID parameters and another for dynamic error correction, while HSA was used to identify the best scaling factors for reducing steady-state error and overshoot. The simulation results indicated that the proposed controller achieved better dynamic stability, faster response, and higher robustness as compared to classical PID and single-fuzzy schemes.

However, the torque ripple in BLDC motor drives can be reduced through the use of a modified Luo converter with an FOPID controller [19]. Its main goal was to achieve the highest possible converter voltage gain and dynamic performance while also lowering torque pulsations and acoustic noise. The reconfigured topology exhibited higher efficiency and more stable torque than typical DC-DC converters with PID control. Nevertheless, the limitations of the approach arise from its reliance on a fixed controller structure without any optimization or self-tuning features; thus, its adaptability to different operating conditions is limited. The analysis only slightly considered the tolerances of the converter components and the switching losses, which may affect the results' accuracy in the real world. The lack of a complete sensitivity study or robustness examination against load disturbances makes this system less suitable for deployment in the real world.

An enhanced adaptive cascade FOPID controller to handle the speed of a BLDC motor in an EV [20]. The goal was to combine adaptive learning and fractional-order control to realize stable operation under parameter uncertainties, nonlinearity, and external disturbances. The technique changed the control gains on its own from the real-time feedback; thus, it was able to achieve superior tracking accuracy and disturbance rejection as compared to classical FOPID and adaptive PID controllers. However, the approach still has a few drawbacks, such as the reinforcement learning requiring a lot of computational power, the tuning complexity being increased, and there being no experimental validation except for the simulation. Besides, the scalability of the framework to multi-motor or distributed EV systems has not been discussed.

An adaptive Bacterial Foraging Optimisation (BFO)-based PID controller for a dual-motor electric vehicle powered by BLDC machines [21]. The simulation results showed that the adaptive BFO was able to improve the convergence speed and reduce the overshoot as compared to the GA- and PSO-tuned controllers. Nevertheless, the method's heavy dependence on the computationally intensive adaptation and bacterial chemotaxis mechanisms makes it less feasible in real-time for high-frequency control loops. The authors also did not consider nonlinearities such as inverter switching delays and mechanical backlash, which may have an effect on the performance of practical systems.

Besides, a hierarchical FOPID controller whose parameters were tuned by the Pelican Optimization Algorithm (POA) to control the speed of a DC motor in the study [22]. Their main intention was to utilize the hierarchical FOPID structure to achieve a very good dynamic response and robustness, while the POA algorithm was finding the controller parameters that would lead to the minimum integral of the error indices. Simulation results revealed that their method reached faster convergence and a lower steady-state error as compared to the PSO- and GWO-based designs. Nevertheless, the model only considered a single-motor DC scenario and did not consider nonlinear effects typical of BLDC or multi-machine systems.

Also, the classical PID controller optimized by the WOA algorithm to increase the efficiency of the BLDC motor, thereby enhancing the transient response and robustness against load changes [23]. A comparative study of GA, PSO, and GWO-derived systems suggested that the application resulted in a major increase in tracking accuracy and energy efficiency. However, the method has some drawbacks, such as the need for proper WOA parameter tuning and the risk of becoming stuck at local minima. Furthermore, the controller was only evaluated through simulation studies without any noise consideration in hardware or sensors.

The tuning of the PID gains using the teaching-learning-based optimization (TLBO) algorithm was proposed in order to enhance the response of the controller and provide smooth speed BLDC control [24]. The achieved results are compared with the classical PID controllers under different operation conditions. In addition, a new intelligent optimization algorithm called adaptive input-output feedback linearization was applied on the BLDC motor drive to enhance the motor's performance under nonlinear decoupling conditions [25]. The new method effectively improved the transient response and tracking error reduction largely in comparison with the conventional nonlinear controllers. The method is characterized by high algorithmic complexity and significant computational demands, which raises concerns about the real-time applicability of embedded EV systems. Additionally, the controller design relies on precise system modeling, making it sensitive to measurement noise and unmodeled dynamics.

A hybrid PSO-TLBO algorithm to tune the gains of the PID method [26]. The comparison tests showed that TLBO had a faster convergence and better stability margins than PSO, thus the transient response was improved and the torque ripple was reduced. However, the method is constrained by its reliance on fixed optimization parameters, which means that it cannot be very flexible when the solar conditions change quickly.

However, a comparative study for optimizing the PID controllers under different metaheuristic techniques such as PSO, GA, and DE algorithms was conducted in reference [27]. It was found that PSO had a superior convergence rate and robustness while DE resulted in smoother torque and lower oscillations. Nevertheless, the study was confined to static load conditions, and the authors did not consider the adaptability under dynamic torque or speed disturbances. Also, the omission of stochastic uncertainty modeling and the absence of hybrid or adaptive mechanisms limit the possibility of generalizing the results to real-world EV applications. Besides, a new optimization called a driving-training-based optimization algorithm (DTOA) was used to tune the parameters of the BLDC controller. The DTOA-PID control method showed quicker convergence and lower overshoot than exciting methods [28]. However, the optimization framework's computational complexity and parameter sensitivity might restrict real-time adaptability in embedded environments. Besides, the algorithm depended on empirically created learning rules, thus being a problem-specific method and less scalable for different motor ratings or inverter topologies.

The developed PID controller based on a heuristic Adaptive Lyrebird Optimization Algorithm (HALOA) was applied to improve the performance of the BLDC motor drive [29]. Nevertheless, the issue of the algorithm's convergence reliability in high-dimensional search spaces and dynamic disturbances was hardly addressed. However, the integration of hybrid learning or online adaptive methods could enhance the practicality and scalability of the electric vehicle and industrial motor-drive systems in the real world.

The speed overshoot, settling time, and steady-state error of the BLDC are improved by using different optimization methods for controlling the BLDC motor while maintaining a smooth torque response and drive efficiency in industrial applications [30]. The results indicated that metaheuristic-based tuning brought about a substantial dynamic change to the system response in comparison to the traditional Ziegler–Nichols methods. Also, an adaptive Tabu Search Algorithm (TSA) to control the speed of a BLDC motor by tuning the parameters [31]. Accordingly, the dynamic response can be made faster, the overshoot reduced, and the steady-state accuracy enhanced by simply adapting the memory and neighborhood search strategies to the TSA. The adaptive method was able to get better convergence than GA and PSO by changing the control parameters online. The results support the enhancement of speed tracking and torque smoothness during load variation.

Furthermore, a BES-based PI controller for a BLDC motor speed control system using a Bald Eagle Search Algorithm (BES) [32]. The main idea was to use BES's inherent balance between exploration and exploitation to find the best proportional and integral gains, thus minimizing the transient error and overshoot. The results of the simulation indicated that the BES-PI controller has improved rise time, stability, and robustness compared to PSO and slide mode methods. Besides, the control strategy only concentrated on single-objective optimization and did not consider multi-criteria trade-offs like the energy efficiency and torque ripple.

The paper [33] presented a nature-inspired optimization framework to enhance the performance of a solar-battery-supercapacitor hybrid energy storage system for a BLDC motor electric vehicle. The main goal was to manage the flow of energy and control the dynamics of the system using PID controllers that were optimized for each energy source. This led to better power management and a longer driving range.

An adaptive PID-fuzzy controller to control the speed of a BLDC motor while simultaneously reducing harmonics [34]. The foremost purpose was to integrate ordinary PID control with fuzzy-logic adaptability in order to automatically modify control gains and at the same time suppress current harmonics and torque ripple. The experimental outcomes showed that the dynamic response, total harmonic distortion (THD), and stability were substantially better as compared to the fixed-gain PID and PI-fuzzy controllers. The redesign of the control system by the addition of fuzzy-metaheuristic tuning and real-time embedded testing could be a way to further improve the control accuracy and energy saving in EV applications.

The novelty of this work lies in developing a GAO-based fractional-order PI tuning framework for EV motor speed control, where the proportional gain, integral gain, and fractional integral order are simultaneously optimized. In contrast to classical PI/PID controllers and conventional optimizer-based tuning methods, the proposed GAO-FOPI structure provides an additional fractional degree of freedom, enabling a better trade-off between overshoot suppression, rise time, settling time, and steady-state accuracy.

1.4 Contributions

This study proposes a new BLDC speed scheme using the GAO based FOPI controller. By optimizing the FOPI gains, the suggested GAO-FOPI aims to improve the performance of

the BLDC motor under different speed profiles. The GAO-FOPI controller effectively limits the overshoot of the motor's speed, rise time, and steady-state error for optimal operation and enhanced tracking speed time, improving the performance and reducing the trainset issues. The presented approach ensures the GAO rapid convergence by effectively tuning the FOPI gains with various iterations and population sizes.

The key contributions of this article are as follows:

(1) Proposing a new method for controlling motor speed using the GAO algorithm with a Fractional-Order PI (FOPI) controller. Moreover, the proposed method was compared with classical methods such as PI and PSO-PI methods, thus improving both transient and steady-state performances.

(2) The proposed controller is tested in three case studies representing different scenarios: step changes, sudden dropping in speed, and random speed variations. Performance is measured via the main indices such as the rise time, magnitude of overshoot, and steady-state speed error.

(3) The achieved results confirm that the GAO-FOPI controller can limit the maximum overshoot to only 2.13%, reduce the rise time to 1.02 sec, and limit the steady-state error to 0.088 rpm. These results represent improvements of 68%, 40%, and 58% in overshoot, response speed, and tracking accuracy, respectively, compared to the classical PI controller.

(4) The GAO-FOPI controller maintains a high level of performance without the need to increase the complexity of the control structure; hence, it is compatible with practical implementation and advancement of electric drive systems.

