AIMS Mathematics

AIMS Mathematics, 11(8): 24552–24581.
DOI: 10.3934/math.2026990
Received: 02 May 2026
Revised: 14 July 2026
Accepted: 30 July 2026
Published: 11 August 2026

https://www.aimspress.com/journal/Math

Research article

Multi-objective metaheuristic optimization of PI controller gains for BLDC motor control in electric vehicle applications

Hani Albalawi[1,2,*]

[1] Zero Emission Technologies Innovation Center, University of Tabuk, Tabuk, 47913, Saudi Arabia
[2] Department of Electrical Engineering, Faculty of Engineering, University of Tabuk, Tabuk, 47913, Saudi Arabia

* **Correspondence:** Email: halbala@ut.edu.sa.

Abstract: Brushless DC (BLDC) motors are widely employed in electric vehicle (EV) propulsion systems because of their high efficiency, compact structure, and favorable dynamic performance. However, achieving accurate and energy-efficient speed regulation remains challenging when the drive operates under time-varying load torque caused by road gradient and operating-resistance variation. In this paper, I present a multi-objective optimization framework for tuning the proportional-integral (PI) controller gains of a BLDC motor drive for EV applications, with the dual objective of improving speed-tracking performance and reducing energy consumption. To balance computational efficiency and practical relevance, a two-stage evaluation strategy was adopted in which Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) were first used to optimize the controller gains over a short 0.4 s simulation horizon, after which the resulting gains were validated over a 120 s drive cycle under inclination-equivalent loading conditions. During the optimization stage, the Ziegler- Nichols (ZN) controller produced a Root Mean Square Error (RMSE) of 0.29420 and an energy consumption of 61.5263 J, whereas PSO reduced these values to 0.12971 and 60.727 J, and GA achieved 0.13052 and 60.729 J, respectively. These results corresponded to RMSE improvements of 55.912% and 55.636%, together with energy reductions of 1.2991% and 1.2958% for PSO and GA, respectively, relative to the ZN baseline. Over 10 independent runs, PSO and GA yielded mean fitness values of 0.65933 and 0.66100, with corresponding standard deviations of 0.00092 and 0.00118, respectively. Under the 120 s full-cycle validation, all three controllers exhibited near-ideal performance in the no-load case, whereas under the varying-load case, the optimization-based controllers outperformed the baseline controller, with PSO achieving the lowest overall loaded-case fitness value. The results demonstrated that metaheuristic multi-objective tuning

provides an effective and practical framework for improving BLDC motor control in EV drive systems, yielding major gains in tracking performance together with modest but consistent energy savings.

Keywords: BLDC motor; electric vehicle; PI control; Particle Swarm Optimization; Genetic Algorithm; multi-objective optimization, energy efficiency
Mathematics Subject Classification: 90C59, 90C30, 34A08, 68T20

1. Introduction

The increasing number of electric vehicles has resulted in greater need for effective propulsion control systems for the EVs [1–3]. The dynamic response, power consumption, comfort, and reliability of these systems highly depend on the motor and the controller in an EV drive [4,5]. The BLDC motor, among others utilized in propulsion systems, offers an appealing prospect in view of its efficiency, small size, excellent torque/weight ratio, as well as reduced maintenance due to electronic commutation, thus becoming applicable in diverse areas like electric vehicles [6]. However, achieving accurate speed regulation in real-life applications remains difficult due to nonlinearity and the dependence of its characteristics on operating conditions [7,8]. Operating conditions in EVs are significantly affected by such factors as the inclination angle of the road, resistance, and load changes that result in varying demands for torque [9,10]. Despite the widespread use of PI controllers due to simplicity of their design and low price, the effectiveness of this control method heavily depends on the selection of proportional and integral gains. Classic tuning strategies like the ZN approach may offer acceptable performance initially, but they usually result in system overshoot, oscillations, and poor disturbance rejection when the motor is loaded by nonlinear and dynamic forces [11]. The above problems tend to worsen when the system is required to sustain stable speed against load torque changes due to slope-related issues. For these reasons, optimization-based controller tuning has become an important direction for improving the dynamic response and energy efficiency of electric-drive systems [12,13]. Popular techniques include PSO, which converges rapidly and requires simple implementation, and GA, which explores the search space effectively via selection, crossover, and mutation [14–16]. While both optimization algorithms have proven highly promising for tuning controllers, only a limited number of studies have provided a unified comparison among ZN, PSO, and GA for PI tuning of BLDC-driven EV systems. The other major assumption made by most researchers while evaluating the performance of the controllers is that of the disturbance inputs used to represent the change in load torque when the vehicle is moving at constant speed. In reality, this variation can be better understood by taking into account the angle of the slope and corresponding resistance force, which have a direct impact on the required torque of the drive system.

Based on the identified research gap, I present a multi-objective optimization framework for tuning the PI controller gains of a BLDC motor drive for EV applications. Three tuning strategies are compared within a common simulation framework: the conventional ZN method, PSO, and GA. The optimization objective combines speed-tracking accuracy and energy efficiency through a normalized weighted fitness function. In order to ensure an appropriate compromise between computational effort and realistic applicability, a two-step procedure is followed wherein PSO and

GA are initially applied using a short 0.4 s time period to determine appropriate controller parameters, after which the determined parameters are tested for 120 s using an incline-equivalent load torque profile. Thus, I present a comparative analysis of heuristic and meta-heuristic approaches to the problem of tuning a PI controller for BLDC-based EVs. The general BLDC motor speed-control configuration considered in this study is illustrated in Figure 1.

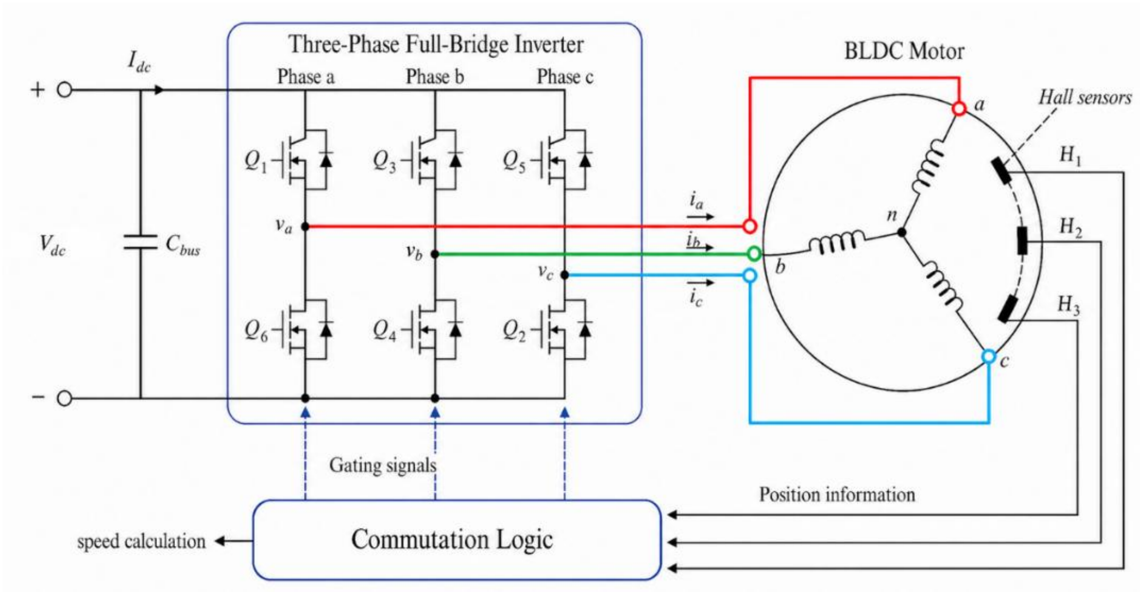

Figure 1. General block diagram of BLDC motor speed control system used for analysis in this study.

The rest of the paper is presented in the following manner. Literature review and research gap identification are addressed in Section 2. Mathematical modeling and problem definition are covered in Section 3. In section 4, I give information about methodology and experimental setup. Comparative results regarding tracking performance, statistical aspects, convergence features, optimization landscapes, and computational complexity are provided in Section 5. I conclude concludes in Section 6.

2. Literature review

Researchers dealing with BLDC motors for driving purposes have mostly considered aspects such as speed regulation, disturbance rejection, and power efficiency [4–8]. The problem in this regard is not just about achieving good speed regulation; it is also important to keep system performance stable even when the drive system experiences nonlinear load disturbances resulting from road conditions and other forms of resistance [4,5,8,9]. Consequently, controller tuning becomes an important aspect in designing efficient BLDC propulsion systems [4–6]. The conventional techniques used to regulate the speed of BLDC motors employ PI and PID controllers owing to their ease of implementation and reduced cost [6–8]. In the context of classical tuning techniques, ZN tuning continues to be one of the most popular due to the availability of a relatively straightforward process of finding the initial values of the gain parameters. Unfortunately, investigations have demonstrated that such tuning yields large overshoot, oscillations, and poor flexibility with regard to nonlinear and time-dependent cases [7,8,11]. To address these issues, controller synthesis based on optimization has been a subject of intensive research during recent years [7,8,10–17]. PSO has been one of the most popular metaheuristics for tuning PI/PID controllers, offering a simple and fast algorithm with good exploitative properties in continuous domains [14–16]. Scientific literature also reveals an increase in interest toward the application of optimization techniques for improved performance of BLDC control in electric vehicles [8,11,17]. GA presents an alternative approach to searching, which is based on principles of selection, crossover, and mutation, diversity maintenance and better global search performance [11–13]. It has been found through empirical comparison studies that PSO has a greater ability to converge faster compared to GA, whereas GA has an ability to generate robust solutions in comparison to PSO [11,14,16]. Optimization studies carried out in energy and power system application domains have shown that swarm optimization, hybrid optimization, and optimization techniques based on machine learning techniques are effective in handling optimization problems related to nonlinearity and constraints in practical engineering applications [18–21]. In this work, this broader optimization perspective is applied to a more specific BLDC motor-control problem by comparing ZN, PSO, and GA for PI gain tuning under EV-relevant variable-load conditions.

The literature highlights the growing importance of multi-objective optimization in electric-drive control, since practical EV propulsion systems must simultaneously satisfy speed-tracking, disturbance-rejection, and energy-efficiency requirements [5,8,11]. However, the following limitations are observed from the literature. First, many articles are dedicated to the development of only one of the control strategies, including fuzzy, adaptive, or sensorless control without comparing conventional and meta-heuristic approaches for tuning of PI controllers [8,22–24]. In [25], an active disturbance rejection control (ADRC) was used under uncertainties and sensor noise to improve the performance of ADRC. In addition to these EV-oriented BLDC studies, several related works have examined conventional BLDC motor modelling, PID control, inverter configuration [26–28], while PID tuning, power-quality improvement, and torque-ripple reduction in BLDC and electric-drive systems [29–31]. Other studies have also demonstrated the usefulness of optimization-based fuzzy control, fault-tolerant traction-drive control [32,33], while neural-network-based propulsion control, and adaptive maximum torque-per-ampere strategies in motor-drive applications [34,35]. Although these works confirm the growing importance of intelligent and optimization-assisted motor-control methods, they generally focus on a specific controller, converter structure, or drive topology. They do not provide a unified comparison of ZN, PSO, and GA for PI gain tuning of a BLDC motor drive under the same objective function, search bounds, and inclination-equivalent EV loading conditions. Second, despite a wide coverage of optimization for designing control systems, comparatively fewer researchers have compared ZN, PSO, and GA tuning approaches for designing BLDC motor PI controllers under equivalent search domains and objective function sets [11,16,17,29]. Third, researchers mostly use some general or simplified load disturbances, while the influence of road gradient and resistance effects should be considered during EV drive operation [4,5,9,10]. Although metaheuristic optimization algorithms have been proposed for engineering optimization problems including fractional order [36–38] hybrid [39] and adaptive approaches [40], primary objective of this study is to provide a rigorous and controlled comparison between the conventional ZN method and two representative, well-established metaheuristic algorithms, namely PSO and GA, under identical objective functions, search spaces, parameter bounds, and simulation conditions These algorithms are selected because they represent two fundamentally different optimization paradigms, swarm intelligence and evolutionary computation and have been widely employed as benchmark techniques for PI controller tuning in BLDC motor applications. Consequently, I focus on establishing a fair comparative baseline rather than conducting an exhaustive benchmarking study across all recently developed optimization algorithms. Therefore, motivated by this gap, I provide a comprehensive multi-objective tuning scheme for PI controllers applied to BLDC motors for driving vehicles and comparing ZN, PSO, and GA methods of tuning under inclination-equivalent loading of EV drives. Table 1 presents some relevant research work.

