fractal and fractional

MDPI

Article

Robust Fractional-Order Control with Master–Slave Mechanism for Motor Speed Regulation

Davut Izci [1], Serdar Ekinci [2], Rizk M. Rizk-Allah [3] and Mohd Ashraf Ahmad [4,*]

[1] Department of Electrical and Electronic Engineering, Bursa Uludag University, 16059 Bursa, Turkey; davutizci@uludag.edu.tr
[2] Department of Computer Engineering, Bitlis Eren University, 13100 Bitlis, Turkey; sekinci@beu.edu.tr
[3] Faculty of Engineering, Menoufia University, Shebin El-Kom 32511, Egypt; rizk_masoud@sh-eng.menofia.edu.eg
[4] Faculty of Electrical and Electronics Engineering Technology, University Malaysia Pahang Al-Sultan Abdullah, Pekan 26600, Pahang, Malaysia
* Correspondence: mashraf@ump.edu.my

Abstract

Robust controller tuning is essential for the accurate regulation of nonlinear dynamic plants operating under variable conditions. This study proposes an enhanced gradient-based optimizer, termed the quadratic wavelet–enhanced gradient-based optimizer (QWS–GBO), which integrates quadratic interpolation mutation (QIM) and a wavelet mutation strategy (WMS). QIM reinforces population diversity, while WMS mitigates stagnation and strengthens local refinement through adaptive perturbations, yielding a more effective balance between global exploration and local exploitation. QWS–GBO is employed in a reference–follower control framework based on Bode's ideal response, where the follower is realized by a fractional-order proportional–integral–derivative (FOPID) controller. The FOPID parameters are optimized using QWS–GBO and evaluated in two stages. First, performance is assessed on the CEC2020 benchmark suite under a uniform protocol. Second, the approach is applied to DC motor speed regulation. On the CEC2020 functions, QWS–GBO consistently achieves lower mean objective values and faster convergence than GBO, dwarf mongoose optimization (DMO), the arithmetic optimization algorithm (AOA), and the salp swarm algorithm (SSA) with only minor computational overhead (35.90 s per trial versus 34.00 s for GBO). In the DC motor case, the QWS–GBO–tuned FOPID controller attains a rise time of 0.0216 s, settling time of 0.0350 s, zero overshoot, and peak time of 0.0509 s. Robustness tests under four operating conditions showed limited deviations (maximum 0.0058 s in rise time, 0.0113 s in settling time, 0.465% in overshoot, and 0.0131 s in peak time). Additional analyses confirmed that both QIM and WMS individually contribute measurable gains, validating their joint integration. Implementation details and parameter settings are provided to ensure reproducibility.

Keywords: fractional order controller; motor speed regulation; gradient based optimizer; quadratic interpolation mutation; wavelet mutation strategy

Check for updates

Academic Editor: Feng Liu

Received: 24 January 2026
Revised: 26 February 2026
Accepted: 10 March 2026
Published: 12 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Direct-current (DC) motors continue to constitute a core actuation element in modern engineering systems. Their persistent relevance stems from a combination of structural simplicity, economic feasibility, and dependable operation under a wide range of operating conditions [1–3]. These properties have supported their integration into applications that span robotic manipulators, lifting mechanisms, precision manufacturing platforms, and

electric transportation systems [4]. Within such contexts, rotational speed is not merely an auxiliary variable but a dominant performance determinant. Inadequate speed regulation can propagate through the entire system, impairing accuracy, degrading safety margins, and undermining operational reliability [5]. Consequently, the design of control strategies capable of delivering fast, stable, and robust speed regulation has remained a central objective in motion-control research.

Over the years, a diverse set of control paradigms has been developed to meet this objective. Classical proportional–integral–derivative (PID) controllers have retained a prominent position owing to their conceptual clarity, ease of implementation, and predictable behavior in practice [6]. To address the growing complexity of industrial systems, more advanced methodologies (including model predictive control [7], adaptive control schemes [8,9], and fuzzy-logic-based designs [10,11]) have been introduced. More recently, fractional-order PID (FOPID) controllers have attracted sustained attention [12,13]. By generalizing the integral and derivative actions to non-integer orders, FOPID controllers provide an expanded design space, enabling more flexible shaping of transient and frequency-domain characteristics. This extension has been shown to improve robustness and dynamic adaptability in comparison with integer-order structures. Regardless of the adopted control architecture, however, closed-loop performance remains fundamentally governed by the quality of parameter tuning.

In practice, controller tuning represents a non-trivial task. Analytical design rules and empirical heuristics often prove insufficient when confronted with nonlinear behavior, higher-order dynamics, or parametric uncertainty. These limitations have encouraged the adoption of metaheuristic optimization techniques, which are capable of exploring complex and high-dimensional search spaces without relying on explicit mathematical models of the plant [14]. Owing to their global search capability, minimal reliance on problem-specific assumptions, and algorithmic simplicity, such methods have become widely used in control-oriented optimization tasks [15–19]. In the domain of DC motor speed regulation, their application has consistently demonstrated that substantial improvements can be achieved when controller parameters are determined through systematic search rather than through fixed tuning rules.