The rest of the article is as follows: the methods and materials are discussed in Section 2. Section 3 introduces the GAO algorithm. Section 4 presents the results and discussion. Section 5 summarizes the conclusion of the work.

2. METHODS AND MATERIALS

2.1 Mathematical model of brushless direct current

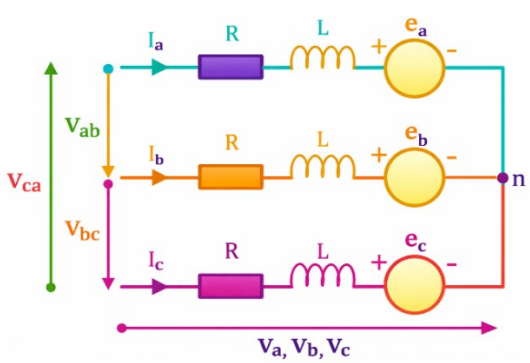

Figure 1. Electrical circuit of the brushless direct current (BLDC) motor

The mathematical model of the BLDC motor incorporates both the electrical and mechanical dynamics that dictate its behavior in an electric vehicle. The phase voltage equations essentially outline the stator electrical behavior by considering resistive losses, inductive effects, and the back electromotive force (back-EMF) [35, 36]. As opposed to sinusoidal machines, the BLDC motor has trapezoidal back-EMF waveforms, which are precisely defined here as the functions of rotor electrical position and mechanical speed. This feature is crucial for the accurate six-step commutation method reflection in practical BLDC drives (Figure 1).

The BLDC voltages were expressed as follows [35, 36]:

$$v_{an} = R_s i_a + \frac{d}{dt}(L_a i_a + M_{ab} i_b + M_{ac} i_c) + e_{an} \quad (1)$$

$$v_{bn} = R_s i_b + \frac{d}{dt}(L_b i_b + M_{ba} i_a + M_{bc} i_c) + e_{bn} \quad (2)$$

$$v_{cn} = R_s i_c + \frac{d}{dt}(L_c i_c + M_{ca} i_a + M_{cb} i_b) + e_{cn} \quad (3)$$

where, R_s is the phase resistance, i_a, i_b, and i_c are the currents of the phases a, b and c, respectively, $v_{an}, v_{bn},$ and v_{cn} are the phase voltages of the motor, $e_{an}, e_{bn},$ and e_{cn} are the induce phase voltages of the motor, L_a, L_b, and L_c represent the self-inductances of the phases, and $M_{ab}, M_{ac}, M_{ba}, M_{bc}, M_{ca}, and\ M_{cb}$ are the mutual inductances between the motor's phases. where $M_{ab} = M_{ba} = M_{ca} = M$ and $L_a = L_b = L_c = L$.

Therefore, Eqs. (4)-(6) are rewritten as:

$$v_{an} = R_s i_a + \frac{di_a}{dt}(L - M) + e_{an} \quad (4)$$

$$v_{bn} = R_s i_b + \frac{di_b}{dt}(L - M) + e_{bn} \quad (5)$$

$$v_{cn} = R_s i_c + \frac{di_c}{dt}(L - M) + e_{cn} \quad (6)$$

However, the EMF voltages (e_a, e_b and e_c) with trapezoidal waveforms are written as follows:

$$e_a = K_e\ \omega_m\ f_a(\theta_e) \quad (7)$$

$$e_b = K_e\ \omega_m\ f_b(\theta_e) \quad (8)$$

$$e_c = K_e\ \omega_m\ f_c(\theta_e) \quad (9)$$

where, K_e is the voltage constant, ω_m represents the motor speed in rad/s, $f_a(\theta_e)$, $f_b(\theta_e)$, and $f_c(\theta_e)$ are the three-phase trapezoidal wavers of unity magnitudes.

However, the electromagnetic torque formula creates a connection between the electrical subsystem and mechanical dynamics by means of phase currents and back-EMF, thus providing the capability of determining torque generation accurately under different load and speed conditions. This motor torque (T_{em}) was written as follows [36]:

$$T_{em} = \frac{(e_a\ i_a + e_b\ i_b + e_c\ i_c)}{\omega_m} \quad (10)$$

Or

$$T_{em} = T_L + J_m \frac{d\ \omega_m}{dt} + B\ \omega_m \quad (11)$$

where, B is the moment of inertia, J_m is the inertia of the motor, and T_L is the load torque.

2.2 Modelling of the electric vehicle

Figure 2 depicts the longitudinal force balance on an electric vehicle that is driving uphill on an inclined road with a slope angle α [37, 38]. The electric drivetrain generates the traction force Ft, which is transmitted through the wheels to counteract the vehicle's resistive forces. The rolling resistance force (F_r) is formed due to the interaction of the tire with the road surface and is directed opposite to the vehicle's forward motion along the slope. On top of that, the aerodynamic drag force (F_{aero}) also opposes the vehicle motion and increases with the vehicle speed, which is the effect of air resistance that becomes quite significant at higher car velocities. The portion of the gravitational force Gravity that acts downwards results in a resistive part that is proportional to the incline of the road, and consequently, it significantly affects the vehicle performance during the uphill running. The total force equation is written as follows [37]:

$$F_{Total} = F_t - F_{rolling} - F_{aero} - F_{gradient} \tag{12}$$

where, F_t represents the total motor force, $F_{rolling}$ is the rolling resistance force, F_{aero} represents the aerodynamic force, and $F_{gradient}$ represents the grade ability force. These forces are shown in Figure 2.

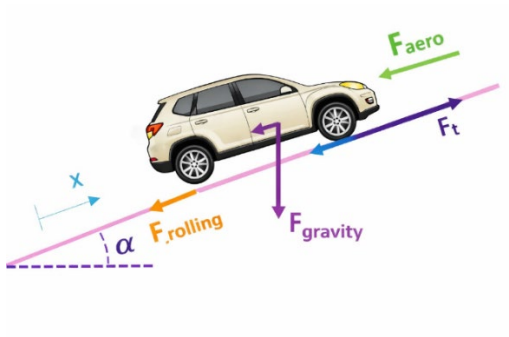

Figure 2. Electric vehicle (EV) traction force distribution

The equation of each force can be expressed as follows [38]:

$$F_{gradient} = M_v\, g\, sin(\alpha) \tag{13}$$

$$F_{\text{rolling}} = C_r M_v\, g\, \cos(\alpha) \tag{14}$$

$$F_{ad} = 0.5\, \rho\, v^2 A\, C_d \tag{15}$$

where, M_v is the vehicle mass, ρ is the air density C_r is the rolling friction coefficient, g represents the gravity acceleration, C_d is the drag coefficient α is the road slop, v is the vehicle speed, and A is the frontal area. Furthermore, the load power produced by the traction engine ($P_{EV}(t)$) is written as follows:

$$P_{EV}(t) = F_t(t) \times v(t) \times \eta_t \tag{16}$$

where, η_t is the mechanical transmission efficiency.

2.3 Control design of the brushless direct current driven electric vehicle

Figure 3 shows the complete control structure of the EV with a BLDC motor drive, which employs optimized PI and FOPI control strategies. The driver control signal produces the reference speed (Nref) that is compared to the motor speed feedback (Nm) thus, the speed error is computed. The error signal is fed into the PI and FOPI speed controllers, the parameters of which are automatically adjusted via the GAO to provide faster responses and more robust operations under different driving scenarios.

The output of the optimized controller is a PWM control signal that regulates the DC/DC boost converter, enabling efficient voltage adaptation from the battery to the DC link. Then the DC/AC inverter converts the regulated DC voltage into three-phase AC excitation for the BLDC motor. The commutation process of the motor is carried out with the aid of rotor position information obtained from Hall sensors, which is used to correctly energize the motor phases for torque generation [37]. The addition of optimization-based PI tuning results in significantly better speed tracking, less overshoot, lower steady-state error, and greater overall system performance; thus, the proposed control strategy becomes a perfect fit for high-performance EV propulsion systems.

Figure 4 shows the proposed controller of the BLDC motor. The mathematical equation of the PI and FOPI controllers are expressed in Eq. (17). Eq. (18) reports the proposed objective function. As observed, the optimization cost function minimizes the error signal $e(t)$ using the GAO algorithm to ensure the optimal speed control technology using the standard time-integral absolute error ($ITAE$).