Table 1. Comparison of representative prior studies and this work.

Ref.	Focus	Method	Validation	Gap	Present Work
[4]	EV traction control review	Comprehensive Review	Review	Reviews advanced EV traction motor control strategies but does not investigate optimization-based PI controller tuning or provide a comparative evaluation of conventional and metaheuristic PI tuning methods for BLDC motor drives.	Develops a unified multi-objective PI controller tuning framework for EV-oriented BLDC motor drives by systematically comparing Ziegler–Nichols, PSO, and GA under identical optimization settings and inclination-equivalent loading conditions.
[6]	BLDC control review	Comprehensive review of advanced control technique	Review	Reviews BLDC control strategies without addressing optimization-based multi-objective PI controller tuning.	Proposes and evaluates EV-specific PI controller tuning considering both tracking performance and energy efficiency.
[8]	BLDC control for EVs	Fuzzy logic speed control	Journal study	Focuses on fuzzy logic speed control rather than optimization-based multi-objective PI controller tuning.	Optimizes PI controller gains using metaheuristic algorithms under a multi-objective formulation.
[16]	Controller tuning	PSO-based PID autotuning	Journal study	Develops an NSPSO-based PID autotuning method for DC motor control but does not investigate comparative optimization of PI controllers for BLDC motor drive developed for EV BLDC applications.	Provides a unified comparative framework between ZN, PSO, and GA for PI controller tuning in EV BLDC drives.

Continued on next page

Ref.	Focus	Method	Validation	Gap	Present Work
[22]	BLDC speed control	Sensorless speed control using variable-slope SMO	IEEE TIA journal study	Presents a variable-slope sliding mode observer for sensorless speed control of high-speed BLDC motors but does not investigate optimization-based PI controller tuning or compare conventional and metaheuristic tuning techniques.	Develops a comprehensive comparative framework for PI controller tuning in BLDC motor drives by evaluating Ziegler–Nichols, PSO, and GA under a unified multi-objective performance criterion to achieve superior dynamic response and control accuracy.
[23]	Light-duty EV BLDC drive	Position-sensorless single-stage BLDC drive	IEEE TIA journal study	Focuses on sensorless BLDC motor drive control rather than optimization-based multi-objective PI controller tuning.	Investigates optimization-based PI tuning with simultaneous consideration of tracking accuracy and energy consumption.
[32]–[35]	Intelligent and adaptive motor-drive control	Optimization-based fuzzy control, fault-tolerant traction control, neural-network propulsion control, adaptive MTPA control	Journal studies	Focus on intelligent and adaptive motor control rather than optimization-based multi-objective PI controller tuning for BLDC motor drives.	Establishes a unified benchmark using conventional and metaheuristic PI tuning methods under identical optimization settings and inclination-equivalent loading conditions.
Present Work	EV BLDC drive	ZN + PSO + GA PI tuning	Simulation + statistical analysis	Addresses the above research gaps through a unified multi-objective optimization framework.	Provides a fair comparison of ZN, PSO, and GA using identical objective functions, search spaces, parameter bounds, and simulation conditions, while simultaneously optimizing speed-tracking accuracy and energy consumption

As shown in Table 1, researchers have investigated BLDC motor control, EV propulsion strategies, and optimization-based controller tuning, with most focusing on either a single control method, a review perspective, or a specific intelligent-control implementation rather than a unified comparison under identical tuning conditions. In addition, many researchers employ simplified disturbance assumptions and do not explicitly evaluate PI gain tuning under inclination-equivalent EV loading conditions. In this context, the novelty of this work lies in three aspects. First, I give a comprehensive comparative study of the ZN, PSO, and GA approaches to PI gain tuning for a BLDC motor drive using identical search ranges, performance criteria, and MATLAB/Simulink environment. Second, the controller tuning problem is posed as a multi-objective optimal speed-energy problem, and the resultant controllers are validated in full cycle operation with no load and variable load cases. Third, I complete the controller tuning study with a statistical test, convergence study, landscape visualization, and computation cost analysis of heuristic and metaheuristic PI tuning for BLDC-powered EVs.

3. Mathematical modeling and problem formulation

To design an effective speed-control strategy for a BLDC motor-driven EV system, it is necessary to establish a mathematical model of the inverter stage, motor dynamics, and controller structure. The drive system considered in this study is supplied from a 48 V DC source and consists of a three-phase voltage-source inverter, a BLDC motor, hall-sensor-based commutation logic, and a closed-loop PI speed controller, as illustrated in Figure 2. The mathematical formulation presented in this section provides the basis for comparing the conventional ZN tuning method with the optimization-based tuning approaches adopted later in the manuscript.

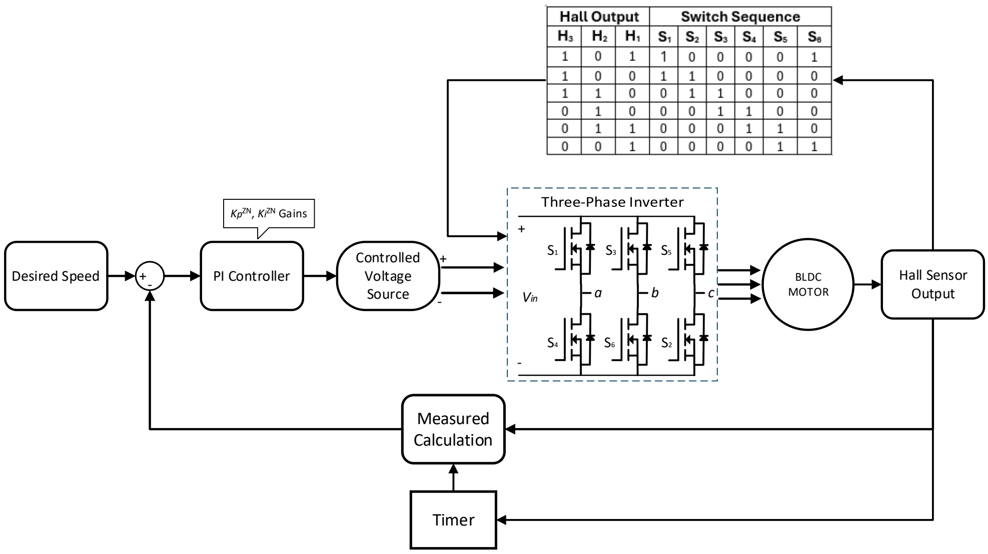

Figure. 2. Baseline closed-loop BLDC motor drive system with Ziegler–Nichols-tuned PI control.

3.1. Three-phase inverter modeling

The BLDC motor is driven by a three-phase voltage source inverter circuit comprising six power switches. Under the proposed commutation technique, the upper inverter leg switches namely S_1, S_3, and S_5 control the voltages fed to the three BLDC phases a, b, and c, respectively. Given a DC link voltage V_{dc}, the voltages supplied by the inverter across each phase could be mathematically represented as functions of the upper inverter leg switching operations and therefore provide the electrical link between the DC power source and BLDC motor. The inverter stage serves to transform the DC source into three controlled voltages needed to run the BLDC motor. The major parameters of the inverter in the MATLAB/Simulink simulation are outlined in Table 2. The phase voltages could be described mathematically by Eqs (1)–(3):

$$V_a = \frac{V_{dc}}{3}(2\mathrm{S}_1 - \mathrm{S}_3 - \mathrm{S}_5), \tag{1}$$

$$V_b = \frac{V_{dc}}{3}(2\mathrm{S}_3 - \mathrm{S}_1 - \mathrm{S}_5), \tag{2}$$

$$V_c = \frac{V_{dc}}{3}(2\mathrm{S}_5 - \mathrm{S}_1 - \mathrm{S}_3), \tag{3}$$

where S_1, S_3, and S_5 are binary switching functions associated with the upper switches of the three inverter legs. These equations may be written in compact matrix form as in Eq (4).

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_3 \\ \mathrm{S}_5 \end{bmatrix}. \tag{4}$$

Table 2. Inverter parameters with protection Diodes.

DC-link voltage ($\boldsymbol{V_{dc}}$)	Switching device	Forward voltage ($\boldsymbol{V_f}$)	On-state resistance ($\boldsymbol{R_{on}}$)	Off-state conductance ($\boldsymbol{G_{off}}$)	Threshold voltage ($\boldsymbol{V_{th}}$)
48 V	IGBT	0.8 V	0.001 Ω	1×10^{-5} S	0.5 V

This formulation establishes the relationship between the DC source and the inverter output voltages applied to the motor phases. Since the controller acts on the drive system through the inverter, these equations provide the electrical basis for linking the PI control signal to the BLDC motor dynamics in the EV propulsion system.

3.2. BLDC motor modeling

The electrical behavior of the BLDC motor is described by the stator voltage equation of each phase. For phase n, where $n \in \{a, b, c\}$, the voltage equation is expressed as in 5:

$$V_n = R_s i_n + L\frac{di_n}{dt} + e_n, \quad (n = a, b, c), \tag{5}$$

where V_n is the phase voltage, R_s is the stator resistance, i_n is the phase current, L is the stator inductance, and e_n is the back electromotive force of the corresponding phase. The above formula shows the electrical behavior of the BLDC motor when excited by an inverter and forms the foundation for understanding the relationship between the applied phase voltages and the generated stator currents.

The mechanical dynamics of the BLDC motor are governed by the balance between the developed electromagnetic torque and the opposing load torque, and may be written as in Eq (6):

$$T_e(t) - T_L(t) = J\frac{d\omega(t)}{dt} + B\,\omega_m(t). \tag{6}$$

Here T_L is the load torque, T_e is the electromagnetic torque, B is the viscous friction coefficient, J is the rotor inertia, and ω_m is the rotor angular speed. This equation shows that the motor speed is determined by the net torque available after overcoming the external load and mechanical losses.