1.1. Related Works

A considerable body of research has documented the use of metaheuristic algorithms for tuning PID and FOPID controllers in DC motor systems. Hekimoğlu [20] introduced the atom search optimization (ASO) algorithm and its chaotic variant for FOPID tuning, reporting marked improvements in both benchmark optimization problems and motor transient responses. Jain et al. [21] employed particle swarm optimization (PSO) to tune FOPID parameters, emphasizing the iso-damping property of the fractional-order structure, which preserves phase margin under gain variations and thereby enhances robustness. Their results demonstrated reduced overshoot and shorter settling times relative to classical PID-based designs. Agarwal et al. [22] applied the grey wolf optimizer (GWO) using the integral of time multiplied by absolute error (ITAE) criterion, showing that GWO-tuned FOPID controllers yield faster responses and favorable robustness against parameter perturbations.

Further advances were reported by Ekinci et al. [23], who proposed an opposition-based hybrid manta ray foraging optimization algorithm combined with simulated annealing. Extensive evaluations in both the time and frequency domains, supported by experimental validation, confirmed superior performance compared with several contemporary methods. An adaptive FOPID controller optimized via the artificial bee colony (ABC) algorithm was presented in [8], where improvements over genetic-algorithm-based designs were demonstrated in terms of overshoot, robustness, and control effort.

Beyond these representative studies, a broad spectrum of metaheuristic strategies has been explored for PID and FOPID tuning. These include the slime mold algorithm (SMA) [24], hybrid stochastic fractal search (HSFS) [25], manta ray foraging optimization and its hybrids [23], atom search optimization and chaotic ASO [20], GWO [22], the arithmetic optimization algorithm (AOA) and its hybrid with a Harris hawks optimizer [26], the covariance matrix adaptation evolution strategy (CMA-ES) [26], the improved marine predator algorithm (MPA) [27], the whale optimization algorithm (WOA) [28], the equilibrium optimizer (EO) [27], the Jaya algorithm [27], and the hybrid Lévy flight distribution with Nelder–Mead (LFDNM) approach [6]. Additional efforts have incorporated opposition-based Henry gas solubility optimization (HGSO) [29], gazelle simplex optimization (GSO) [30], invasive weed optimization (IWO) [31], and particle swarm optimization [28]. Collectively, these studies establish metaheuristic optimization as a mature and effective paradigm for controller tuning in DC motor applications. Recently, fractional model-free adaptive control strategies [31] have also been explored for nonlinear discrete systems, offering a data-driven alternative to model-based fractional control design. However, these approaches pursue a fundamentally different control paradigm than the optimization-based framework considered in this work.

Table 1 summarizes several notable contributions, with an emphasis on works that employ the same DC motor model adopted in this study. The table reports key time-domain indices, including rise time (t_r), settling time (t_s), percent overshoot ($\%OS$), and peak time (t_p). By juxtaposing the outcomes obtained with different optimization-based controllers, the table illustrates both the diversity of applied techniques and the variability of their performance. A salient observation emerges: Although many algorithms achieve competitive results, none exhibited uniform dominance across all metrics. This variability indicates that performance remains strongly problem-dependent and that further refinement of optimization frameworks remains warranted.

Table 1. Performances of recent and good works in the literature that used the same DC motor model in this work.

Controller	Algorithm	Reference	t_r (s)	t_s (s)	$\%OS$	t_p (s)
PID	Arithmetic optimization algorithm with Harris hawks optimizer	[26]	0.0743	0.2508	2.8376	0.1832
	Improved model of marine predator algorithm	[27]	0.0635	0.2793	7.0059	0.1516
	Hybrid Lévy flight distribution and Nelder–Mead algorithm	[6]	0.0462	0.0813	0	0.1452
	Opposition-based Henry gas solubility optimizer	[29]	0.0546	0.0946	0	0.1749
	Gazelle simplex optimizer	[30]	0.0365	0.0650	0	0.1218
	Particle swarm optimizer	[28]	0.3560	1.8028	24.2406	0.8498
FOPID	Slime mold algorithm	[24]	0.0285	0.0451	0	0.0660
	Hybrid stochastic fractal search algorithm	[25]	0.0505	0.0766	1.7674	0.1350
	Manta ray foraging optimization	[23]	0.0355	0.0562	0.1546	0.0868
	Atom search optimization	[20]	0.0376	0.0616	0	0.0987
	Grey wolf optimizer	[22]	0.0488	0.0814	0.3145	0.1528

1.2. Motivation

The continued prevalence of PID controllers in industrial practice reflects their reliability, computational efficiency, and ease of deployment [32]. For many systems, appropriately tuned PID controllers provide acceptable stability and transient behavior. Their fixed integer-order structure, however, imposes inherent limitations on design flexibility. In systems characterized by nonlinear dynamics, higher-order behavior, or pronounced parameter variability, these constraints may prevent the simultaneous attainment of rapid response, low overshoot, and robust stability.