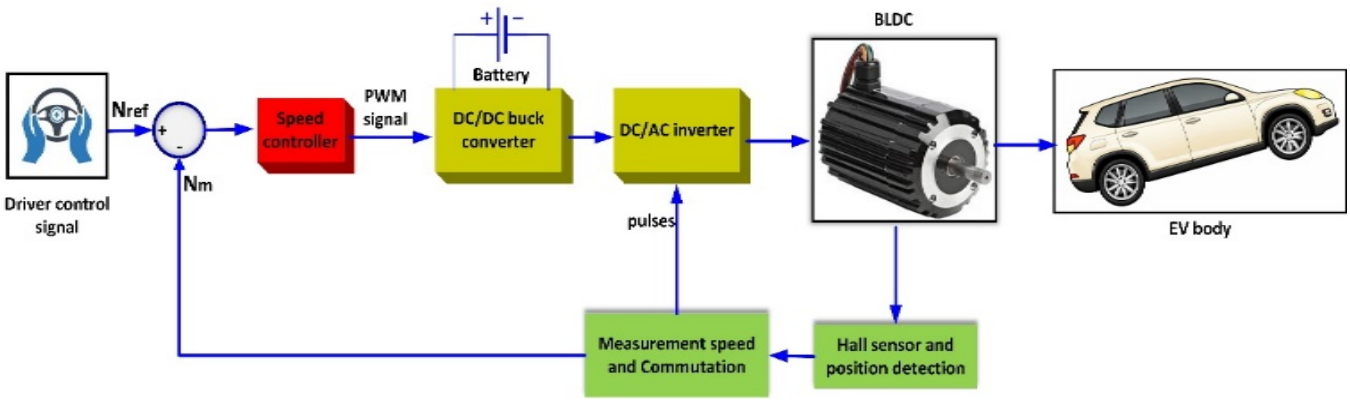

Figure 3. Control structure of the electric vehicle (EV) with brushless direct current (BLDC) motor drive

$$\begin{cases} U_{PI} = e(t) \times k_p + \int k_i \times e(t) \\ U_{FOPI} = e(t) \times k_p + k_i \times D_t^{-\lambda} \times e(t) \end{cases} \quad (17)$$

where, k_p and k_i are the proportional gain and integral gain of the PI and FOPI controllers, λ represents the fractional order of the integrating action and $e(t)$ was written as $e(t) = N_{ref}(t) - N_m(t)$. where $N_{ref}(t)$ is the reference speed and $N_m(t)$ is the measured speed in RPM.

$$ITAE = \int_{t=0}^{t} t \times |e(t)|\, dt \quad (18)$$

where, t represents the time response at steady-state condition. The limits of the optimized gains are written in Eq. (19):

$$0.001 < k_p < 2, 0.004 < k_i < 2 \text{ , and } 0 < \lambda < 1 \quad (19)$$

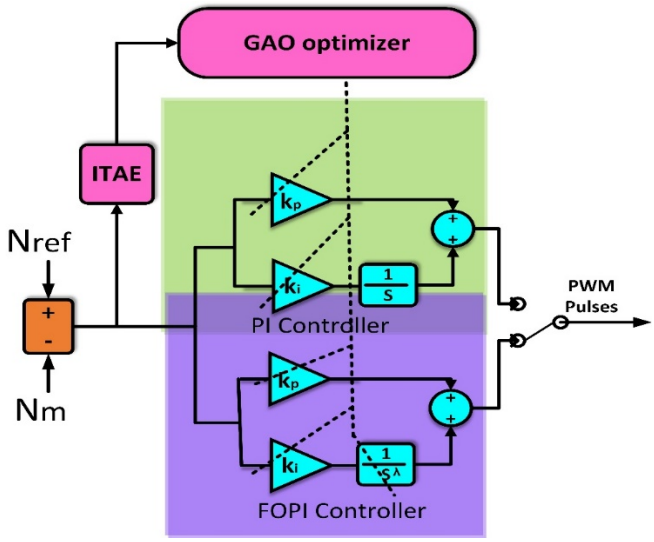

Figure 4. Proposed optimized PI and FOPI controller
Note: PI = proportional integral; FOPI = fractional order-proportional integral

3. GIANT ARMADILLO OPTIMIZATION ALGORITHM

3.1 Giant Armadillo Optimization algorithm overview

The GAO algorithm is one of the newer bio-inspired metaheuristics that rely on nature to design their strategies, in this case, the armadillo's search for food, digging, and staying safe from danger. GAO was proposed in reference [15] to tackle the most difficult nonlinear and multimodal optimization problems, which classical and swarm-based algorithms can hardly solve because they usually remain prematurely stuck in a local optimum. The "armadillo" in the algorithm represents a solution vector, so the whole population of potential solutions is interpreted as a group of armadillos searching for the best one. In GAO, an armadillo changes its position either by its foraging movement (looking for food here and there) or its burrowing activity around a promising nest (good solution) found previously, and so, there is intensive exploitation around a region. The algorithm is equipped with position-updating strategies that take into account and appropriately distribute the needed amount of global exploration and local exploitation during the search [15]. Figure 5 shows the Giant armadillo.

At the beginning of the optimization, the cells of the search space are explored via large random movements of the armadillos, enabling them to cover vast areas, thus increasing population diversity and impeding premature convergence. Subsequently, the armadillo's activities become more oriented towards the burrow, where the solution refinement is achieved in a much more localized manner. The idea of defensive escape of the armadillo is metaphorically incorporated as an algorithmic strategy to intentionally cause disruption in the direction and/or the magnitude of the position update, providing an opportunity for the algorithm to come out of its local optimum.

The way a giant armadillo hunts consists of two main activities: first, it locates and approaches termite mounds, and second, it digs in the termite mounds to eat the termites. Mathematical description of the giant armadillo's natural hunting behaviors is used in the development of the novel GAO method, which is elaborated below.

Figure 5. Giant armadillo taken and edited based on reference [15]

3.2 Mathematical analysis

3.2.1 Exploration phase

Giant armadillos form the population of the algorithm that can be mathematically represented by a matrix, as in Eq. (20). The initial locations of giant armadillos in the solution space are randomly generated at the start of the algorithm run as per Eq. (21) [15].

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1d} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{id} & \cdots & x_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_N & \cdots & x_{Nd} & \cdots & x_{Nm} \end{bmatrix} \quad (20)$$

$$x_{id} = lb_d + r \times (ub_d - lb_d) \quad (21)$$

where, X is the optimizer matrix, X_i represents the i[th] candidate solution, x_{id} is the decision variable in dimension search space d[th], N denotes the number of agents, m is the variables of decision, ub_d and lb_d are the upper and lower limits of the search space, respectively, and r is a random number [0,1].

The position of each giant armadillo in the problem-solving space reflects that it is a candidate solution for the problem,

and thus, a value for the objective function can be evaluated corresponding to each giant armadillo. Thus, the objective function was written as follows:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix} \quad (22)$$

where, F is the objective function vector, F_i represents the evaluated objective function.

The first phase of the GAO algorithm updates the positions of the population members in the problem-solving space by imitating the hunting behavior of giant armadillos when they attack termite mounds. The inspiration comes from a giant armadillo's movement to the termite mounds, which is mathematically modeled to lead the search process. Just considering the modeling of the movement of giant armadillos to termite mounds, a different location is determined for every individual of the population according to Eq. (23). Next, this new position becomes the current position of the joint if it increases the objective function value according to Eq. (24).

$$x_{i,j}^{p1} = x_{i,j} + r_{i,j} \times (stm_{i,j} - I_{i,j} \times x_{i,j}) \quad (23)$$

$$X_i = \begin{cases} X_i^{p1}, & F_i^{p1} \le F_i \\ X_i, & else, \end{cases} \quad (24)$$

where, $stm_{i,j}$ represents the i^{th} giant armadillo termite mound, $x_{i,j}^{p1}$ is the j^{th} dimension of the new agent, X_i^{p1} denotes the new search agent, F_i^{p1} denotes the objective value, $r_{i,j}$ is the random number and $I_{i,j}$ is the random coefficient which equals 1 or 2 [15].

3.2.2 Exploitation phase

In the GAO framework, the digging ability of the giant armadillo in the termite mounds is mathematically described to change the location of each individual in the population, as shown in Eq. (25). This act of digging is like an intensive local search procedure that allows the armadillo to carefully exploit the areas near the good solutions in the solution space. Once the new location is calculated, it is evaluated by the objective function. In case the new location results in a better solution, it is used to replace the previous location of the individual in the population, as in Eq. (26) [15].

$$x_{i,j}^{p2} = x_{i,j} + \left(1 - 2r_{i,j}\right) \times \frac{ub_j - lb_j}{t} \quad (25)$$

$$X_i = \begin{cases} X_i^{p2}, & F_i^{p2} \le F_i \\ X_i, & else, \end{cases} \quad (26)$$

where, $x_{i,j}^{p2}$ is the j^{th} dimension of the new agent for d the igging phase, X_i^{p2} denotes the new search agent for digging phase, F_i^{p2} denotes the objective value, $r_{i,j}$ is the random number and t is the iteration counter.

The flowchart of the AGO is shown in Figure 6. This figure displays the step-by-step execution of the GAO algorithm for the optimal PI controller tuning based on the ITAE performance index. The two-phase search mechanism of GAO, consisting of exploration (Phase 1) and exploitation (Phase 2), allows attaining a proper trade-off between global search diversification and local search intensification. The candidate solutions are progressively refined according to Eqs. (23)-(26). Thus, the algorithm not only prevents stagnation at local optima but also continuously enhances the tracking performance. However, choosing ITAE as the fitness function is a clear signal of the focus on the long-term error minimization, which eventually leads to faster settling time and better steady-state accuracy in the BLDC motor speed control.