In this study, the load torque is regarded as the effect of the road inclination and resistance change rather than a change in the mass and weight of the passengers or the car itself when moving. From the perspective of physics, it is a reasonable notion, particularly if consider the movement of an electric car at a constant speed. An increase in the opposite torque means that the vehicle goes up on the road and is facing greater resistance. Consequently, a growing value of T_L suggests a great slope of the road or greater resistance faced. On the other hand, a decline in T_L suggests that the road is flat or less resistance being faced. The load-torque profile adopted in this study is an inclination-equivalent representation rather than a direct conversion of a specific road slope angle. In practical EV operation, the propulsion system is subjected to a combination of gravitational forces associated with road gradients, rolling resistance, aerodynamic drag, and drivetrain mechanical losses. Instead of modeling these components individually, their combined effect is represented by an equivalent external load torque applied to the motor shaft. Consequently, the load variation from 0 to –4 N·m represents progressively increasing propulsion demand, ranging from near-level road operation to relatively demanding uphill driving conditions. This equivalent loading approach enables a systematic evaluation of controller robustness under progressively increasing disturbance levels while maintaining a computationally efficient simulation framework.

The BLDC motor chosen to conduct the experiments in this work is the Fulling Motor FL86BLS98. Its specifications are described in Table 3.

Table 3. BLDC motor parameters used in simulation.

Nominal Voltage	**Stator Resistance (R_s)**	**Stator Inductance (L)**	**Permanent Magnet Flux (λ)**	**Rotor Inertia (J)**	**Pole Pairs (P)**
48 V	0.1 Ω	0.24 mH	0.02167 Wb	0.00016 kg.m^2	4

3.3. *PI controller formulation*

The BLDC motor speed is regulated using a proportional–integral controller. The control law is expressed as in Eq (7):

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau, \tag{7}$$

where $u(t)$ is the controller output,

$$e(t) = \omega_{ref}(t) - \omega_{act}(t)$$

is the speed error, and K_p and K_i are the proportional and integral gains, respectively. The control signal affects the inverter-based BLDC drive, which then controls the electromagnetic torque and the speed. Using such a feedback mechanism, the PI controller offers the required correction to minimize the error between the desired and the observed speed. The performance of the closed loop system greatly depends on the chosen values of K_p and K_i. If the tuning is not properly performed, the BLDC motor drive can demonstrate undesirable characteristics such as overshooting, oscillations, poor response when operating at various loads, as well as wasting power. As our goal of this research is to enhance the dynamic performance and increase the efficiency, the tuning of the controller gains becomes the key optimization task.

3.4. Multi-objective problem formulation

Controller design is treated as a multiple objective optimization problem wherein the control parameters (gains K_p and K_i) are chosen to minimize speed tracking errors and power consumption. The former is achieved by minimizing the difference between the reference speed and actual speed of the motor using the RMSE defined as in Eq (8):

$$RMSE = \sqrt{\frac{1}{T}\int_0^T \left(\omega_{ref}(t) - \omega_{act}(t)\right)^2 dt}, \tag{8}$$

where T is the time horizon employed for the simulation when optimizing. This represents the error between the motor speed and the reference speed and indicates how well the speed controller performs. The second criterion is the total energy used by the electric motors during the simulation period and is calculated by Eq (9) as:

$$Energy = \int_0^T P_{in}(t).dt, \tag{9}$$

where $P_{in}(t)$ denotes the instantaneous DC input power of the drive system. Thus, the energy term represents the cumulative electrical energy drawn over the selected simulation interval. To achieve the above two objectives, a normalized weight is proposed as in Eq (10):

$$J(K_p, K_i) = \alpha\left(\frac{RMSE}{RMSE_{norm}}\right) + \beta\left(\frac{Energy}{Energy_{norm}}\right), \tag{10}$$

where α and β are the weights of the speed-tracking and energy components, respectively, and $RMSE_{norm}$ and $Energy_{norm}$ are normalization factors which are obtained based on the output of the initial system configuration. In this research, the weights are selected as $\alpha = 0.6$ and $\beta = 0.4$, such that the speed-tracking component has slightly more importance than the energy component, but the latter is highly weighted as well. This will help the optimization algorithms find gain values for the controller that provide an optimal trade-off between performance and energy consumption. The adopted weighting combination of 0.6/0.4 is selected as a practically balanced scalarization, and its influence on the optimization outcome is further examined through the sensitivity analysis reported in Section 5.3.

This optimization process occurs under certain constraints on gain values to make sure that the optimization is physically viable, and the stability of the controller is maintained. In this case, the proportional gain can be found between $1 \leq K_p \leq 10$, while the integral gain can be found between $100 \leq K_i \leq 2000$. These two intervals constitute the space in which the gains must be found using PSO and GA algorithms. Under this setup, the ZN tuning technique constitutes the baseline method.

4. Methodology and experimental setup

The methodology adopted in this study is based on a comparative PI tuning framework for BLDC motor drives operating under representative EV loading conditions. Three approaches are examined: The conventional ZN method as a benchmark, and the metaheuristic PSO and GA methods as optimization-based alternatives. All three controllers are compared for the same BLDC motor model, inverter setup, constraints on gain values, and objective function. The problem of gain optimization is considered as a multi-objective optimization (MOO) problem with performance measures based on speed-tracking error and energy consumption. To ensure reasonable computation time and relevance, the optimization algorithms run for only 0.4 s, and the gains thus obtained are further validated for 120 s under equivalent inclined-load torque condition.

4.1. Ziegler–Nichols (ZN) tuning method

The tuning approach for the basic control system used in this work is determined by employing the tuning techniques described by the ZN algorithm within the closed loop design of the BLDC motor drive as illustrated in Figure 2, where the error in the speed is calculated by comparing the reference speed to that of the motor. The control output controls the voltage source that feeds the three-phase inverter, while Hall sensor output and corresponding calculations make up the rest of the speed control loop. This structure provides the reference configuration against which the optimization-based tuning methods are evaluated. The ZN approach is adopted to estimate the initial values of the proportional and integral gain settings for the PI controller and hence serve a classical reference point for this study. While the ZN tuning algorithm provides an easy-to-follow tuning process, it may have some limitations when working under nonlinear and nonstationary conditions. The ZN tuning algorithm is therefore considered as the initial controller tuning technique to compare its performance against the controllers optimized by the PSO and GA approaches under similar operating conditions for EV applications.

4.2. Optimization methods and algorithms

To overcome the limitations of the heuristic baseline tuning method, the closed-loop BLDC motor drive model is extended into an optimization-based framework, as illustrated in Figure 3. In this framework, a metaheuristic optimization module is incorporated into the tuning loop to determine the PI controller gains K_p and K_i by minimizing a multi-objective fitness function based on speed-tracking error and energy consumption. These optimum gain values are then fed into the PI controller, thus creating a direct connection between the optimization process and the model of the inverter-driven BLDC motor drive. Such a configuration is used by both PSO and GA, whereas the ZN method remains as the conventional tuning technique for comparison purposes. As PSO and GA algorithms operate under the same gain limits, simulations, and assessment criteria, the comparison of the two becomes a fair ground for evaluation. It should be emphasized that PSO and GA solve exactly the same scalarized multi-objective optimization problem formulated in Section 3.4. In both algorithms, the objective is to minimize the weighted normalized fitness function comprising the speed-tracking error (RMSE) and energy consumption. Consequently, the distinction between the two methods lies exclusively in their search strategies rather than in the optimization objective. This common optimization formulation ensures that the comparative evaluation reflects differences in optimization capability rather than differences in problem definition.

4.2.1. PSO

For the search space of PI parameters, the upper and lower boundaries are defined to guarantee physical and control implementation feasibility. The proportional parameter, K_p, searches for its optimal value in the range of [1,10], whereas K_i searches in the interval [100, 2000]. In this way, in the bounded search space, the PSO algorithm uses 15 particles for searching through the defined search space during 50 iterations. Each particle stands for one potential controller characterized by a gain pair $[K_p,\ K_i]$, whose performance is evaluated using the widely used multi-objective performance criterion. As an additional measure to achieve stability during convergence, the linearly decreasing PSO inertia weight ω_{PSO} is defined as in Eq (11).

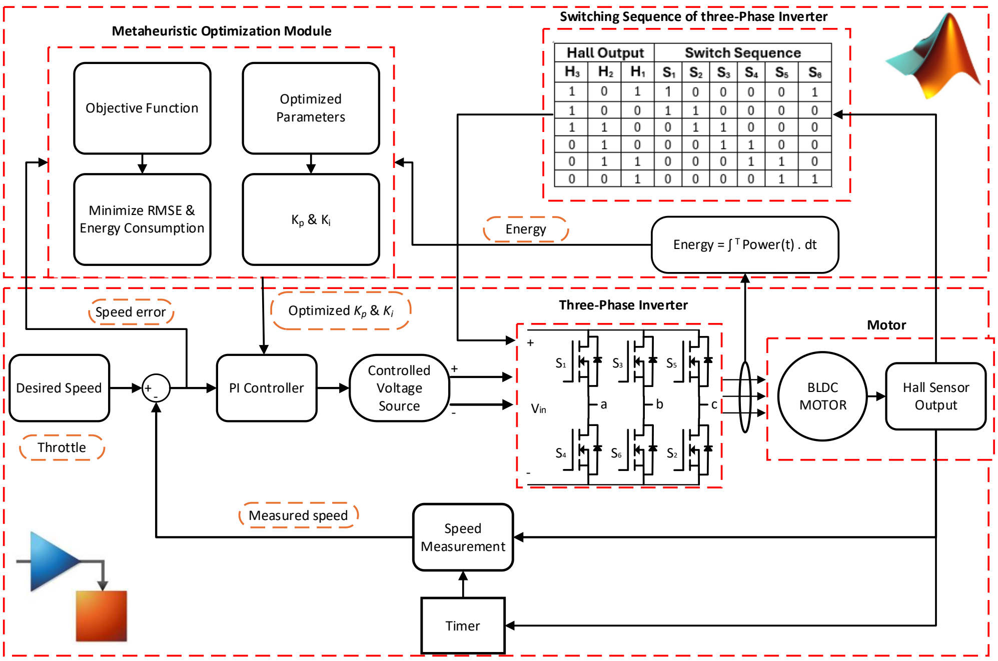

Figure. 3. Optimization-based BLDC motor drive framework in which the PI controller gains are tuned using metaheuristic algorithms through multi-objective minimization of RMSE and energy consumption.

$$\omega_{PSO,curr} = \omega_{PSO,max} - \left(\omega_{PSO,max} - \omega_{PSO,min}\right).\frac{iter}{max\_iter}, \tag{11}$$

where $\omega_{PSO,max} = 0.9$ and $\omega_{PSO,min} = 0.4$. This strategy enables the advantage of searching through the solution space in the early stages of the optimization process; however, in the latter stage, more exploitation of the solution space is expected. Acceleration coefficients of cognition and society are equal to 1.5, which helps provide a good combination of the influence of the personal best and global best solutions on the particle movement. During each iteration, the candidate gain values are tested using the simulation model, and then the performance is checked, along with updating the personal best and global best solutions. The particles then move through the search space according to the standard velocity and position update mechanism until the stopping criterion is satisfied.

4.2.2. PSO pseudocode and flow chart

The computational processes involved in the tuning scheme based on PSO are explained via Algorithm 1, whereas the operational process involved in PSO can be visualized through Figure 4. In the pseudocode, one can see that the process of gaining the optimal gain involves an iterative scheme, which starts with initializing the swarm particles and their velocity, evaluating the gain candidates, finding the personal best and global best solutions, and then optimizing the position of swarm particles until the process converges. The flowchart complements this description by showing the operational sequence of the PSO loop, including initialization, simulation-based evaluation, fitness assessment, update steps, and convergence checking. Together, Algorithm 1 and Figure 4 provide a complete procedural representation of the PSO-based multi-objective tuning strategy used in this study.