Fractional-order PID (FOPID) controllers extend the classical framework by introducing two additional degrees of freedom through the fractional integral and derivative orders [33]. This generalization enables more nuanced shaping of dynamic behavior in both the time and frequency domains. Of particular significance is the iso-damping property associated with fractional-order structures, whereby the phase margin remains nearly invariant under gain variations, thereby enhancing robustness [34]. As a result, FOPID controllers have gained increasing prominence in advanced applications, including motor drives, energy systems, and manufacturing processes [35–40]. Their enlarged parameter space offers the potential to reconcile competing performance objectives that are difficult to satisfy within the integer-order paradigm.

The benefits of FOPID control, however, are inseparable from the challenge of parameter tuning. The expanded design space renders conventional analytical and heuristic methods inadequate, thereby necessitating systematic optimization strategies. At the same time, the "no free lunch" theorem [41] establishes that no optimization algorithm can be universally optimal across all problem classes. The effectiveness of any given optimizer is therefore intrinsically linked to the characteristics of the underlying system and performance criteria. In control design, these characteristics encompass plant order, nonlinearity, disturbance profiles, and the nature of the imposed objectives.

These considerations motivate the development of refined optimization frameworks that are capable of adapting to complex search landscapes while maintaining reliable convergence behavior. In the context of DC motor speed regulation, such frameworks must support both aggressive transient shaping and robust stability under uncertainty. The present study is driven by this requirement, seeking to construct an optimization-based control architecture that advances both stability and dynamic performance beyond the levels reported in existing literature. In the specific context of DC motor drives, the presence of coupled electrical–mechanical dynamics and parameter variability makes fixed integer-order compensation sometimes insufficient to simultaneously guarantee rapid response and robustness. Fractional-order controllers provide additional phase-shaping capability and memory effects that are particularly advantageous for such electromechanical systems. This motivates the adoption of a FOPID-based framework in the present study rather than relying solely on classical PID structures.

1.3. Contribution

This study proposes a new optimization-driven control framework for high-performance DC motor speed regulation. The central contribution is the formulation of an enhanced metaheuristic, termed the quadratic wavelet–enhanced gradient-based optimizer (QWS–GBO), developed as a refined variant of the gradient-based optimizer (GBO) [42]. While GBO has demonstrated promising performance across a range of optimization tasks, its efficiency may degrade in complex or high-dimensional landscapes, where premature convergence and stagnation can arise. These limitations are addressed through the integration of two complementary mechanisms: quadratic interpolation mutation (QIM) [43] and the wavelet mutation strategy (WMS) [44].

The QIM mechanism strengthens exploratory capability by constructing informative candidate solutions through polynomial interpolation. By exploiting fitness information from neighboring individuals, QIM generates approximated optima that direct the search toward unexplored yet promising regions of the solution space, thereby enhancing diversity and reducing the risk of early convergence. In parallel, WMS introduces adaptive, wavelet-based perturbations whose amplitude evolves over the course of iterations. Large perturbations are promoted during early stages to encourage global exploration, whereas progressively smaller adjustments are applied during later stages to refine promising so-

lutions. Through this adaptive behavior, stagnation is mitigated and local exploitation is strengthened.

The coordinated integration of QIM and WMS within the GBO framework establishes a more balanced interaction between exploration and exploitation. As a result, QWS–GBO exhibits accelerated convergence and improved solution quality in comparison with the original GBO and other contemporary optimizers.

Beyond the algorithmic development, the proposed optimizer is embedded within a master–slave control architecture for DC motor speed regulation. A reference model derived from Bode's ideal response [45,46] serves as the master system, defining the desired dynamic behavior. A fractional-order PID controller, whose parameters are optimized by QWS–GBO, operates as the slave. The tuning process is formulated as a constrained minimization problem using the integral of squared error (ISE) index [47], which penalizes large deviations and promotes smooth, overshoot-free transients. Through this arrangement, the DC motor is compelled to closely emulate the reference dynamics while preserving robustness under varying operating conditions.