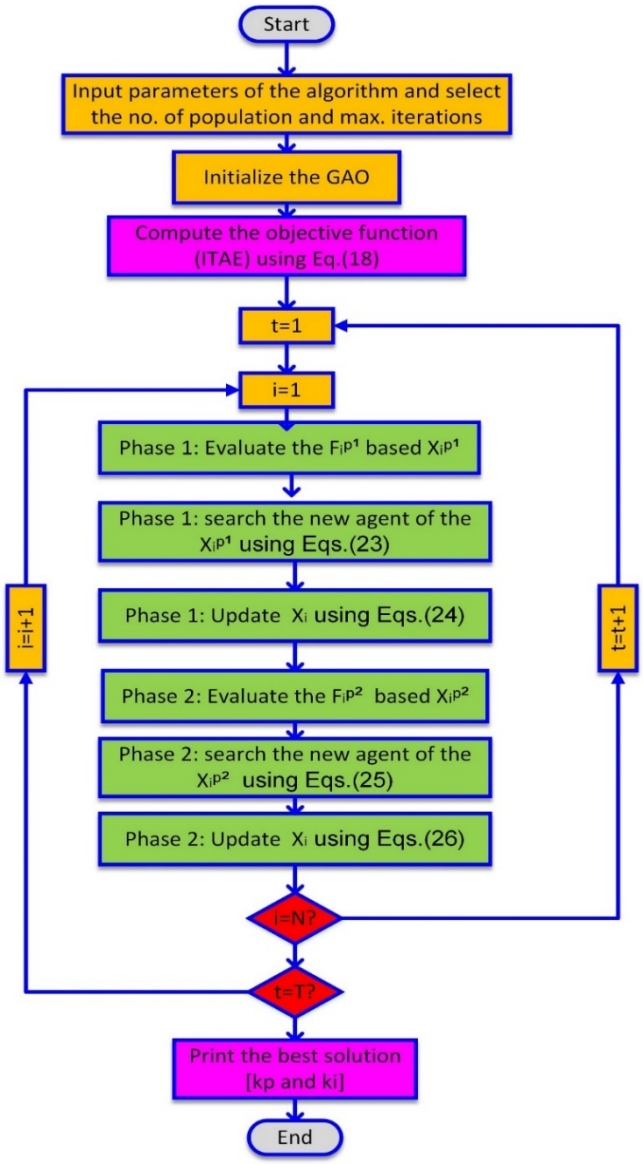

Figure 6. Flowchart of the GAO algorithm for the optimal PI controller tuning

Note: GAO = Giant Armadillo Optimization; PI =proportional integral

3.3 Comparison the Giant Armadillo Optimization with other classical algorithms

The GAO algorithm, which stands for Group Armadillo Optimization, is one of the most recent metaheuristic optimizers. The foraging and defensive behavior of armadillos inspired the GAO algorithm. If we compare GAO with traditional algorithms such as GWO, GA, WOA and PSO, it can be seen that the former is capable of providing a better, more adaptive balance between exploration and exploitation. As a result, this approach not only improves the convergence

speed but also decreases the possibility of premature convergence. There is a previous study highlighting the capability of GAO to efficiently solve nonlinear optimization problems, including parameter tuning and energy system optimization. Nonetheless, its use in fractional-order controller tuning for EV motor speed control is still a virgin area, which marks one of the main contributions of this paper.

Table 1 presents a basic comparative analysis of the most commonly used algorithms in relation to the GAO. As presented in Table 1, the classical algorithms have either strong exploitation or strong exploration, but usually, they struggle to strike a good balance between both activities. Techniques like GWO and WOA enhance this balance but sometimes experience slow convergence issues at the fine-tuning phases. On the other hand, GAO has a quite capable exploration and exploitation adaptive mechanism, which leads to faster convergence and a lower chance of premature stagnation. For this reason, GAO is very suitable for the tuning of fractional-order controllers in nonlinear EV motor control systems, wherein both global searching ability and local adjustments matter.

For comparison, the PSO is widely used, but the risk of stagnation remains; WOA is a good balance, but fine-tuning takes time; GWO is effective but susceptible to parameter changes; and GA is a robust global search, but it demands a lot of computation. Unlike the used GAO, it's a great choice for nonlinear control optimization.

Table 1. Comparative analysis of GAO with other algorithms

Algorithm	PSO	WOA	GWO	GA	GAO
Inspiration	Swarm intelligent (birds)	Whale bubble-net	Grey wolf hunting	Natural selection	Armadillo foraging & defense
Exploration	Moderate	High	High	High	High
Exploitation	High	Moderate	Moderate	Moderate	High
Computational time	Fast	Moderate	Balanced	Slow	Fast
Convergence speed	Low	Low	Low	Medium	Very fast

Note: PSO = particle swarm optimization; WOA = whale optimization algorithm; GWO = grey wolf optimizer; GA = Genetic algorithm; GAO = Giant Armadillo Optimization

4. RESULTS AND DISCUSSION

4.1 System details and parameters

Figure 7 displays the overall layout of the proposed EV driven by a brushless DC (BLDC) motor drive using MATLAB/Simulink software. The studied system has essentially three main components. Block (1) is the layer of speed control where the reference speed of the vehicle taken from the drive cycle is first changed into the motor speed, and then the motor speed is compared with the measured speed to produce the control error signal. The optimized PI controller gives the control signal (duty cycle and braking commands) for the accurate speed tracking in block (2). Block (3) comprises the power conversion and electromechanical subsystems, including a DC source, inverter, and BLDC motor, where the duty cycle directly controls the motor's electromagnetic torque. Block (4) simulates the electric vehicle mechanical dynamics, where the motor torque is converted to vehicle motion and the load and inertial effects are also considered. The closed-loop operation of these blocks allows uninterruptible speed control at different load conditions; thus, the proposed control strategy is apt for the implementation of realistic EVs. The system parameters of the BLDC motor and EV are reported in Table 2. The GAO algorithm was tested under different numbers of the iterations and population size. Table 3 reports the obtained PI and FOPI results. The gains of the PI and FOPI controllers are achieved under a number of runs (20 runs) based on a simulation time of 150 sec. Also, different numbers of iterations (T) and population sizes (N) are used to test and validate the proposed controllers.

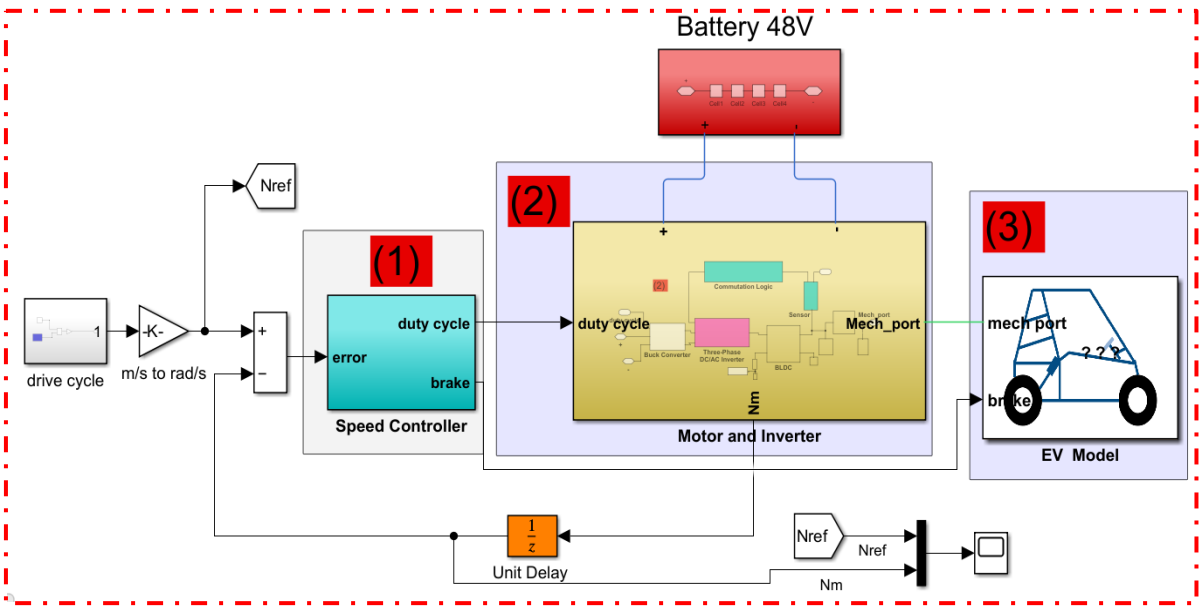

Figure 7. Proposed system diagram under MATLAB/Simulink

Table 2. System parameters

Item	Parameter	Value
brushless direct current (BLDC) motor	Supply rated voltage	48 V
	Rated speed	1000 RPM
	Stator d-axis inductance (L_d)	10 mH
	Stator q-axis inductance (L_q)	10 mH
	Stator resistance per phase (R_s)	0.001 Ω
	Flux leakage	0.05 wb
	Number of pole pairs (p)	4
	Rotor inertia	0.1 $kg.m^2$
	Rotor damping	0.01 $N.m^2/(rad/s)$
Vehicle	Vehicle mass (M_v)	250 kg
	Weel radius (r)	0.2794 m
	Front axle (d_f)	1 m
	Rear axle (d_r)	0.35 m
	Rolling resistance (C_r)	0.015
	gravity acceleration (g)	9.81 m/s^2
	Air density (ρ)	1.18 kg/m^3
	the drag coefficient (C_d)	1.92
	frontal area (A)	0.735 m^2

Table 3. Obtained PI and FOPI results using the GAO algorithm

Iteration and Population Size	PI Gains		FOPI Gains		
	k_p	k_i	k_p	k_i	λ
$T = 10, N = 10$	0.4	0.02	0.3	0.08	0.8
$T = 20, N = 15$	0.5	0.01	0.24	0.088	0.83
$T = 50, N = 10$	0.44	0.06	0.34	0.05	0.76
$T = 100, N = 20$	0.64	0.088	0.42	0.057	0.97
$T = 100, N = 25$	0.74	0.032	0.5	0.074	0.91

Note: PI = proportional integral; FOPI = fractional order-proportional integral; GAO = Giant Armadillo Optimization

4.2 Case studies

This section presents a detailed examination of testing the BLDC-EV system under three different case studies, which are no-load operation, sudden speed increase, and random speed variation. These scenarios are evaluated to show the stability, transient robustness, and adaptability of the proposed PI and FOPI controllers framework. The performance of the GAO algorithm was compared with the classical PI and PSO-PI controllers.