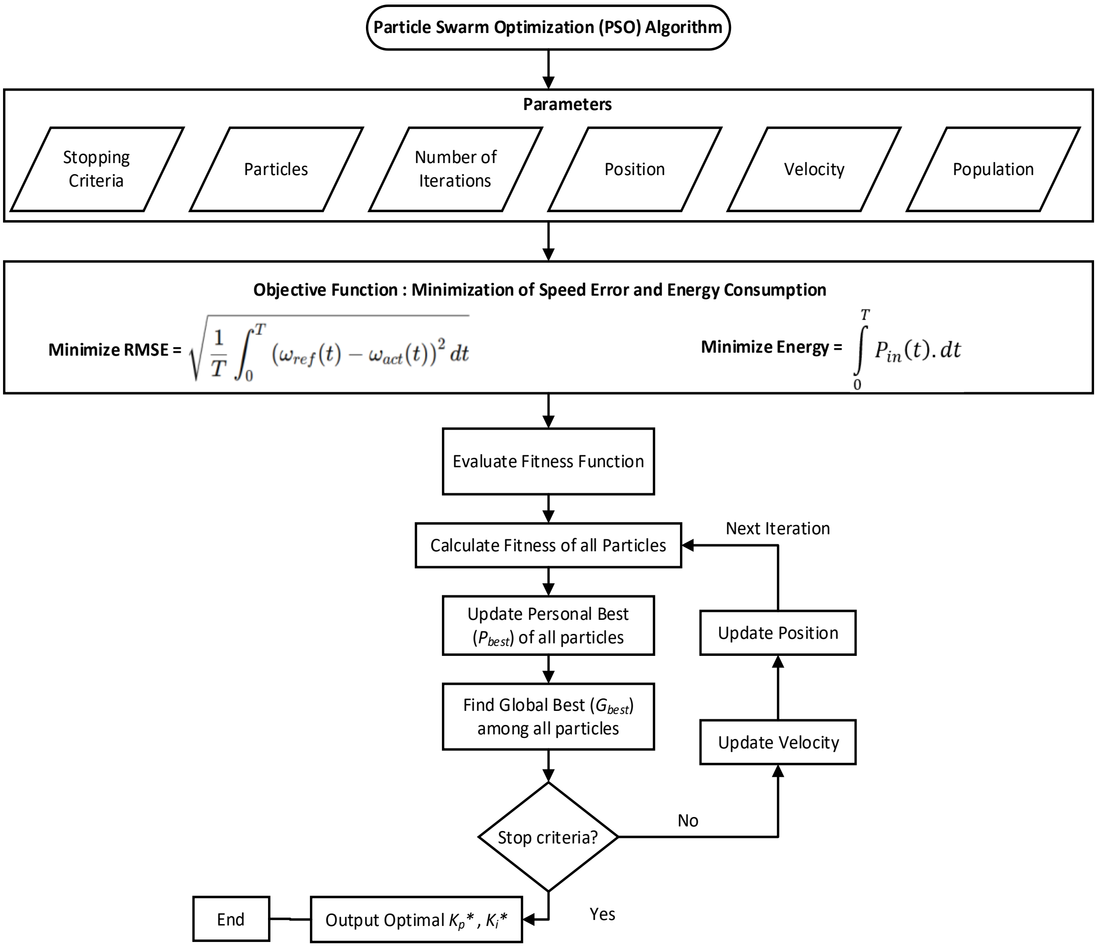

Figure.4. Flowchart of PSO-based multi-objective optimization.

Algorithm 1. PSO-Based Multi-Objective Optimization of PI Gains.

1: Initialize swarm particles $\boldsymbol{X_i}$ *and velocities* $\boldsymbol{V_i}$
2: Set personal best $\boldsymbol{P_{best\_i}} = \boldsymbol{X_i}$ *and global best* $\boldsymbol{G_{best}}$
3: for $\boldsymbol{t = 1}$ *to* $\boldsymbol{max\_iter}$ *do*
4: Update inertia weight $\boldsymbol{\omega_{PSO}}$
5: for each particle $\boldsymbol{i}$ *do*
6: Simulate BLDC model using $\boldsymbol{X_i} = [\boldsymbol{K_{p\_i}}, \boldsymbol{K_{i\_i}}]$
7: Compute $\boldsymbol{RMSE}$ *and* $\boldsymbol{energy}$ *consumption*
8: Evaluate fitness:

$$\boldsymbol{f_i} = \boldsymbol{\alpha}\left(\frac{\boldsymbol{RMSE_i}}{\boldsymbol{RMSE_{norm}}}\right) + \boldsymbol{\beta}\left(\frac{\boldsymbol{Energy_i}}{\boldsymbol{Energy_{norm}}}\right)$$

9: Update $\boldsymbol{P_{best\_i}}$ *if current fitness is better*
10: Update $\boldsymbol{G_{best}}$ *if current fitness is better*
11: end for
12: for each particle $\boldsymbol{i}$ *do*
13: Update velocity $\boldsymbol{V_i}$
14: Update position $\boldsymbol{X_i}$
15: Enforce search bounds
16: end for
17: end for
18: Return optimal gains $\boldsymbol{G_{best}} = [\boldsymbol{K_p}^*, \boldsymbol{K_i}^*]$

4.2.3. GA

In order to make an unbiased comparison between PSO and GA optimization methods, the following criteria must be observed. The search space, objective function, and simulation environment are exactly the same for both optimizations. A gene contains information regarding two parameters $[K_p, K_i]$, where the proportional parameter is sought in the range [1, 10] while the integral is sought in the range [100, 2000]. The GA with the population size, and a number of generations as mentioned in Table 8, thereby maintaining parity with the PSO implementation in terms of search scale and computational effort.

The value for crossover probability is assigned as 0.8, whereas the value for mutation probability is assigned to be 0.2, while elitism is enforced by keeping the top two chromosomes from every generation. The selection of parents is based on the tournament method, the arithmetic crossover method is employed to create the new generation, and the Gaussian mutation method is used to ensure diversity within the population. Subsequently, the gain values obtained after mutation are constrained to lie between the allowable limits, and then these values are assessed using the simulation model. The ranking process based on their performance takes place at every stage of the generation, while keeping track of the most efficient chromosome. The exploration process that GA offers is wider compared to PSO owing to the use of crossover and mutation repeatedly, although the convergence rate tends to be slower.

4.2.4. GA pseudocode and flow chart

These computational procedures associated with the implementation of the GA based tuning method have been described in detail in Algorithm 2. The same flow of operations can be shown using a flowchart as illustrated in Figure 5 below. This pseudocode explains how the population of chromosomes is initialized, how fitness computation is performed, followed by fitness ranking, elitism, parent chromosome selection, crossover, mutation, and formation of a new population in subsequent iterations of the generation cycle. The flowchart is an illustration of this process that leads to an optimal set of gains for PI controllers through evolutionary search operations.

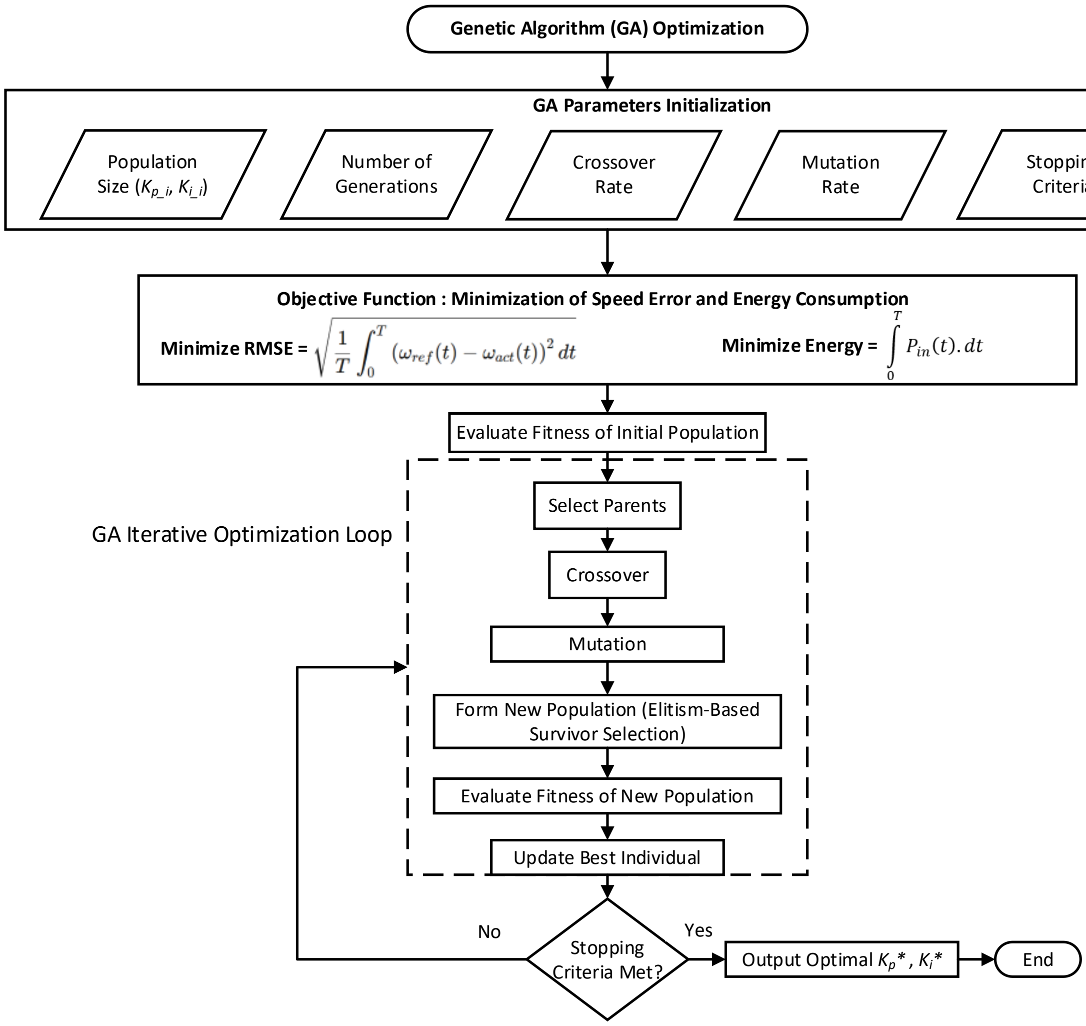

Figure. 5. Flowchart of GA-based multi-objective optimization.

Algorithm 2. GA-Based Multi-Objective Optimization of PI Gains.