The effectiveness of the proposed framework is established through a two-stage evaluation. First, QWS–GBO is validated on the CEC2020 benchmark suite and compared with the original GBO [42], dwarf mongoose optimization (DMO) [48], the arithmetic optimization algorithm (AOA) [49], and the salp swarm algorithm (SSA) [50] under uniform experimental conditions. Second, the optimizer is applied to the tuning of a FOPID controller for DC motor speed control and benchmarked against both these optimizers and a broad set of optimization-based controllers reported in the literature, including HSFS [25], MRFO [23], ASO [20], GWO [22], AOA-HHO [26], HGSO [29], MPA [27], and LFDNM [6]. The resulting analyses demonstrate consistent improvements in convergence behavior, statistical robustness, and time-domain performance, thereby establishing QWS–GBO as a competitive and effective tool for control-oriented optimization.

1.4. Organization

The remainder of the paper is organized as follows. Section 2 presents the theoretical background of gradient-based optimization and details the operators of the original GBO. Section 3 introduces the proposed enhancements, describing the integration of quadratic interpolation and wavelet mutation strategies that together form the QWS–GBO algorithm. Section 4 evaluates the performance of the proposed optimizer on the CEC2020 benchmark functions. Section 5 formulates the mathematical model of the DC motor considered in this study. Section 6 outlines the recommended design methodology, including the fractional-order PID controller and the master–slave control architecture. Section 7 provides a comprehensive comparative analysis, encompassing statistical evaluation, convergence behavior, time-domain responses, robustness assessments, and comparisons with published methods. Finally, Section 8 concludes the paper by summarizing the main findings and identifying directions for future research.

2. Gradient-Based Optimization

Gradient-based optimization (GBO) [42] is a population-oriented metaheuristic inspired by gradient concepts rooted in Newtonian mechanics. It is designed to address complex and multimodal optimization problems by maintaining a dynamic balance between global exploration and local exploitation. This balance is realized through two complementary operators: the gradient search rule (GSR) and the local escaping operator (LEO). The GSR primarily promotes wide-ranging exploration of the search space, whereas the LEO enhances local refinement and prevents premature convergence to suboptimal solutions. The optimization process begins with a population initialization stage, in which

N candidate solutions are randomly distributed within the feasible domain. This initialization ensures sufficient diversity and provides multiple starting points for the subsequent search. Each individual Z_n is generated as:

$$Z_n = Z_{min} + rand \times (Z_{max} - Z_{min}),\ n = 1, 2, \ldots, N \tag{1}$$

where $rand$ is a uniformly distributed random vector in $[0, 1]$, $Z_{\min}$ and $Z_{\max}$ denote the lower and upper bounds of the decision variables, and N represents the population size. The GSR constitutes the principal exploration mechanism of GBO. It employs a numerical-gradient-inspired formulation to update the current solution vector Z_n^t at iteration t. The update is governed by:

$$Z1_n^t = Z_n^t - randn \times B_1 \times \frac{2\Delta Z.\ Z_n^t}{(Z_{\text{worst}} - Z_{best} + \varepsilon)} + rand \times B_2 \times (Z_{best} - Z_n^t) \tag{2}$$

where Z_{best} and Z_{worst} denote the best and worst solutions obtained so far, respectively, ε is a small positive constant in $[0, 0.1]$ introduced to avoid division by zero, $randn$ is a normally distributed random number, and B_1 and B_2 are transition coefficients that regulate the balance between exploration and exploitation. These coefficients are defined as:

$$B_1 = 2.rand.a - a \tag{3}$$

$$B_2 = 2.rand.a - a \tag{4}$$

with

$$a = \left| G.sin\left(\frac{3\pi}{2} + sin\left(G.\frac{3\pi}{2}\right)\right)\right| \tag{5}$$

$$\mathrm{G} = 0.2 + \left(1 - \left(\frac{\mathrm{t}}{\mathrm{T}}\right)^3\right)^2 \tag{6}$$

$$\Delta Z = rand(1 : D).|step| \tag{7}$$

$$step = \frac{(Z_{best} - Z_{r1}^t) + \gamma}{2} \tag{8}$$

$$\gamma = 2.rand.\left(\left|\frac{Z_{r1}^t + Z_{r2}^t + Z_{r3}^t + Z_{r4}^t}{4} - Z_n^t\right|\right) \tag{9}$$

Here, T denotes the maximum number of iterations, D is the problem dimensionality, and r_1, r_2, r_3, r_4 are mutually distinct indices randomly selected from the population such that $r_i \neq n$. The vector $rand(1 : D)$ contains uniformly distributed random values. The quantity $step$ represents the adaptive step length derived from both the global best solution and randomly selected individuals. By substituting Z_{best} in place of Z_n^t in Equation (2), an alternative candidate vector is generated as:

$$Z2_n^t = Z_{\text{best}} - randn \times B_1 \times \frac{2\Delta Z.\ Z_n^t}{(yp_n^m - yq_n^m + \varepsilon)} + rand \times B_2 \times (Z_{r1}^t - Z_{r2}^t) \tag{10}$$