4.2.1 No load operation

Figure 8 shows the speed of the motor by considering the reference value of 1030 RPM using the studied conventional PI, PSO-tuned PI, GAO-tuned PI, and GAO-based FOPI methods.

As shown in this figure, at the beginning, each controller was able to speed up the motor to the reference speed. However, the conventional PI controller shows a high overshoot in the speed curve of more than 1100 RPM and a slow settling time. The fixed-gain PI control's limited adaptability, which may result in mechanical damage and the inefficiency of the motor. The rise time of the PI method is 1.63 sec, and the overshoot in the motor's speed is 7.8 %.

Furthermore, the PSO-PI controller improves the stability by limiting the overshoot (3.8 %) and speeding up convergence (1.59 sec). The speed curve reaches the final state more quickly (0.16 RPM) than with the traditional PI due to the parameter tuning based on PSO optimization. Also, the steady-state error of the method is still high. Besides faster settling time and less overshoot, the GAO-PI controller can improve the dynamic performance of the BLDC drive. Thus, the better balance between exploration and exploitation of GAO is confirmed by the outstanding results of the controller adjustments. Against, the GAO-FOPI controller is working with the most exceptional results. The controller shows very little overshoot, very fast convergence, and a very small steady-state error. As observed in this figure, it is clear that GAO-FOPI remains very close to the reference speed during both transient and steady-state stages. The fractional-order design provides a significant enhancement with a more precise and accurate tracking speed method. The overshoot values of the GAO-PI and GAO-FOPI are 3.5 % , and 2.5 % , respectively. The rise times of GAO-PI and GAO-FOPI are 1.61 sec and 1.6, respectively. The steady state error for the PI is 0.15 sec , for the PSO-PI is 0.16 sec , for GAO-PI is 0.152 sec and for the GAO-FOPI is 0.089 sec. Based on the achieved indices, the suggested GAO-FOPI presents less overshoot in the speed of the motor, faster tracking time, and less steady-state error when compared with other methods.

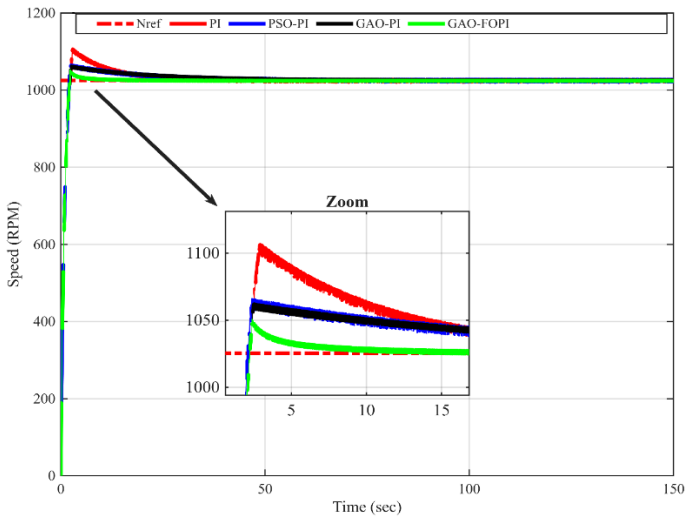

Figure 8. Speed response of the brushless direct current (BLDC) at the no-load condition

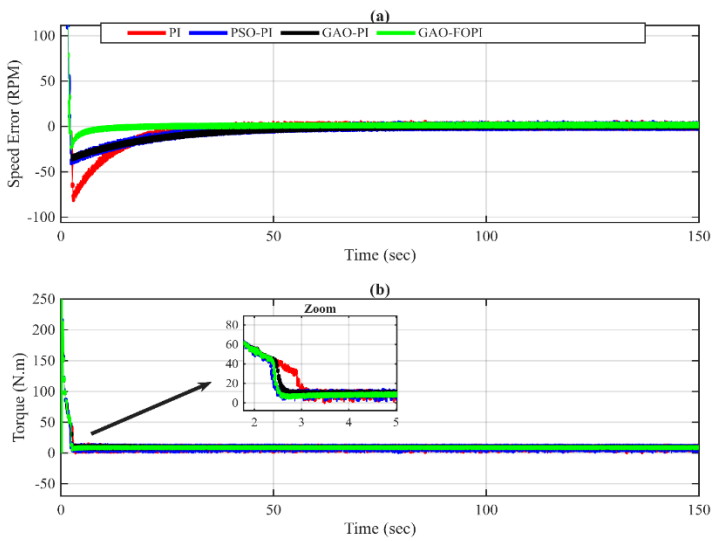

Figure 9. (a) Speed error, (b) electromagnetic torque of brushless direct current (BLDC)

Figure 9 displays the tracking error of the speed and

electromagnetic torque of the BLDC motor in this case study. As shown in this figure, the speed error using the PI method exhibits a large initial deviation of approximately -84 RPM, along with a slow convergence time. This results in poor damping response and high transients. In contrast, the second controller, PSO-PI, significantly enhances this response by reducing the error to -44 RPM, representing a 46% improvement compared to the classical PI controller. The GAO-PI controller improves the response of the speed with a limited peak error about (-30 RPM). The improvement of this controller is 65% compared with the PI method. The best improvement was shown using the GAO-FOPI controller, which presents the best response with a limited maximum error of (-14 RPM), ensuring improvement of more than 81% relative to the PI technique. Thus, the presented GAO-FOPI has zero steady-state and very high tracking time.

The electromagnetic torque response under the four controllers has been shown in Figure 9(b). As observed in this figure, the PI controller causes the motor to generate a very high transient torque peak of over 220 Newton-meter (N·m), which could theoretically lead to wear and tear of the drivetrain due to mechanical stress. The PSO-PI controller limits this peak torque to only about 150 N·m, which is a reduction of nearly 32%. The GAO-PI controller also significantly reduces this peak, to almost 110 N·m, whereas the proposed GAO-FOPI technique results in the most stable operation and the maximum torque below 90 N·m, which corresponds to a reduction of almost 60% in comparison with the classical PI controller.

The GAO-FOPI ensures the fastest torque settling time at approximately 2.5 seconds, whereas the PSO-PI and PI controllers take about 4 seconds and 5 seconds, respectively. This superior torque control results in a reduced load on the current, which leads to decreased vibration of the mechanical components, thereby enhancing the system's overall reliability.

The achieved results show that the suggested GAO-FOPI approach is more effective than the conventional PI and PSO-PI methods. In comparison with the speed response of the conventional methods, the result shows that the new GAO-FOPI controller has less overshoot, fast convergence, and less steady-state error. Moreover, the proposed method always ensures the dynamics are smooth and fast. Furthermore, when compared to PSO-PI and GAO-PI methods, the recommended GAO-FOPI demonstrates superior performance in terms of robustness and consistency, particularly during specific stress transient phases of the BLDC.

4.2.2 Sudden decrease of speed

This scenario describes the case of a step reduction in the BLDC speed from $1030\,RPM$ to $618\,RPM$ at time $t = 75\,sec$, as indicated in Figure 10. As observed in this figure, the controllers can ensure the kinetic energy stored in the rotating system while at the same time avoiding overshoot and a long recovery period. Thus, the conventional PI controller has the worst dynamics with the speed falling to $582\,RPM$ with high an undershoot about (7.9%) very long rise time ($1.65\,sec$). PSO-PI and GAO-PI reduce the transient response significantly, but they still suffer from plenty of undershoot of 3.9 % and 3.4 %, respectively. Especially in the case of PSO-PI that takes more than $23\,sec$ to settle fully. The rise times of the PSO-PI and GAO-PI are $1.6\,sec$, and $1.5\,sec$, respectively.

On the other hand and under the sudden change of speed, the proposed GAO-FOPI controller yields a much better response with a rise time of $1.42\,sec$. The lowest speed is therefore around $618\,RPM$ which corresponds to the smallest undershoot of (2.3%) while the system is quickly brought back within ±2% range less than 1.7 sec after the disturbance. Thus, the undershoot has been reduced by more than half and the settling time by almost 90% as compared to the baseline PI controller. The better results can be explained by the fractional-order integration operation. Hence, the GAO-FOPI controller can smoothly decelerate, quickly recover and be more robust than other controllers.

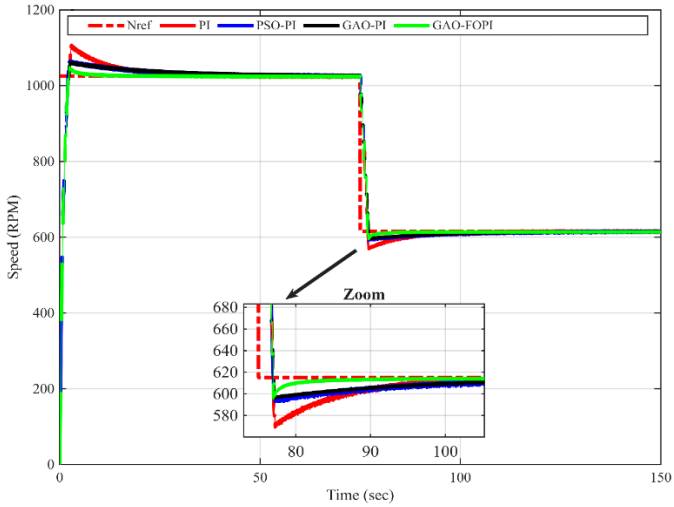

Figure 10. Speed response of the brushless direct current (BLDC) in the second case study