Initialize a population of chromosomes $\boldsymbol{X_i} = [\boldsymbol{K_{p\_i}}, \boldsymbol{K_{i\_i}}]$
Evaluate fitness of each chromosome using $\boldsymbol{RMSE}$ *and* $\boldsymbol{energy}$ *consumption*
for $\boldsymbol{g = 1}$ *to* $\boldsymbol{max\_gen}$ *do*
    Sort population by fitness
    Preserve elite chromosomes
    Select parents using tournament selection
    Apply crossover to generate offspring
    Apply mutation to offspring
    Enforce parameter bounds
    Form the new population
    Re-evaluate fitness of all chromosomes
    Update the best chromosome if improved
end for
Return optimal gains $[\boldsymbol{K_p}^*, \boldsymbol{K_i}^*]$

4.3. MATLAB/Simulink co-simulation

The PSO and GA approaches are realized in the framework of co-simulation between MATLAB and Simulink, where the former carries out the optimization procedure while the latter serves as the environment for simulating the behavior of the dynamic plant. To test each solution, the parameters K_p and K_i are fed into the controller, and the simulation of the BLDC drive model runs within the chosen optimization period. Then, the outputs ($RMSE$ and $Energy$) are obtained from the simulation result and serve as metrics of controller performance considering the proposed multi-objective problem formulation. If the simulated response is unstable or does not converge at all, an artificially high-cost value is applied to that solution, thereby excluding infeasible regions of the search space from the optimization process. As the same co-simulation approach, objective function definition, and gain constraints are applied to PSO and GA, the difference between the two techniques lies in how they conduct the search process, not in the plant evaluation and/or control system definition aspects.

4.4. Overall workflow

Figure 6 summarizes the overall workflow of the proposed study. The procedure begins with the development of the MATLAB/Simulink-based BLDC motor drive model and the formulation of PI controller tuning as a multi-objective optimization problem involving speed-tracking error and energy consumption. The ZN method is first used to obtain baseline gains, after which PSO and GA are independently executed, each solving the same multi-objective optimization problem within identical objective functions, parameter bounds, and feasible search spaces to determine the optimal K_p and K_i values. The optimized gains obtained from each algorithm are not combined; instead, they are separately validated, together with the ZN baseline, over the 120-second drive cycle under inclination-equivalent loading conditions. Their respective performances are then compared in terms of speed tracking, energy consumption, convergence behavior, optimization landscape, and statistical robustness.

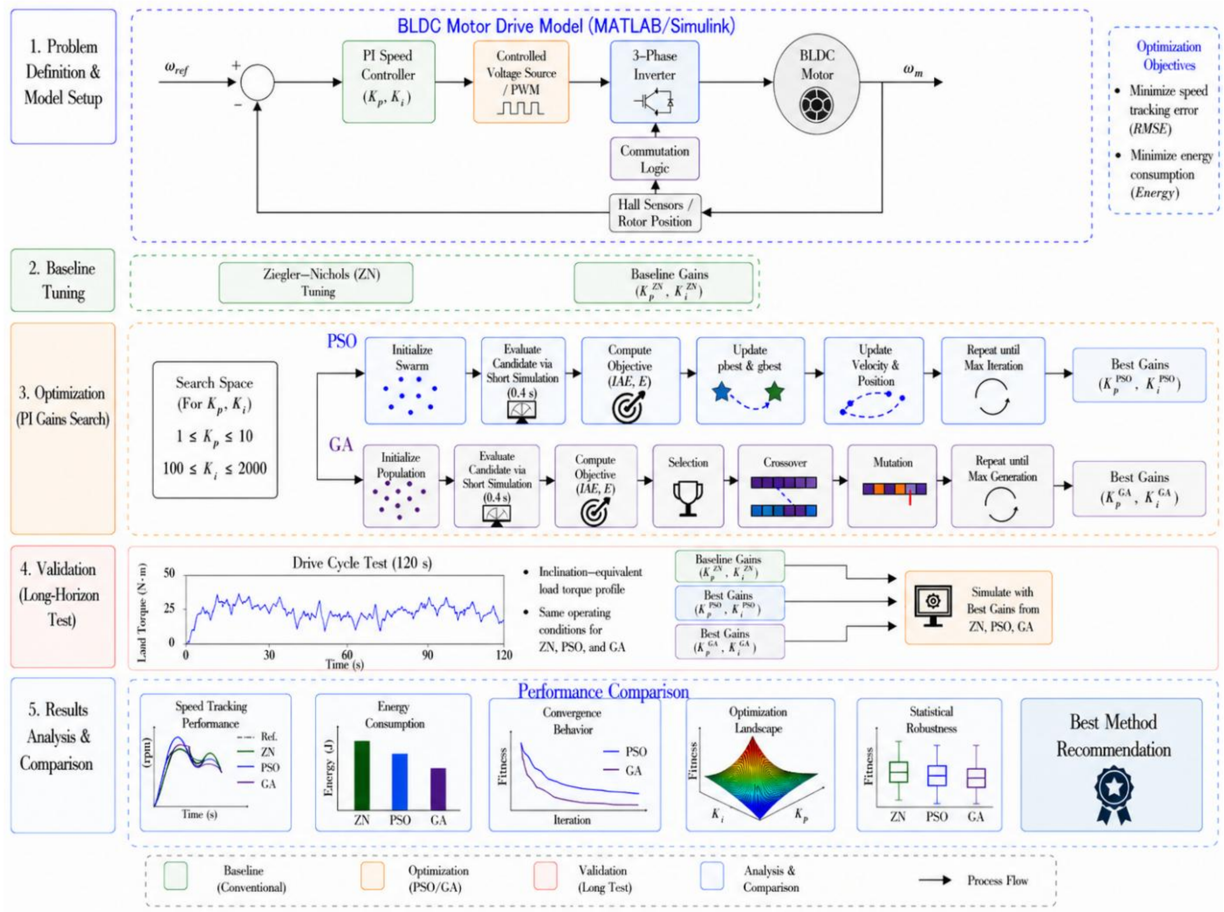

Figure 6. Overall workflow of the proposed ZN–PSO–GA comparative evaluation framework.

4.5. Two-stage evaluation strategy

To achieve a good balance between computational feasibility and applicability, the experiment follows a two-step approach. First, the gains of the controller are optimized for a short period of 0.4 s, which enables transient behavior analysis, but not too much computational cost. Second, the obtained gains from PSO and GA are verified over the 120 s driving cycle, using the load torque equivalent to an inclined road profile. The ZN-tuned control system is tested under similar conditions, ensuring fairness among all the three tuning techniques. However, it is important to clarify that the gains tuned over the short 0.4 s window may not represent the global optimum for the longer 120 s driving cycle. Accordingly, the first stage is interpreted as a computationally efficient gain-selection step, while the second stage serves to evaluate the transferability of the optimized gains under longer-horizon operating conditions. This two-stage strategy therefore provides an engineering approximation rather than a formal proof of long-horizon optimality.

5. Results and discussion

In this section, I focus on comparing the performance of the three PI parameter tuning techniques, including ZN, PSO, and GA, when used for EVs. The experiment uses a two-step procedure where the optimization algorithms are first tested using a relatively shorter time frame of 0.4 seconds for controller tuning and then validated on a more extended period of 120 seconds when used for driving. Such an approach makes it possible to investigate short-term optimization performance and long-term closed-loop system performance simultaneously. Furthermore, the gains selected on the short horizon are evaluated for transferability on the long horizon rather than being claimed as globally optimal for the full-cycle case. Therefore, the long-horizon results should be interpreted as empirical evidence of transferability within the adopted framework.

5.1. Full-cycle speed-tracking performance under no-load and varying-load conditions

The full-cycle validation results under no-load and varying-load operating conditions is shown in Figure 7. Figure 7(a) and Table 4 show that the no-load case yields highly accurate speed regulation for all three controllers, with very small RMSE, MAE, and maximum absolute error values, together with very low energy consumption. Under this nominal condition, the differences among the controllers are minor. In fact, the ZN-tuned controller provides the lowest RMSE (0.007556 kph), the lowest maximum absolute error (0.1106 kph), the lowest energy consumption (69.949 J), and the best fitness value, whereas PSO yields only a slightly lower MAE (0.0046024 kph) than ZN (0.0046921 kph). These results indicate that, in the absence of external load disturbance, even the conventional ZN tuning method is sufficient to maintain near-ideal speed regulation, and the benefit of metaheuristic tuning is not pronounced under this operating condition.

The situation changes significantly when the varying-load profile is introduced. The effect of this disturbance on speed regulation is reflected in Figure 7(b) and Table 4, where all error measures and the total energy consumption increase substantially relative to the no-load case. As illustrated in Figure 7(c), the applied load torque varies between 0 and -4 $N \cdot m$, and the corresponding mechanical power magnitude increases from relatively small levels such as 37.699 W at 10.8 kph and -1.2 $N \cdot m$ to 477.522 W at 43.2 kph and -3.8 $N \cdot m$, 586.431 W at 50.4 kph and -4 $N \cdot m$, and a maximum of 628.319 W at 54.0 kph under -4 $N \cdot m$ load. These values confirm that the most demanding portions of the drive cycle occur under simultaneous high-speed and high-torque operation. Under these conditions, the ZN controller exhibits noticeably stronger oscillatory behavior and larger deviations from the reference trajectory, whereas the PSO and GA based controllers remain considerably closer to the desired speed. This is confirmed quantitatively in Table 4, where PSO and GA reduce the RMSE from 0.57215 kph for ZN to 0.18986 and 0.19027 kph, the MAE from 0.32719 kph to 0.12257 and 0.12259 kph, and the maximum absolute error from 2.5124 kph to 1.0386 and 1.0747 kph, respectively, while lowering the energy consumption from 26090 J to 25353 J and 25415 J. Therefore, Figure 7 and Table 4 together show that although all controllers perform very well under no-load operation, the advantage of the metaheuristic optimization-based controllers becomes much more evident under dynamically varying load conditions, where improved disturbance tolerance and reduced oscillatory behavior are required.

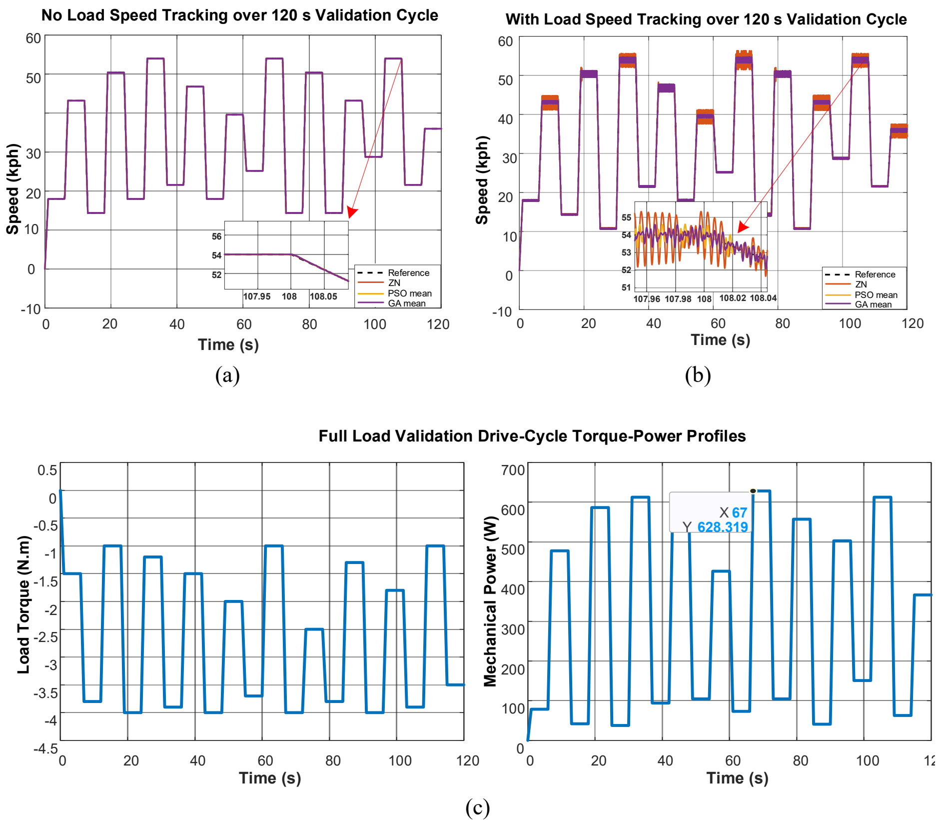

Figure 7. Full-cycle validation results under no-load and varying-load operating conditions: (a) Speed-tracking responses under no-load operation, (b) speed-tracking responses under the varying-load condition, and (c) load-torque and mechanical-power profiles used in the varying-load case.