$$yp_n = rand.\left(\frac{[Y_n^t + Z_n^t]}{2} + rand.\Delta Z\right) \tag{11}$$

$$yq_n = rand.\left(\frac{[Y_n^t + Z_n^t]}{2} - rand.\Delta Z\right) \tag{12}$$

where Y_n^t denotes the previous position of the n-th individual. A binary flag mechanism is employed such that, when the flag equals unity, Y_n^t is replaced by $Z_{1,n}^t$; otherwise, it is replaced by $Z_{2,n}^t$. While (10) is effective for local refinement, it is insufficient for global exploration, whereas Equation (2) supports global search but weakens local exploitation. Therefore, both strategies are combined to determine the updated position as:

$$Z_n^{t+1} = r_a \times \left(r_b \times Z1_n^t + (1 - r_b) \times Z2_n^t\right) + (1 - r_a) \times Z3_n^t \tag{13}$$

$$Z3_n^m = Z_n^t - \rho_1 \times \left(Z2_n^t - Z1_n^t\right) \tag{14}$$

where r_a and r_b are uniformly distributed random numbers in $[0, 1]$, and ρ_1 is a scaling factor. The local escaping operator further enhances the search by constructing a perturbed solution Z_{LEO}^t through the combination of $Z_{1,n}^t$, $Z_{2,n}^t$, two randomly selected solutions $Z_{r_1}^t$ and $Z_{r_2}^t$, the global best Z_{best}, and a randomly generated vector Z_k^t. The LEO is activated with probability p_r:

$$\text{if rand} < p_r \begin{cases} \text{if rand} < 0.5, \\ Z_{\text{LEO}}^t = Z_n^{t+1} + f_1(v_1 Z_{\text{best}} - v_2 Z_k^t) + \frac{f_2 B_1}{2}[v_3(Z_{2,n}^t - Z_{1,n}^t) + v_2(Z_{r_1}^t - Z_{r_2}^t)], \\ Z_n^{t+1} = Z_{\text{LEO}}^t, \\ \text{else}, \\ Z_{\text{LEO}}^t = Z_{\text{best}} + f_1(v_1 Z_{\text{best}} - v_2 Z_k^t) + \frac{f_2 \rho_1}{2}[v_3(Z_{2,n}^t - Z_{1,n}^t) + v_2(Z_{r_1}^t - Z_{r_2}^t)], \\ Z_n^{t+1} = Z_{\text{LEO}}^t, \end{cases} \tag{15}$$

where $f_1 \in [-1, 1]$ is uniformly distributed, f_2 is a normally distributed random variable with zero mean and unit variance, and p_r is the switching probability. The coefficients v_1, v_2, and v_3 are defined as:

$$v_1 = H_1 \times 2 \times rand + (1 - H_1) \tag{16}$$

$$v_2 = H_1 \times rand + (1 - H_1) \tag{17}$$

$$v_3 = H_1 \times rand + (1 - H_1) \tag{18}$$

where H_1 is a binary variable determined as $H_1 = 1$ if $\vartheta_1 < 0.5$, and $H_1 = 0$ otherwise. The auxiliary vector Z_k^t is generated as:

$$Z_k^m = H_2 \times Z_p^t + (1 - H_2) \times Z_{rand} \tag{19}$$

where Z_p^t is an individual randomly selected from the population, $Z_{rand} = Z_{min} + rand(Z_{max} - Z_{min})$, and H_2 is determined by a random variable $\vartheta_2 \in [0, 1]$ in the same manner as H_1. Through the coordinated action of GSR and LEO, GBO achieves a dynamic balance between exploration and exploitation, enabling effective navigation of complex search landscapes.

3. Quadratic Interpolation and Wavelet Mutation Integrated GBO

Although GBO exhibits strong local search capability and offers easily adjustable parameters, its performance may deteriorate in highly complex or high-dimensional problems. In such cases, the search process may stagnate around local optima, thereby limiting the algorithm's ability to reach the global solution. To overcome these limitations, enhancement mechanisms can be embedded within the original framework to reinforce both exploration and exploitation. In the present study, the standard GBO is augmented through the integration of quadratic interpolation mutation (QIM) and a wavelet mutation strategy

(WMS), resulting in the proposed quadratic wavelet–enhanced GBO (QWS–GBO). Within this framework, the original GBO operators initiate and sustain the iterative search, while QIM enriches population diversity and expands the exploratory scope. Simultaneously, WMS refines the search trajectory by introducing adaptive perturbations that prevent stagnation and guide the population toward promising regions. The coordinated operation of these components strengthens the optimizer's capacity to handle complex landscapes with improved accuracy and convergence behavior. The QIM stage is incorporated into the GBO cycle to identify unexplored regions of the search space and enhance population diversity. This mechanism relies on polynomial interpolation, whereby known objective function values at selected candidate positions are employed to construct a low-order polynomial that approximates the underlying objective surface. The minimum of this polynomial yields an estimated solution that may exhibit superior fitness. For a given individual Z_i, two neighboring individuals Z_{i+1} and Z_{i+2} are selected.