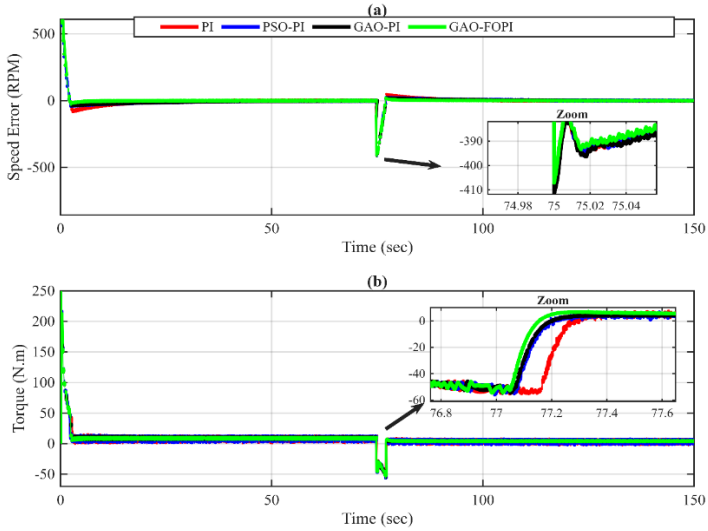

Figure 11. Results of the second case study (a) Speed error, (b) Electromagnetic torque of brushless direct current (BLDC)

Figure 11 shows the obtained response of the speed error and the motor's torque. At starting, all controllers show a big transient error due to initial acceleration from standstill, but the GAO-FOPI controller quickly eliminates this error and then smoothly goes to zero, which means it has good damping and fast transient regulation. At the sudden drop time, there is an immediate and big negative error spike for all methods, which is due to the sudden deceleration demand. The PI control results in the greatest negative deviation and the slowest recovery, whereas the PSO-PI and GAO-PI reduce the peak error but still have a significant oscillation. On the other hand, GAO-FOPI facilitates the smallest negative speed error

and the fastest return to zero error. The electromagnetic torque curve under the applied controllers was shown in Figure 11(b).

The torque is very closely related to the speed error, which is why it explains how the speed error works. All controllers produce a big positive torque at the start of the motor to overcome inertia. However, GAO-FOPI seems to be the best at limiting the torque peak and quickly settling to a value that is almost zero, which reduces the mechanical stress. A negative braking torque is created right away when the speed drops quickly. When the torque direction changes, the PI controller acts slowly and with a delay, which makes the speed error last for a long time. PSO-PI and GAO-PI make it possible to get a good response, but they are still slow. But GAO-FOPI quickly makes a large negative torque that is well-controlled, with very little overshoot, and then smoothly returns to normal operation. Finally, this investigation shows that the newly developed GAO-FOPI is able to provide excellent transient behavior, minimal torque ripple, and increased robustness; it is a good candidate for the electric vehicle motor drive system, which is exposed to sudden speed changes.

4.2.3 Random speed profile

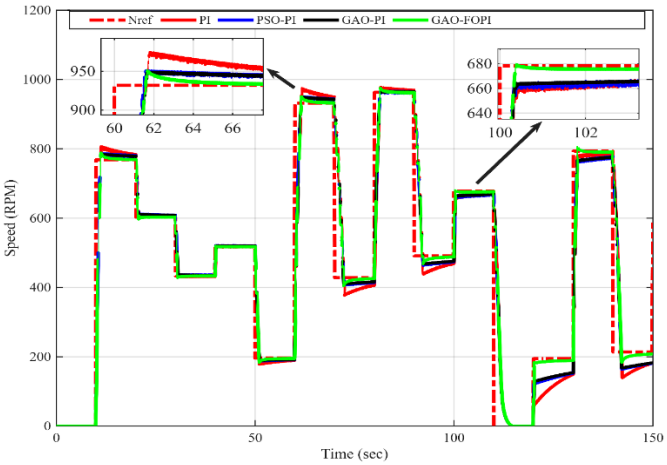

Figure 12. Brushless direct current (BLDC) motor speed response under random variations case

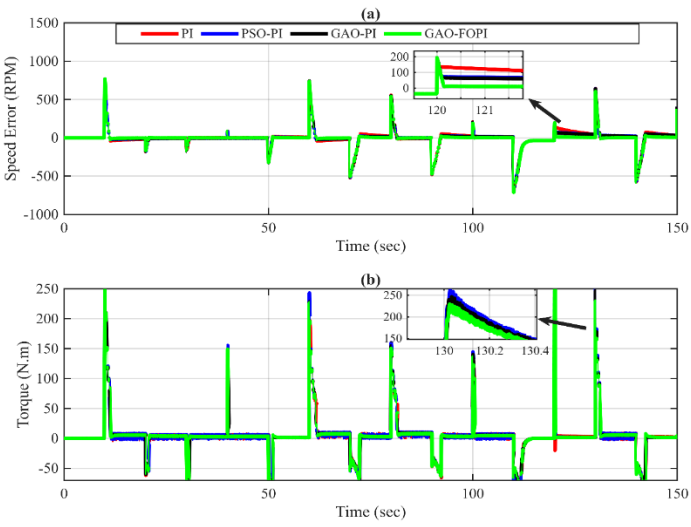

Figure 13. Results of the third case study (a) Speed error, (b) Electromagnetic torque of brushless direct current (BLDC)

In this section, the random speed profile is used to show the performance of the BLDC motor under the PI, PSO-PI, GAO-PI, and GAO-FOPI controllers. This profile represents a highly fluctuating reference speed simulating real driving situations with frequent speeding up and slowing down, as shown in Figure 12. As indicated, the traditional PI controller shows slow and poor response in terms of the convergence speed, maximum overshoot and rise time, which suggests that the controller has limited damping ability and is not able to handle rapid changes in the operating conditions.

On the other hand, although PSO-PI and GAO-PI provide a well response with small unwanted errors and longer settling intervals, especially at medium and high speeds. The rise times of the PI, PSO-PI, and GAO-PI are $1.72\ sec$, $1.64\ sec$, and $1.65\ sec$, respectively. The suggested GAO-FOPI provides the best performance with a minimal rise time of $1.02\ sec$. The overshot value of the PI and PSO-PI are $8.4\ \%$ and 3.03%, respectively. The proposed controllers (GAO-PI and GAO-FOPI) present a less overshoot value of 2.98% and 2.13%, respectively. As observed, the proposed algorithm GAO can improve the dynamic response of the PI and FOPI controllers by adjusting the optimal gains. The GAO-FOPI avoids the steady state error in the speed, where it provides less than $0.088\ RPM$ under this case study when comprwed to the PI and PSO methods of $0.21\ RPM$ and $0.18\ RPM$.

Meanwhile, the new GAO-FOPI regulator regulates the deviation at the minimum steady-state level, and consequently, it provides the highest quality tracking in the BLDC speed range with minimum deviation from the reference profile. Thus, GAO-FOPI is more efficient in reducing the time for the error to become stable close to zero, and this error is smaller in the steady state, while at the same time, it is less overshooting and oscillating as presented in the response of the error and torque figure (Figure 13). Moreover, these results prove that the GAO-FOPI controller is capable of ensuring a stable operation with robust performance of the electromagnetic torque under rapidly changing speed values.

4.3 Comparative analysis and assessment

To prove the effectiveness of the applied controllers, a comprehensive quantitative comparison of the performance indices obtained by the PI, PSO-PI, GAO-PI, and GAO-FOPI methods was conducted under three scenarios as listed in Table 4.

Table 4. Performance indices of the speed under the PI, SO-PI, GAO-PI, and GAO-FOPI controllers

Case Study	Control Method	Rise Time (sec)	Overshoot (%)	Steady State Error (RPM)
Case 1	PI	1.63	7.8	0.15
	PSO-PI	1.59	3.8	0.16
	GAO-PI	1.61	3.5	0.152
	GAO-FOPI	1.6	2.5	0.089
Case 2	PI	1.62	7.9	0.155
	PSO-PI	1.65	3.86	0.146
	GAO-PI	1.54	3.45	0.142
	GAO-FOPI	1.42	2.3	0.09
Case 3	PI	1.72	8.4	0.21
	PSO-PI	1.64	3.03	0.18
	GAO-PI	1.65	2.98	0.16
	GAO-FOPI	1.02	2.13	0.088

Note: PI = proportional-integral; PSO-PI = particle swarm optimization-proportional-integral; GAO-PI = Giant Armadillo Optimization proportional-integral; GAO-FOPI = Giant Armadillo Optimization – fractional-order proportional integral

This table shows the evaluation of the studied controllers in

terms of rise time, overshoot in the speed, and steady state error of the speed for each case study. The rise time, overshoot, and the steady-state error for each case study are displayed in Figures 14-16. The classical PI controller exhibits the worst overshoot of 7.8 %, a huge steady-state error of 0.15 sec. Although the enhanced PSO-PI and GAO-PI methods decrease the overshoot of the speed to 3.8 %, and 3.5 %, respectively, their rise times remain large (1.59 sec, and 1.61 sec). Unlike, the presented GAO-FOPI obtains the best rise time of 1.6 sec, the lowest overshoot of 2.5 %, and the best steady state error less than 0.089 RPM. Thus, the proposed GAO-FOPI reduces the overshoot to 67% relative to the classical PI method.