Table 4. Comparative full-cycle validation metrics of the ZN, PSO, and GA tuned PI controllers under no-load and varying-load operating conditions.

No-load full-cycle validation metrics							
Method	K_p	K_i	$RMSE$ (kph)	MAE	Maximum Absolute Error	Energy (J)	Fitness
ZN	3.6000	1080.00	0.0075560	0.0046921	0.1106	69.949	1.0000
PSO mean	2.8163	950.00	0.0080101	0.0046024	0.1349	70.114	1.0370
GA mean	2.8309	957.78	0.0082822	0.0049519	0.1318	70.794	1.0625
Varying-load full-cycle validation metrics							
ZN	3.6	1080	0.57215	0.32719	2.5124	26090	1
PSO mean	2.8163	950	0.18986	0.12257	1.0386	25353	0.58781
GA mean	2.8309	957.78	0.19027	0.12259	1.0747	25415	0.58917

5.2. Sensitivity of search-budget parameters

Figure 8 illustrates the sensitivity of the adopted search-budget configuration for PSO and GA using the combined settings 10/40, 15/50, and 20/60, where the first number denotes swarm/population size and the second denotes iteration/generation count. As shown in Figure 8(a), for PSO, reducing the search budget from the baseline 15/50 to 10/40 increases the fitness by approximately 0.95%, whereas increasing it to 20/60 improves the fitness by only about 0.44%. A similar trend is observed for GA in Figure 8(b), where the corresponding fitness changes are approximately 0.91% and 0.46%, respectively. In contrast, the normalized computational cost varies much more strongly, decreasing to about 0.53 for the reduced setting and increasing to about 1.60 for the larger setting relative to the baseline. These results indicate that the adopted configuration of 15 individuals and 50 iterations/generations lies close to a saturated-performance region for the present two-parameter tuning problem, while larger search budgets mainly increase computational cost with only marginal gain in solution quality.

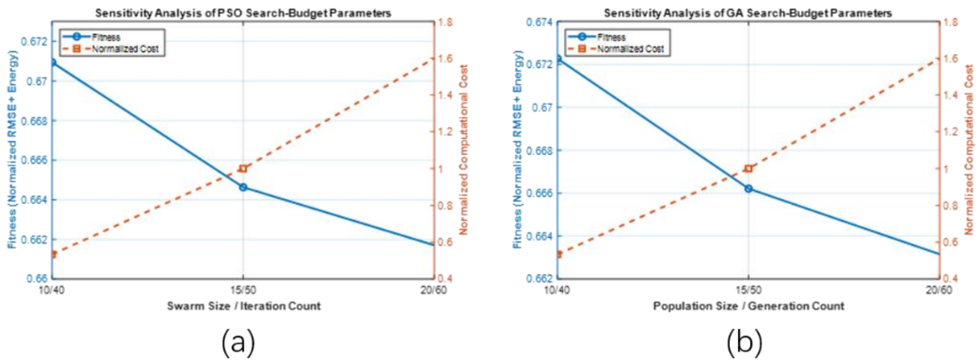

Figure 8. Sensitivity analysis of the adopted search-budget settings for the two metaheuristic optimizers: (a) PSO sensitivity to combined variation in swarm size and iteration count, and (b) GA sensitivity to combined variation in population size and generation count.

5.3. Sensitivity of fitness-function weighting factors

Figure 9 illustrates the sensitivity of the optimization outcome to representative weight combinations for the speed-tracking and energy terms. For both PSO (Figure 9(a)) and GA (Figure 9(b)), increasing the tracking emphasis from 0.6/0.4 to 0.7/0.3 slightly improves the normalized RMSE at the expense of a small increase in normalized energy, whereas shifting to 0.5/0.5 slightly improves the energy-related behavior while producing a small degradation in tracking quality. However, the variation in the overall fitness remains limited for all tested weight combinations, indicating that the adopted weighting of 0.6/0.4 provides a practically balanced trade-off between tracking performance and energy consumption. Therefore, moderate variation in the weighting factors does not materially alter the main comparative conclusion of the study.

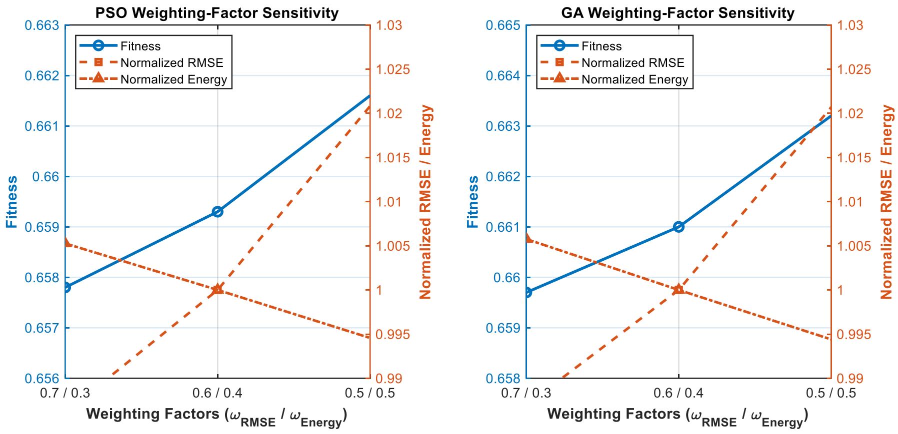

Figure 9. Sensitivity analysis of the adopted fitness-function weighting factors: **(a)** PSO response to representative weight combinations for the speed-tracking and energy terms, and **(b)** GA response to the same weight combinations.

5.4. Statistical analysis

To evaluate the consistency and performance comparison between the two techniques, PSO and GA were conducted on 10 separate trials within similar search range, objective function, and MATLAB/Simulink co-simulation framework for the 0.4 s optimization process. During each trial, the best result achieved was documented based on the criteria of fitness value, RMSE, energy cost, gain values, and time taken for the optimization process. Comparisons between the values using ZN controller and the optimized benchmark average values by PSO and GA are presented in Table 5. Summary statistics of the ten runs conducted are illustrated in Table 6 while the results from the Wilcoxon rank sum test are displayed in Table 7. From Table 5, it is evident that the two optimization-based approaches yield significant improvements in the optimization phase compared to the ZN approach. For instance, the ZN approach yields RMSE values of 0.29420 and an energy value of 61.5263 J. In comparison, PSO yields RMSE values of 0.12971 and an energy value of 60.727 J. Additionally, GA yields RMSE values of 0.13052 and an energy value of 60.729 J. The respective fitness values for ZN, PSO, and GA are 1, 0.65933, and 0.66100. Furthermore, as depicted in Table 6, PSO achieves a slightly lower mean fitness value than GA, indicating marginally better average optimization-stage performance. This pattern can be seen in RMSE as well; in which PSO results in slightly lower mean compared to GA. However, the standard deviation of PSO is considerably lower than GA in fitness and RMSE. It shows the stability of PSO. In contrast, the means of energy values obtained using both algorithms are identical. Therefore, it is clear from the above results that the major difference in the performance of PSO and GA lies in their tracking capability rather than the energy savings. Further validation of this observation can be gained through the data provided by Wilcoxon rank sum test in Table 7 below, in which I find a statistically significant difference for fitness ($p=0.0058$) and RMSE ($p=0.0257$) values, but there is no statistically significant difference in terms of energy saving ($p=0.8501$). Hence, the result of the multiple runs test demonstrates that PSO algorithm performs better than GA in terms of optimization of the fitness and RMSE values. In contrast, energy saving capabilities of PSO and GA are identical. The distribution of the repeated-run results is further illustrated in Figure 10, where the boxplots compare the fitness (Figure 10(a)), RMSE (Figure 10(b)), and energy-consumption variations (Figure 10(c)) of PSO and GA across the 10 independent optimization runs.

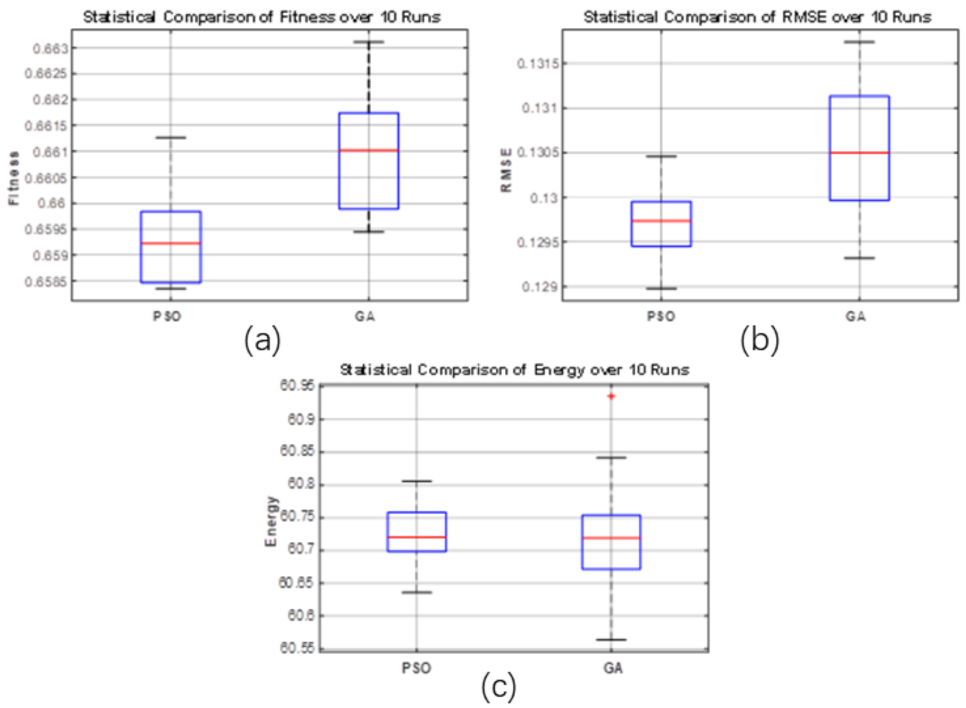

Figure 10. Boxplot comparison of (a) fitness, (b) *RMSE*, and (c) *Energy* consumption for PSO and GA over 10 independent runs.

Table 5. Optimization-stage performance comparison of ZN, PSO, and GA over the 0.4 s horizon.

Method	K_p	K_i	*RMSE*	Energy (J)	Fitness
ZN	3.6	1080	0.2942	61.5263	1
PSO	2.8163	950	0.12971	60.727	0.65933
GA	2.8309	957.78	0.13052	60.729	0.661

Table 6. Summary statistics of PSO and GA over 10 independent optimization-stage runs.