A quadratic polynomial $Q(Z)$ is constructed such that:

$$\begin{cases} Q(Z_i) = a_0 + a_1 Z_i + a_2 Z_i^2 = f(Z_i) \\ Q(Z_{i+1}) = a_0 + a_1 Z_{i+1} + a_2 Z_{i+1}^2 = f(Z_{i+1}) \\ Q(Z_{i+2}) = a_0 + a_1 Z_{i+2} + a_2 Z_{i+2}^2 = f(Z_{i+2}) \end{cases} \tag{20}$$

where $f(\cdot)$ denotes the objective function. The minimum of the interpolated polynomial is obtained as in Equation (21).

$$Z^* = \frac{1}{2} \times \frac{(Z_{i+1}^2 - Z_{i+2}^2) f(Z_i) + (Z_{i+2}^2 - Z_i^2) f(Z_{i+1}) + (Z_i^2 - Z_{i+1}^2) f(Z_{i+2})}{(Z_{i+1} - Z_{i+2}) f(Z_i) + (Z_{i+2} - Z_i) f(Z_{i+1}) + (Z_i - Z_{i+1}) f(Z_{i+2})} \tag{21}$$

The newly generated candidate Z^* is incorporated using a greedy update rule:

$$\begin{cases} Z_i^{t+1} = Z_i^t & f(Z_i^t) \le f(Z^*) \\ Z_i^{t+1} = Z^* & f(Z_i^t) > f(Z^*) \end{cases}. \tag{22}$$

By applying this procedure to all individuals, the population quality is systematically enhanced, and additional diversity is injected into the search. To further mitigate stagnation and improve solution accuracy, the wavelet mutation strategy is introduced. WMS generates new candidate solutions in the vicinity of selected individuals by exploiting the dilation and translation properties of wavelet functions. The mutation amplitude is gradually reduced as iterations progress, enabling wide exploration in early stages and fine-grained refinement in later stages. For a selected solution $Z_i = (Z_{i,1}, Z_{i,2}, \ldots, Z_{i,D})$, determined by mutation probability m, the update is defined as:

$$\begin{cases} Z_{i,j}^{t+1} = Z_{i,j}^t + p_1 & r < 0.5 \\ Z_i^{t+1} = Z_{i,j}^t + p_2 & r \ge 0.5 \end{cases} \tag{23}$$

where $r \in [0,1]$ is a random number, and $p_1 = \sigma.\left(Z_{max,j} - Z_{i,j}^t\right)$, $p_2 = \sigma.\left(Z_{i,j}^t - Z_{min,j}\right)$. The wavelet coefficient σ is given by:

$$\sigma = \frac{1}{\sqrt{a}} \psi\left(\frac{\varphi}{a}\right) \tag{24}$$

$$\psi(y) = e^{-y^2/2}.cos(5y) \tag{25}$$

where $\psi(y)$ denotes the Morlet wavelet function, φ is randomly selected from $[-2.5a, 2.5a]$, and a is a stretching parameter defined as $a = s \cdot (1/s)^{(1-(t/T))}$ with s randomly drawn

from [800, 1200]. This formulation enables large perturbations during early iterations and progressively smaller adjustments as the algorithm converges.

The complete working scheme of QWS–GBO is illustrated in Figure 1. As shown, the algorithm begins with the standard GBO cycle, applies QIM to enhance exploration and population diversity, and subsequently employs WMS to refine candidate solutions and prevent stagnation. Through this layered integration, QWS–GBO achieves a robust balance between global search and local refinement, leading to faster convergence and improved solution quality across complex optimization problems. For clarity, the overall workflow of QWS–GBO can be summarized as follows. The algorithm begins with the standard GBO population update driven by the gradient search rule and local escaping operator. Subsequently, the QIM stage generates informative candidate solutions to enhance population diversity and exploration capability. Finally, the WMS applies adaptive perturbations that progressively shrink with iterations, thereby strengthening local refinement and preventing stagnation. Through this sequential interaction, the proposed method maintains a dynamic balance between global exploration and local exploitation while preserving the lightweight structure of the original GBO.