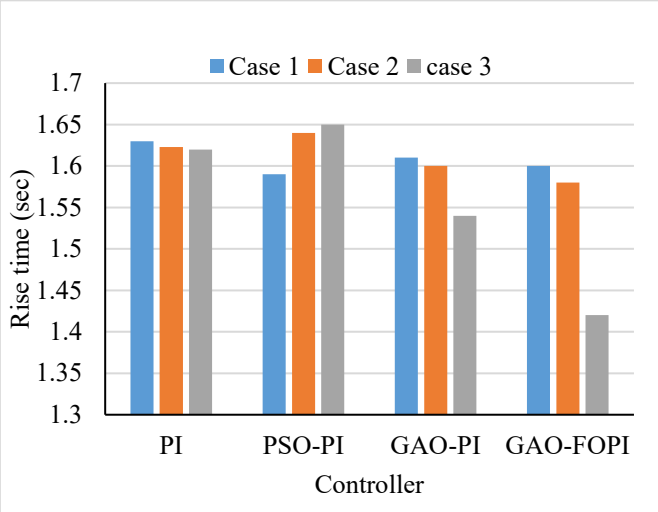

Figure 14. Rise time comparison of the applied controllers

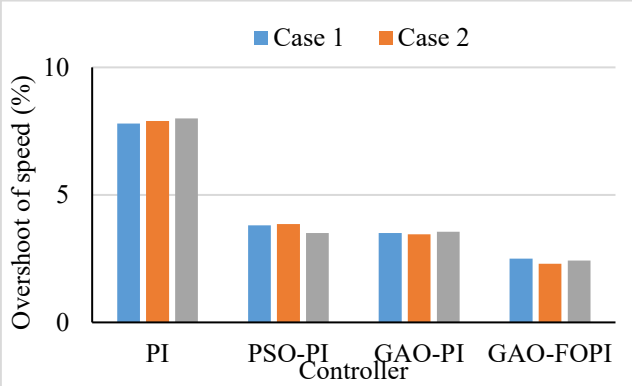

Figure 15. Overshoot of the speed of the applied controllers

In Case 2, the rise times of the PI, PSO-PI, GAO-PI, and GAO-FOPI are 1.62 sec, 1.65 sec, 1.54 sec, and 1.42 sec , respectively. The overshoot of the GAO-FOPI is the best value of 0.09 RPM. The other controllers provide a 0.142 RPM in the GAO-PI, 0.146 RPM using PSO-PI, and 0.155 RPM under the PI. The achieved results indicate superior transient-steady state error using the suggested GAO-FOPI as shown in these figures. Furthermore, the superiority of the proposed method is further illustrated in Case 3, which is a reflection of the most intense dynamic scenario. In this case, the PI controller shows a peak overshoot of 8.4 % and the highest steady-state error (0.21 RPM), while the PSO-PI and GAO-PI achieves acceptable overshoot values of 3.03 % and 2.98 %, respectively. On the other hand, GAO-FOPI drastically reduced the rise time to just 1.02 sec, thus having a 40% quicker reaction than PI, and at the same time, the overshoot is limited to 2.13 % and the steady-state error is reduced to 0.088 RPM. These figures thus prove that the proposed GAO-FOPI controller can maintain faster dynamics, better damping, and higher tracking accuracy for all of the cases tested. The results demonstrate that the GAO-based fractional-order PI control is a powerful and scalable solution that can outperform the conventional PSO algorithm or classical PI method. To clear the novelty of the proposed algorithm.

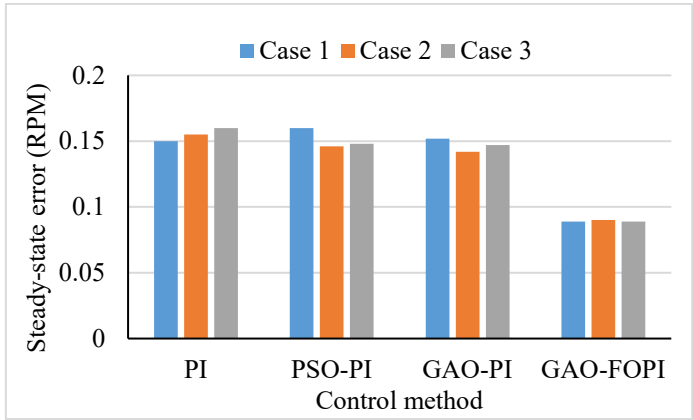

Figure 16. Steady-state error comparison of the applied controllers

Table 5. Performance indices of the torque ripple and energy consumption results under the applied controllers

Case Study	Control Method	Torque Ripple (%)	Energy Consumption Index (%)
Case 1	PI	65	100
	PSO-PI	50	88
	GAO-PI	35	75
	GAO-FOPI	20	62
Case 2	PI	60	100
	PSO-PI	48	90
	GAO-PI	38	80
	GAO-FOPI	25	68
Case 3	PI	70	100
	PSO-PI	58	92
	GAO-PI	45	82
	GAO-FOPI	30	70

Note: PI = proportional-integral; PSO-PI = particle swarm optimization-proportional -integral; GAO-PI = Giant Armadillo Optimization-proportional-integral; GAO-FOPI = Giant Armadillo Optimization-fractional-order proportional integral

Table 5 shows the performance indices of the torque ripple and energy consumption results under the applied controllers. As illustrated in this table, the present GAO-FOPI controller is found to have the minimum torque ripple at each operation condition. This means the motor runs a lot smoother with less mechanical wear. When benchmarked against the traditional PI controller, the reduction in torque ripple is approximately 69%, 58%, and 57% in the cases of no-load, sudden disturbance, and random conditions, respectively. In addition, the new controller has the lowest energy consumption index, indicating that it is more efficient due to quicker responses and fewer oscillations. This drop in energy consumption would be really useful for electric vehicle applications since it means directly better energy efficiency, longer driving range, and higher system reliability.

Table 6 presents the performance indices of the applied controllers, including the best, worst, mean, and standard

deviation of the algorithms based on 20 independent runs.

Table 6. Statistical results of speed independent runs

Metric	Control Method	Best	Worst	Mean	Std.
Steady state error (RPM)	PI	0.151	0.148	0.15	0.07
	PSO-PI	0.158	0.164	0.16	0.06
	GAO-PI	0.151	0.153	0.152	0.02
	GAO-FOPI	0.088	0.092	0.089	0.004
Rise time (sec)	PI	1.62	1.66	1.63	0.04
	PSO-PI	1.58	1.65	1.59	0.02
	GAO-PI	1.601	1.62	1.61	0.019
	GAO-FOPI	1.60	1.611	1.612	0.011
Overshoot (%)	PI	7.82	6.87	7.8	0.95
	PSO-PI	3.76	3.83	3.8	0.07
	GAO-PI	3.49	3.53	3.5	0.04
	GAO-FOPI	2.53	2.521	2.5	0.009

5. CONCLUSION AND FUTURE WORK

This paper introduces a novel speed control strategy for BLDC EVs utilizing the GAO algorithm. The proposed method effectively tunes both PI and fractional-order proportional-integral (FOPI) controllers, which enhances the flexibility in managing the nonlinear dynamics of electric vehicle (EV) drive systems. The performance of the proposed controllers was evaluated under multiple operating scenarios, including no-load conditions, sudden speed variations, and random speed profiles, in order to assess their robustness and tracking capability. The simulation results demonstrate that the GAO-based controllers significantly improve the transient response, damping characteristics, and steady-state accuracy compared to conventional PI and PSO-based methods.

Quantitative analysis shows that the highest overshoot was reduced to 2.13 seconds from 8.4% for the classical PI controller, while the rise time decreased to 1.02 seconds, representing an improvement of nearly 40%. Additionally, the steady-state error was minimized to 0.088 RPM, leading to a more than 58% enhancement in tracking accuracy compared to traditional methods.

The limitation of this work is that the control method does not account for real-time implementation, sensor noise, variations in motor parameters, fluctuations in DC-link voltage, inverter delay, sampling time, or hardware-in-the-loop (HIL) validation. The future work direction is to implement the GAO-FOPI control in experiments using HIL setups or embedded digital controllers to evaluate computational efficiency and real feasibility. Furthermore, the present single-objective optimization framework may be transformed into a multi-objective optimization that simultaneously calculates the accuracy of speed tracking, energy consumption, torque ripple, and thermal stress.

REFERENCES

[1] Sanguesa, J.A., Torres-Sanz, V., Garrido, P., Martinez, F.J., Marquez-Barja, J.M. (2021). A review on electric vehicles: Technologies and challenges. Smart Cities, 4(1): 372-404. https://doi.org/10.3390/smartcities4010022

[2] Vk, A.R., Prasad, V. (2023). Online adaptive gain for passivity-based control for sensorless BLDC motor coupled with DC motor for EV application. IEEE Transactions on Power Electronics, 38(11): 13625-13634. https://doi.org/10.1109/tpel.2023.3288939

[3] Prabhu, N., Thirumalaivasan, R., Ashok, B. (2023). Critical review on torque ripple sources and mitigation control strategies of BLDC motors in electric vehicle applications. IEEE Access, 11: 115699-115739. https://doi.org/10.1109/access.2023.3324419

[4] Poudel, Y.K., Bhandari, P. (2023). Control of the BLDC motor using ant colony optimization algorithm for tuning PID parameters. Archives of Advanced Engineering Science, 2(2): 108-113. https://doi.org/10.47852/bonviewaaes32021184

[5] Naqvi, S.S.A., Jamil, H., Iqbal, N., Khan, S., Lee, D., Park, Y.C., Kim, D.H. (2024). Multi-objective optimization of PI controller for BLDC motor speed control and energy saving in Electric Vehicles: A constrained swarm-based approach. Energy Reports, 12: 402-417. https://doi.org/10.1016/j.egyr.2024.06.019

[6] Sayed, K., El-Zohri, H.H., Ahmed, A., Khamies, M. (2024). Application of tilt integral derivative for efficient speed control and operation of BLDC motor drive for electric vehicles. Fractal and Fractional, 8(1): 61. https://doi.org/10.3390/fractalfract8010061