Metric	PSO Mean	PSO Std	PSO Min	PSO Max	GA Mean	GA Std	GA Min	GA Max
Fitness	0.65933	0.00092	0.65835	0.66127	0.661	0.00118	0.65944	0.66311
RMSE	0.12971	0.00045	0.12898	0.13046	0.13052	0.00081	0.12932	0.13174
Energy	60.727	0.05009	60.636	60.806	60.729	0.10116	60.563	60.936
K_p	2.8163	0.04013	2.7293	2.8682	2.8309	0.02936	2.7861	2.8871
K_i	950	50.34	897.15	1045.8	957.78	100.5	859.96	1155.9
Time(s)	9265.2	842.33	6949.4	9728.8	9256	829.25	7404.3	9804.7

Table 7. Wilcoxon rank-sum test results comparing PSO and GA over 10 optimization-stage runs.

Metric	P-Value	Rank Sum Statistic	Interpretation
Fitness	0.0057954	68	Significant
RMSE	0.025748	75	Significant
Energy	0.85011	108	Not significant

However, it is necessary to point out that the repeated run test is constrained by the fact that it consists of only 10 runs, and consequently, the statistical significance results must be taken as preliminary evidence regarding statistical reliability. The reason for such a constraint lies in the significant amount of actual wall clock computer time required to perform an optimization using the MATLAB/Simulink-based optimization framework. Consequently, the statistical analysis should be viewed as an initial but useful indication of comparative consistency rather than as a fully exhaustive statistical validation.

5.5. Convergence characteristics of PSO and GA

In this section, I investigate investigates the performance of the two optimization schemes in terms of convergence in the 0.4 s tuning period. Figure 11 shows the convergence plots of the chosen PSO and GA schemes from the analysis of repeated runs. These plots represent the change in the best fitness value achieved as the search proceeds and thus indicate how quickly each algorithm converges to a nearly optimal set of gains using the proposed multiple-objective approach.

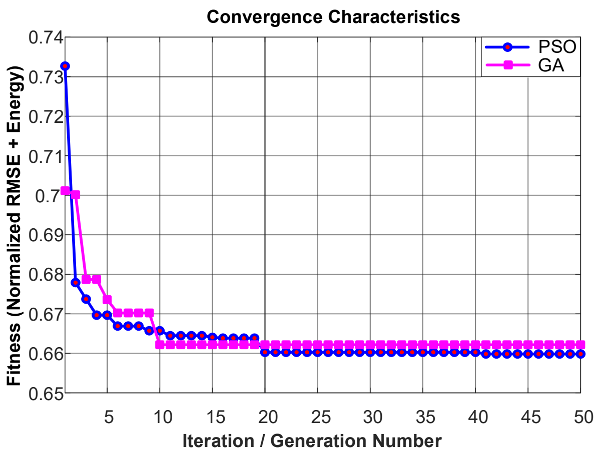

Figure 11. Convergence characteristics of PSO and GA during the 0.4 s optimization stage.

PSO demonstrates a more pronounced decline in the fitness value, meaning that the algorithm moves toward better regions at a quicker pace. Such characteristics of PSO are caused by the swarm intelligence approach, in which all particles move both to the position corresponding to their personal best fitness value and to that of the best particle in the whole population. The GA performance curve declines much slower than the other two because of the nature of the GA, which explores the solution region through natural evolution processes. Even if GA converges slower, it maintains a stable trend during all periods of optimization. This behavior supports the interpretation drawn from the statistical analysis. In relation to the efficiency of finding the solutions that cost less at a faster pace, PSO converges more rapidly to lower-cost regions, whereas GA provides a broader exploratory search pattern but reaches a slightly higher final fitness value. This is also seen from the mean fitness results where PSO has a value of 0.65933 and GA has 0.66100. This shows that the difference is in the search method and not in the computational balance.

5.6. Optimization landscape analysis

Further analysis of the tuning problem may be achieved through the three-dimensional fitness optimization map depicted in Figure 12, where the multi-objective fitness function is graphed as a function of the proportional and integral gains K_p and K_i. The depiction shows the general nature of the fitness map as well as the positioning of the ZN reference and the PSO and GA optimizations. Through the visualization of the fitness function, it becomes clear how complex the tuning problem is as well as the extent of the success of the optimization methods.

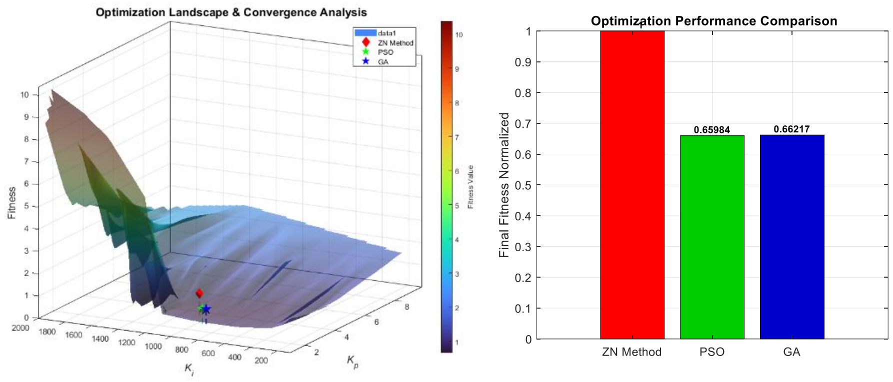

Figure 12. Three-dimensional optimization landscape of the multi-objective fitness function as a function of K_p and K_i, together with a 2D comparison of the ZN, PSO, and GA solutions.

As observed from Figure 12, the objective surface is nonlinear, exhibiting the presence of a low-cost region in contrast to the presence of a monotonically decreasing path. The controller optimized through ZN tuning is far from the lower-cost region; this can be attributed to the poor fitness and RMSE values compared to those obtained by optimizing using the other two techniques. However, the two metaheuristic methods (PSO and GA) have their solutions closer to the fitness valley, which implies that both techniques are successful in identifying gain parameters that result in better tracking and reduced energy expenditure. Therefore, the optimization landscape reinforces the main conclusion of the study. The process of tuning PI control of an EV drive system using a BLDC motor is highly nonlinear and requires better means to find optimum tuning parameters, other than heuristic methods.

5.7. Computational complexity

The computational burden of the proposed tuning framework is primarily governed by the number of simulation-based evaluations required by each optimization algorithm. Since PSO and GA are executed using the same MATLAB/Simulink model, identical gain bounds, a population or swarm size of 15, and 50 iterations or generations, the comparison remains methodologically fair with an equal nominal evaluation budget of 750 candidate solutions per run. As summarized in Table 8, the measured runtimes are also very similar, with mean execution times of 9265.2 s for PSO and 9256.0 s for GA over 10 runs. These results indicate that, within the adopted simulation framework, the two methods impose nearly identical computational cost, and their practical difference is more appropriately interpreted in terms of search behavior and final solution quality than runtime alone. Furthermore, it should be emphasized that the proposed framework is intended for offline controller tuning rather than online real-time optimization. For the practical implementation case, the optimization algorithm runs while designing or calibrating the controller, following which the gains for the PI controller become finalized and embedded within the controller of the BLDC motor drive. Thus, it would be more accurate to regard the computational cost reported here as the design-stage cost rather than the online cost. There is always room for future improvement in this aspect.

Table 8. Computational-burden comparison of PSO and GA under the adopted MATLAB/Simulink tuning framework.

Method	Population / Swarm Size	Iterations / Generations	Nominal Evaluations per Run	Mean Runtime over 10 Runs (s)	Approx. Runtime per Evaluation (s)	Interpretation
PSO	15	50	750	9265.2	12.35	Slightly better fitness, similar runtime burden
GA	15	50	750	9256	12.34	Similar runtime burden, competitive search stability

As shown in Table 8, both methods are executed under the same search budget and produce nearly identical wall-clock runtimes, indicating that the practical distinction between them lies primarily in search behavior and solution quality rather than in computational cost alone.

6. Conclusions

In this paper, I present multi-objective optimization framework for tuning the PI controller of a BLDC motor drive for EV applications, with the dual objective of improving speed-tracking performance and reducing energy consumption. A comparative evaluation was carried out among the conventional ZN method and two metaheuristic approaches, namely PSO and GA, using a two-stage strategy in which the controller gains were first optimized over a 0.4 s simulation horizon and then validated over a 120 s drive cycle under inclination-equivalent load torque variation. During the optimization stage, PSO and GA reduced the RMSE from 0.29420 for ZN to 0.12971 and 0.13052, respectively, while producing modest reductions in energy consumption from 61.5263 J to 60.727 J and 60.729 J. The repeated-run statistical analysis further showed that PSO achieved a slightly lower mean fitness value than GA and provided statistically better performance in fitness and RMSE, whereas the difference in energy consumption between the two methods was not statistically significant. Under full-cycle validation, all three controllers performed very well in the no-load case, where the differences among them remained small and the ZN controller was highly competitive. By contrast, under the varying-load case, both optimization-based controllers outperformed the ZN controller in tracking accuracy, maximum error, and cumulative energy consumption, with PSO achieving the lowest loaded-case fitness value. Overall, the results indicate that multi-objective metaheuristic tuning provides an effective approach for enhancing BLDC motor control in EV drive systems, particularly under dynamically varying load conditions, with PSO offering the most favorable speed-energy trade-off within the adopted simulation framework. Furthermore, it should be noted that the repeated-run statistical study was limited to 10 independent runs because of the substantial real wall-clock computational cost associated with repeated MATLAB/Simulink evaluations; therefore, the statistical findings should be interpreted as an initial but useful indication of comparative consistency rather than as a fully exhaustive statistical validation. In addition, since the controller gains were optimized over a short 0.4 s horizon and then validated over a 120 s drive cycle, the reported long-horizon performance should be interpreted as empirical evidence of gain transferability within the adopted framework rather than as proof of full-cycle global optimality. In future work, researchers may extend this comparative framework by incorporating developed optimization algorithms, including adaptive, hybrid, and fractional-order metaheuristic approaches, as well as advanced intelligent controller tuning techniques to provide a broader benchmarking study under identical simulation conditions. In addition, the repeated-run analysis may be expanded to larger sample sizes to strengthen statistical robustness, while the optimization process may be extended to longer time horizons to provide a more comprehensive assessment of long-cycle optimality. Experimental validation using hardware-in-the-loop (HIL) or real-time BLDC drive platforms will also be considered to further assess the practical applicability of the proposed methodology. From a deployment perspective, the proposed framework is intended as an offline gain-calibration approach rather than an online optimization scheme, with the optimized PI gains remaining fixed during real-time controller operation.

Use of Generative-AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors extend their appreciation to the Research, Development, and Innovation Authority (RDIA), Saudi Arabia, for funding this work through grant number (13385-Tabuk-2023-UT-R-3-1-SE).

Funding

This article is derived from a research grant funded by the Research, Development, and Innovation Authority (RDIA) Kingdom of Saudi Arabia with grant number (13385-Tabuk-2023-UT-R-3-1-SE).

Conflict of interest

The authors declared no conflicts of interest.