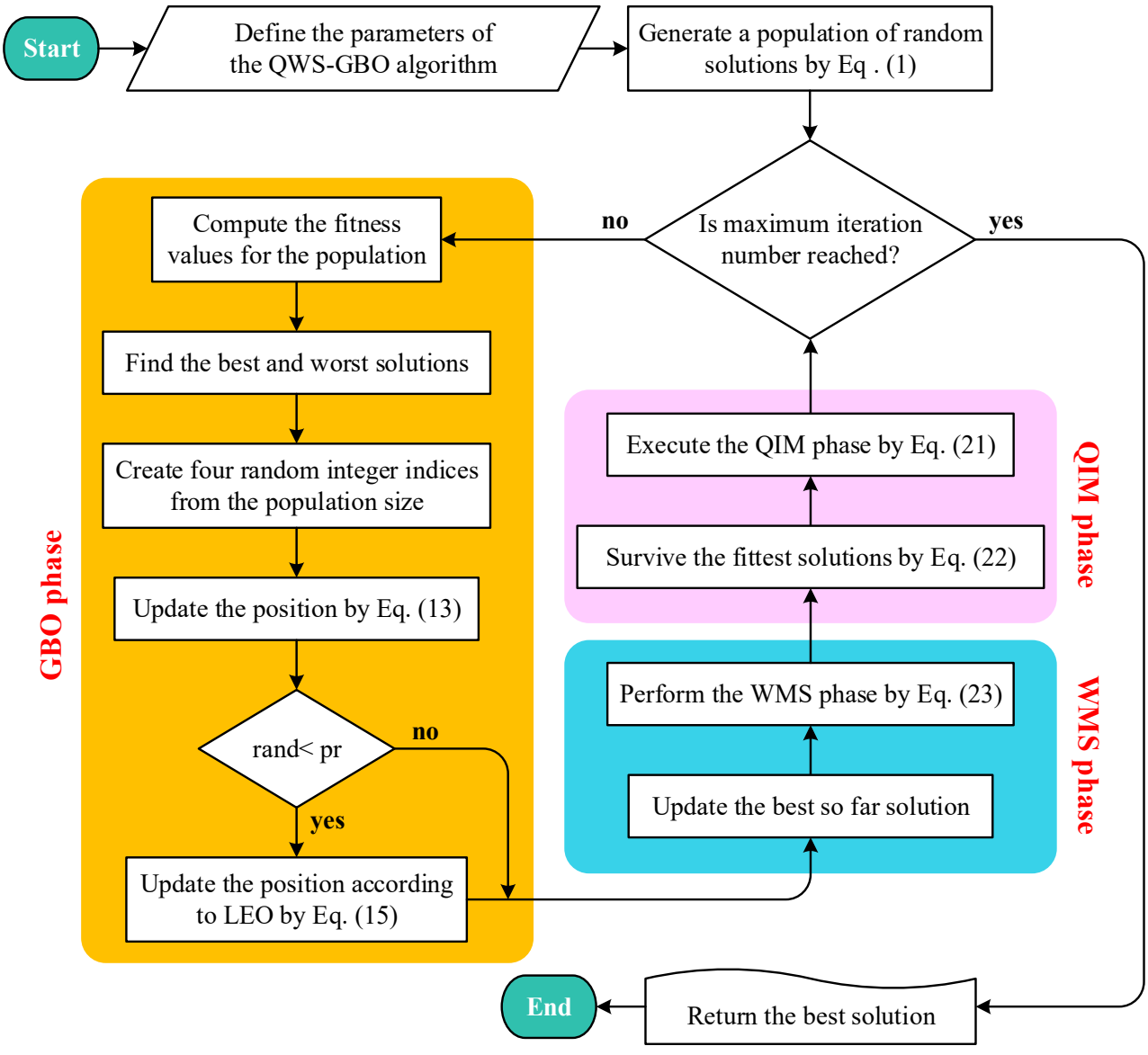


Figure 1. The working scheme of the QWS–GBO approach.

4. Empirical Evaluation of CEC2020 Benchmark Functions

This section presents an empirical assessment of the proposed QWS–GBO algorithm on the CEC2020 benchmark suite. The benchmark set, summarized in Appendix A, consists of ten challenging test functions that are widely adopted for evaluating the effectiveness and robustness of metaheuristic optimizers. Each function exhibits a distinct mathematical structure, dimensionality, and predefined global optimum. The fundamental objective for all algorithms is to locate this global minimum within the prescribed search domain [51], thereby providing a rigorous and standardized basis for comparative analysis. The performance of QWS–GBO is examined in relation to several contemporary and representative algorithms, namely the original gradient-based optimizer (GBO) [42], dwarf mongoose optimization (DMO) [48], the arithmetic optimization algorithm (AOA) [49], and the salp swarm algorithm (SSA) [50]. These methods were selected due to their recent development and established effectiveness in complex optimization tasks, thus forming a demanding baseline for evaluation. To ensure fairness and reproducibility, all of the algorithms were executed under identical experimental conditions. Each method was run 30 independent times on every benchmark function using its default parameter configuration. For each run, a population size of 50 individuals and a maximum of 1000 iterations were employed. Algorithmic performance was quantified using standard statistical indicators, including the best, worst, mean, and standard deviation of the obtained fitness values across repeated trials. These metrics provide complementary insights into convergence quality, robustness, and repeatability. The complete statistical outcomes are reported in Table 2.