[7] Shenbagalakshmi, R., Mittal, S.K., Subramaniyan, J., Vengatesan, V., Manikandan, D., Ramaswamy, K. (2025). Adaptive speed control of BLDC motors for enhanced electric vehicle performance using fuzzy logic. Scientific Reports, 15(1): 1-22. https://doi.org/10.1038/s41598-025-90957-6

[8] Sharma, M., Sharma, S., Vajpai, J. (2024). A novel approach to design and analyze fractional order PID controller for speed control of brushless DC motor. Renewable Energy and Sustainable Development, 10(2): 279-293. https://doi.org/10.21622/resd.2024.10.2.903

[9] Kumar, P., Sivaraju, S.S. (2025). An investigation on employing optimized FOPID controller to expand the performance of BLDC motor drives. Electrical Engineering, 107(8): 10981-10996. https://doi.org/10.1007/s00202-025-03072-0

[10] Aziz, M.A.A., Saidon, M.S., Romli, M.I.F., Othman, S.M., Mustafa, W.A., Manan, M.R., Aihsan, M.Z. (2023). A review on BLDC motor application in Electric Vehicle (EV) using battery, supercapacitor and hybrid energy storage system: Efficiency and future prospects. Journal of Advanced Research in Applied Sciences and Engineering Technology, 30(2): 41-59. https://doi.org/10.37934/araset.30.2.4159

[11] Kethiri, M.F., Charrouf, O., Betka, A., Tibermacine, I.E., Napoli, C. (2025). Hybrid fuzzy-PSO based self-tuning fractional order PI controller for BLDC motors in electric vehicles: Comparative analysis and experimental validation. Journal of Vibration and Control. https://doi.org/10.1177/10775463251374112

[12] Dakheel, H.S., Abdullah, Z.B., Shneen, S.W. (2023). Advanced optimal GA-PID controller for BLDC motor. Bulletin of Electrical Engineering and Informatics, 12(4): 2077-2086.

[13] Dutta, P., Nayak, S.K. (2021). Grey wolf optimizer based PID controller for speed control of BLDC motor. Journal of Electrical Engineering & Technology, 16(2): 955-961. https://doi.org/10.1007/s42835-021-00660-5

[14] Malla, S.G., Malla, P., Malla, J.M.R., Singla, R.,

Choudekar, P., Koilada, R., Sahu, M.K., Ruchira, A. (2022). Whale optimization algorithm for PV based water pumping system driven by BLDC motor using sliding mode controller. IEEE Journal of Emerging and Selected Topics in Power Electronics, 10(4): 4832-4844. https://doi.org/10.1109/jestpe.2022.3150008

[15] Alsayyed, O., Hamadneh, T., Al-Tarawneh, H., Alqudah, M., Gochhait, S., Leonova, I., Malik, O.P., Dehghani, M. (2023). Giant armadillo optimization: A new bio-inspired metaheuristic algorithm for solving optimization problems. Biomimetics, 8(8): 619. https://doi.org/10.3390/biomimetics8080619

[16] George, M.A., Kamat, D.V., Indiran, T. (2021). OTA-C realization of an optimized FOPID controller for BLDC motor speed control. IETE Journal of Research, 69(1): 118-136. https://doi.org/10.1080/03772063.2021.1951380

[17] Temir, A., Durmuş, B. (2023). Equilibrium optimizer based FOPID control of BLDC motor. European Journal of Science and Technology, 51: 153-161. https://doi.org/10.31590/ejosat.1256908

[18] Wang, T., Wang, H., Wang, C., Hu, H. (2022). A novel PID controller for BLDCM speed control using dual fuzzy logic systems with HSA optimization. Scientific Reports, 12(1): 1-19. https://doi.org/10.1038/s41598-022-15487-x

[19] Dasari, M., Reddy, A.S., Kumar, M.V. (2022). Modified Luo converter based FOPID controller for torque ripple minimization in BLDC drive system. Journal of Ambient Intelligence and Humanized Computing, 14(6): 7091-7108. https://doi.org/10.1007/s12652-021-03562-6

[20] Ghamari, S.M., Ghahramani, M., Habibi, D., Aziz, A. (2025). Design of a robust adaptive cascade fractional-order proportional-integral-derivative controller enhanced by reinforcement learning algorithm for speed regulation of brushless DC motor in electric vehicles. Energies, 18(19): 5056. https://doi.org/10.3390/en18195056

[21] Kumar, R., Kumar, A. (2025). A novel technique for optimization of BLDC-based dual-motor electric vehicles using adaptive BFO-based PID controller. International Journal of Power Electronics and Drive Systems (IJPEDS), 16(1): 10-24. https://doi.org/10.11591/ijpeds.v16.i1.pp10-24

[22] Jabari, M., Ekinci, S., Izci, D., Bajaj, M., Zaitsev, I. (2024). Efficient DC motor speed control using a novel multi-stage FOPD(1 + PI) controller optimized by the Pelican optimization algorithm. Scientific Reports, 14(1): 1-25. https://doi.org/10.1038/s41598-024-73409-5

[23] Krishnamoorthy, S.K., Das, N., Gudimetla, P., Emami, K. (2024). Enhanced speed control for BLDC motors using WOA-integrated PID controller optimization. IEEE Access, 12: 162465-162475. https://doi.org/10.1109/access.2024.3480349

[24] Kumar, R., Bera, C., Kumar, A. (2024). Optimization of BLDC-based electric vehicles: Vehicle dynamics modelling through dual-motor approach and designing a novel augmented TLBO algorithm for PID control. Engineering Research Express, 6(2): 025334. https://doi.org/10.1088/2631-8695/ad45b3

[25] Joshi, D., Deb, D., Giri, A.K. (2025). Metaheuristic adaptive input output feedback linearization control for BLDC motor drive. IEEE Transactions on Consumer Electronics, 71(2): 6120-6130. https://doi.org/10.1109/tce.2025.3553388

[26] Lins, A.W., Krishnakumar, R. (2022). Tuning of PID controller for a PV-fed BLDC motor using PSO and TLBO algorithm. Applied Nanoscience, 13(4): 2911-2934. https://doi.org/10.1007/s13204-021-02272-x

[27] Abdolhosseini, M., Abdollahi, R. (2023). Performance analysis of PID controller-based metaheuristic optimisation algorithms for BLDC motor. Australian Journal of Electrical and Electronics Engineering, 20(4): 400-411. https://doi.org/10.1080/1448837x.2023.2249205

[28] Sarma, H., Bardalai, A. (2023). An intelligent PID controller tuning for speed control of BLDC motor using driving training-based optimization. International Journal of Power Electronics and Drive Systems (IJPEDS), 14(4): 2474-2486. https://doi.org/10.11591/ijpeds.v14.i4.pp2474-2486

[29] Jarkas, A.M., Doss, M.A.N. (2025). Optimized PI controller tuning for improved performance in BLDC motor speed control using heuristic adaptive lyrebird optimization algorithm. Electrical Engineering, 107(7): 9467-9487. https://doi.org/10.1007/s00202-025-02984-1

[30] Rani, M.A.F., Sankaragomathi, B. (2015). Performance enhancement of PID controllers by modern optimization techniques for speed control of PMBL DC motor. Research Journal of Applied Sciences, Engineering and Technology, 10(10): 1154-1163. https://doi.org/10.19026/rjaset.10.1883

[31] Pakdeeto, J., Wansungnoen, S., Areerak, K., Areerak, K., Areerak, K. (2023). Optimal speed controller design of commercial BLDC motor by adaptive tabu search algorithm. IEEE Access, 11: 79710-79720. https://doi.org/10.1109/access.2023.3300233

[32] Arwa, A.A., Ietiqal, M.A., Salam, J.Y., Center, M.O.E.R.E. (2025). Bald eagle search algorithm based PI control method for speed control of BLDC motor drives. AIMS Electronics and Electrical Engineering, 9(4): 565-588. https://doi.org/10.3934/electreng.2025025

[33] Tiwari, V.K., Kumar, A., Yadav, S., Tiwari, N. (2025). Improved performance of solar, battery, and supercapacitor powered BLDC motor-based electric vehicle using nature-inspired optimizations. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 1-18. https://doi.org/10.1007/s40998-025-00899-y

[34] Kroičs, K., Būmanis, A. (2024). BLDC motor speed control with digital adaptive PID-fuzzy controller and reduced harmonic content. Energies, 17(6): 1311. https://doi.org/10.3390/en17061311

[35] Subbarao, M., Dasari, K., Duvvuri, S.S., Prasad, K., Narendra, B., Krishna, V.M. (2023). Design, control and performance comparison of PI and ANFIS controllers for BLDC motor driven electric vehicles. Measurement: Sensors, 31: 101001. https://doi.org/10.1016/j.measen.2023.101001.

[36] Mondal, S., Mitra, A., Chattopadhyay, M. (2015). Mathematical modeling and simulation of Brushless DC motor with ideal Back EMF for a precision speed control. In 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, pp. 1-5. https://doi.org/10.1109/ICECCT.2015.7225944

[37] Cherif, K., Sahbani, A., Ben Saad, K. (2024). Performance evaluation of PI and sliding mode control

for PMSM in applications for electric vehicles. Engineering, Technology & Applied Science Research, 14(4): 15464-15470. https://doi.org/10.48084/etasr.7172
[38] Challoob, A.F., Bin Rahmat, N.A., Ramachandaramurthy, V.K.A., Humaidi, A.J. (2025). Robust energy management system for electric vehicle. International Review of Applied Sciences and Engineering, 16(1): 98-117. https://doi.org/10.1556/1848.2024.00839