References

1. A. Q. Al-Shetwi, M. Z. Sujod, K. A. Mahafzah, A. Abuelrub, H. M. K. Al-Masri, M. A. Hannan, Climate change and global energy transformation: The role of renewable energy and electric vehicles, *Sci. Total Environ.*, **1018** (2026), 181521. https://doi.org/10.1016/j.scitotenv.2026.181521
2. Y. Bülbül, M. B. Baydar, Decarbonizing transportation: Cross-country evidence on electric vehicle sales and carbon dioxide emissions, *WEVJ*, **16** (2025), 660. https://doi.org/10.3390/wevj16120660
3. Y. Lee, M. Bencekri, J. Kwak, I. Jeong, S. Lee, Electrification pathways toward transport decarbonization with a systematic review, *Clean Technol. Environ. Policy*, **27** (2025), 7855–7877. https://doi.org/10.1007/s10098-025-03321-7
4. M. L. De Klerk, A. K. Saha, A comprehensive review of advanced traction motor control techniques suitable for electric vehicle applications, *IEEE Access*, **9** (2021), 125080–125108. https://doi.org/10.1109/ACCESS.2021.3110736
5. L. Shao, A. E. H. Karci, D. Tavernini, A. Sorniotti, M. Cheng, Design approaches and control strategies for energy-efficient electric machines for electric vehicles—A review, *IEEE Access*, **8** (2020), 116900–116913. https//doi.org/10.1109/ACCESS.2020.2993235
6. D. Mohanraj, R. Aruldavid, R. Verma, K. Sathiyasekar, A. B. Barnawi, B. Chokkalingam, et al., A review of BLDC motor: State of art, advanced control techniques, and applications, *IEEE Access*, **10** (2022), 54833–54869. https//doi.org/10.1109/ACCESS.2022.3175011
7. Y. Liu, Y. Huang, H. Zhang, Q. Huang, Stability analysis of a phase-shifted full-bridge circuit for electric vehicles based on adaptive neural fuzzy PID control, *Sci. Rep.*, **11** (2021), 20040. https://doi.org/10.1038/s41598-021-99559-4
8. R. Shenbagalakshmi, S. K. Mittal, J. Subramaniyan, V. Vengatesan, D. Manikandan, K. Ramaswamy, Adaptive speed control of BLDC motors for enhanced electric vehicle performance using fuzzy logic, *Sci. Rep.*, **15** (2025), 12579. https://doi.org/10.1038/s41598-025-90957-6
9. P. Saiteja, B. Ashok, D. Upadhyay, Evaluation of electric vehicle performance characteristics for adaptive supervisory self-learning-based SR motor energy management controller under real-time driving conditions, *Vehicles*, **6** (2024), 509–538. https://doi.org/10.3390/vehicles6010023
10. K. Zhao, D. Lou, Y. Zhang, L. Fang, Optimization and realization of the coordination control strategy for extended range electric vehicle, *Machines*, **10** (2022), 297. https://doi.org/10.3390/machines10050297
11. S. S. A. Naqvi, H. Jamil, N. Iqbal, S. Khan, D. I. Lee, Y. C. Park, et al., Multi-objective optimization of PI controller for BLDC motor speed control and energy saving in electric vehicles: A constrained swarm-based approach, *Energy Rep.*, **12** (2024), 402–417. https://doi.org/10.1016/j.egyr.2024.06.019

12. J. A. Glenn, S. Alavandar, Hybrid optimized PI controller design for grid tied PV based electric vehicle, *Intell. Autom. Soft Comput.*, **36** (2023), 1523–1545. https://doi.org/10.32604/iasc.2023.033545
13. M. A. Shamseldin, Design of auto-tuning nonlinear PID tracking speed control for electric vehicle with uncertainty consideration, *WEVJ*, **14** (2023), 78. https://doi.org/10.3390/wevj14040078
14. A. G. Gad, Particle swarm optimization algorithm and its applications: A systematic review, *Arch. Comput. Meth. Eng.*, **29** (2022), 2531–2561. https://doi.org/10.1007/s11831-021-09694-4
15. X. Li, K. Mao, F. Lin, X. Zhang, Particle swarm optimization with state-based adaptive velocity limit strategy, *Neurocomputing*, **447** (2021), 64–79. https://doi.org/10.1016/j.neucom.2021.03.077
16. W. Assawinchaichote, C. Angeli, J. Pongfai, Proportional-integral-derivative parametric autotuning by novel stable particle swarm optimization (NSPSO), *IEEE Access*, **10** (2022), 40818–40828. https://doi.org/10.1109/ACCESS.2022.3167026
17. A. Intidam, H. El Fadil, H. Housny, Z. El Idrissi, A. Lassioui, S. Nady, et al., Development and experimental implementation of optimized PI-ANFIS controller for speed control of a brushless DC motor in fuel cell electric vehicles, *Energies*, **16** (2023), 4395. https://doi.org/10.3390/en16114395
18. A. M. Alatwi, H. Albalawi, A. Wadood, I. E. Atawi, K. S. S. Alatawi, Novel GBest–Lévy adaptive differential ant bee colony optimization for optimal allocation of electric vehicle charging stations and distributed generators in smart distribution systems, *Energies*, **18** (2025), 6018. https://doi.org/10.3390/en18226018
19. A. Wadood, H. Albalawi, A. M. Alatwi, H. Park, Modified swarm-based artificial intelligence optimization for optimal coordination of directional overcurrent relays in power system, *IEEE Access*, **13** (2025), 71007–71026. https://doi.org/10.1109/ACCESS.2025.3563338
20. A. Wadood, B. S. Khan, B. M. Khan, H. Park, B. O. Kang, A fractional hybrid strategy for reliable and cost-optimal economic dispatch in wind-integrated power systems, *Fractal Fract.*, **10** (2026), 64. https://doi.org/10.3390/fractalfract10010064
21. H. Albalawi, Y. Muhammad, A. Wadood, B. S. Khan, S. T. Zainab, A. M. Alatwi, Leveraging the performance of integrated power systems with wind uncertainty using fractional computing-based hybrid method, *Fractal Fract.*, **8** (2024), 532. https://doi.org/10.3390/fractalfract8090532
22. S. Ok, Z. Xu, D. H. Lee, A sensorless speed control of high-speed BLDC motor using variable slope SMO, *IEEE Trans. Industry Appl.*, **60** (2023), 3221–3228. https://doi.org/10.1109/TIA.2023.3348081
23. B. Saha, B. Singh, Adaptive delay signal cancellation-synchronous reference frame phase-locked loop based position sensorless single stage BLDC motor drive for light duty electric vehicle, *IEEE Trans. Industry Appl.*, **60** (2023), 3229–3236. https://doi.org/10.1109/TIA.2023.3348085
24. H. Wang, Z. Li, X. Jin, Y. Huang, H. Kong, M. Yu, et al., Adaptive integral terminal sliding mode control for automobile electronic throttle via an uncertainty observer and experimental validation, *IEEE Trans. Vehic. Technol.*, **67** (2018), 8129–8143. https://doi.org/10.1109/TVT.2018.2850923

25. W. Xue, W. Bai, S. Yang, K. Song, Y. Huang, H. Xie, ADRC with adaptive extended state observer and its application to air–fuel ratio control in gasoline engines, *IEEE Trans. Indust. Elect.*, **62** (2015), 5847–5857. https://doi.org/ 10.1109/TIE.2015.2435004
26. Y. I. M. Al Mashhadany, A. K. Abbas, S. S. Algburi, Modeling and analysis of brushless DC motor system based on intelligent controllers, *Bull. Elect. Eng. Inform.*, **11** (2022), 2995–3003. https://doi.org/10.11591/eei.v11i6.4365
27. M. Mahmud, S. M. A. Motakabber, A. H. M. Z. Alam, A. N. Nordin, Control BLDC motor speed using PID controller, *Int. J. Adv. Comput. Sci. Appl.*, **11** (2020), 477–481. https://doi.org/10.14569/IJACSA.2020.0110359
28. M. M. Gujja, D. Ishak, M. N. Hamidi, M. Salem, Performance comparison and analysis of four- and six-switch inverter controls of BLDC motor employing deep learning, *J. Kejuruter.*, **36** (2024), 1409–1422. https://doi.org/10.17576/jkukm-2024-36(4)-07
29. H. Sarma, A. Bardalai, An intelligent PID controller tuning for speed control of BLDC motor using driving training-based optimization, *Int. J. Power Elect. Drive Syst. (IJPEDS)*, **14** (2023), 2474–2486. https://doi.org/ 10.11591/ijpeds.v14.i4.pp2474-2486
30. L. Bin, Z. U. Abdeen, M. Abrar, H. A. Muqeet, M. Shahzad, M. Zulfiqar, et al., A novel approach to design of a power factor correction and total harmonic distortion reduction-based BLDC motor drive, *Front. Energy Res.*, **10** (2023), 963889. https://doi.org/10.3389/fenrg.2022.963889
31. T. VenkateswaraRao, D. V. N. Ananth, Torque ripple reduction of a brushless DC motor using Y-source converter, *J. Eng. Res.*, **10** (2022), 168–203. https://doi.org/10.36909/jer.10059
32. M. A. Hannan, J. A. Ali, M. S. H. Lipu, A. Mohamed, P. J. Ker, T. M. I. Mahlia, et al., Role of optimization algorithms based fuzzy controller in achieving induction motor performance enhancement, *Nature Commun.*, **11** (2020), 3792. https://doi.org/10.1038/s41467-020-17623-5
33. M. Benbouzid, D. Diallo, M. Zeraoulia, Advanced fault-tolerant control of induction-motor drives for EV/HEV traction applications: From conventional to modern and intelligent control techniques, *IEEE Trans. Vehic. Technol.*, **56** (2007), 519–528. https://doi.org/10.1109/TVT.2006.889579
34. G. Banda, S. G. Kolli, An intelligent adaptive neural network controller for a direct torque controlled eCAR propulsion system, *WEVJ*, **12** (2021), 44. https://doi.org/10.3390/wevj12010044
35. L. Ortombina, F. Tinazzi, M. Zigliotto, Adaptive maximum torque per ampere control of synchronous reluctance motors by radial basis function networks, *JESTPE*, **7** (2018), 2531–2539. https://doi.org/ 10.1109/JESTPE.2018.2858842
36. B. M. Khan, A. Wadood, H. Park, S. Khan, H. Ali, Optimal coordination of directional overcurrent relays using an innovative fractional-order derivative war algorithm, *Fractal Fract.*, **9** (2025), 169. https://doi.org/10.3390/fractalfract9030169
37. A. Wadood, H. Albalawi, A. M. Alatwi, H. Anwar, T. Ali, Design of a novel fractional whale optimization-enhanced support vector regression (FWOA-SVR) model for accurate solar energy forecasting, *Fractal Fract.*, **9** (2025), 35. https://doi.org/10.3390/fractalfract9010035

38. A. Wadood, A. F. Yousaf, A. M. Alatwi, An enhanced multiple unmanned aerial vehicle swarm formation control using a novel fractional swarming strategy approach, *Fractal Fract.*, **8** (2024), 334. https://doi.org/10.3390/fractalfract8060334
39. B. M. Khan, A. Wadood, S. Khan, H. Ali, T. Khurshaid, A. Iqbal, et al., Analysis of war optimization algorithm in a multi-loop power system based on directional overcurrent relays, *Energies*, **17** (2024), 5542. https://doi.org/10.3390/en17225542
40. B. M. Khan, A. Wadood, H. Albalawi, S. Khan, A. M. Alatwi, O. H. Albalawi, A hybrid ensemble-based intelligent decision framework for risk-aware photovoltaic panel soiling detection and cleaning, *Electronics*, **15** (2026), 1192. https://doi.org/10.3390/electronics15061192