Table 2. Statistical evaluation of the algorithms on CEC2020 benchmark functions.

Function	Algorithm	Best	Worst	Average	Standard Deviation
$F_{CEC2020-1}$	QWS–GBO	100.35	1439.7	449.22	420.47
	GBO	105.65	9709.4	2186.3	2242.6
	DMO	1.12×10^7	4.10×10^9	7.29×10^8	1.21×10^9
	AOA	1.81×10^9	1.29×10^{10}	6.54×10^9	2.57×10^9
	SSA	107.53	12,243	3236	2872.5
$F_{CEC2020-2}$	QWS–GBO	1111.8	1902.3	1596.7	244.58
	GBO	1130.2	2128.5	1749.3	226.13
	DMO	1453.2	2666.7	2194.6	288.25
	AOA	1359.4	2707.7	2021.1	288.74
	SSA	1138.5	2333.4	1766.4	262.51
$F_{CEC2020-3}$	QWS–GBO	713.77	742.39	732.81	7.1684
	GBO	719.44	765.58	737.6	10.643
	DMO	747.27	835.07	778.54	19.441
	AOA	769.12	834.54	797.74	15.613
	SSA	716.57	772.57	732.7	12.868
$F_{CEC2020-4}$	QWS–GBO	1900	1900	1900	0
	GBO	1900	1900	1900	0
	DMO	1903.1	27,556	3264.3	5087.8
	AOA	2571.8	2.00×10^5	52,046	57,296
	SSA	1900.4	1903.4	1901.5	0.73147
$F_{CEC2020-5}$	QWS–GBO	1774.8	2282.5	2054.4	147.56
	GBO	1845.7	2917.4	2298.5	223.54
	DMO	3674.5	5.71×10^5	69,217	1.63×10^5
	AOA	3679.4	3.10×10^5	1.55×10^5	70,051
	SSA	3015	20,479	6712.5	3874.4
$F_{CEC2020-6}$	QWS–GBO	1600	1601.1	1600.2	0.31117
	GBO	1600	1601.1	1600.2	0.33604
	DMO	1600.9	1659.5	1606.2	12.062
	AOA	1601.1	1628.8	1615.1	8.062
	SSA	1600.5	1601.6	1601	0.29051

Table 2. *Cont.*

Function	Algorithm	Best	Worst	Average	Standard Deviation
$F_{CEC2020-7}$	QWS–GBO	2101.4	2540	2303	124.09
	GBO	2118.8	2961.1	2447.9	197.69
	DMO	4674	27,880	11226	6225.9
	AOA	3334.3	10,767	6361.5	2263.6
	SSA	2438.6	21,675	7310	5184.9
$F_{CEC2020-8}$	QWS–GBO	2200	2302.7	2280.8	37.768
	GBO	2200	2305.2	2296.8	21.621
	DMO	2238.5	2544.7	2362.6	71.069
	AOA	2456.4	3388.6	2770.9	237.13
	SSA	2222.9	2305.7	2295.9	23.22
$F_{CEC2020-9}$	QWS–GBO	2500	2763.1	2717.7	87.119
	GBO	2500	2781.4	2727.3	77.953
	DMO	2521.9	2830.3	2733.4	97.457
	AOA	2679.8	2975.7	2823.7	66.405
	SSA	2722.3	2784.5	2745.1	10.568
$F_{CEC2020-10}$	QWS–GBO	2600.1	2945.8	2911.8	63.154
	GBO	2897.7	3024.4	2929.6	29.082
	DMO	2901	3055.1	2955.5	38.61
	AOA	2908.4	3433.7	3121.4	113.74
	SSA	2897.8	2951.3	2924.2	24.521

A detailed examination of Table 2 reveals clear and consistent performance advantages of the proposed QWS–GBO across the benchmark set. For $F_{CEC2020-1}$, QWS–GBO achieves the lowest average fitness value and the smallest standard deviation among all algorithms, indicating both high solution accuracy and strong repeatability. In contrast, DMO and AOA exhibit substantially larger average values and dispersions, reflecting increased sensitivity to initialization and reduced stability on this multimodal landscape. A similar pattern is observed for $F_{CEC2020-2}$, where QWS–GBO again yields the smallest mean value together with the lowest dispersion. Although SSA and GBO attain competitive best values in isolated runs, their larger standard deviations reveal less consistent convergence behavior across repeated trials. For $F_{CEC2020-3}$, QWS–GBO produces the lowest average error with a comparatively small standard deviation, outperforming all reference algorithms. These results indicate that the integration of QIM and WMS enhances the optimizer's capability to navigate narrow basins of attraction while maintaining convergence stability. In the case of $F_{CEC2020-4}$, QWS–GBO consistently converges to the known optimum value with zero standard deviation, demonstrating deterministic behavior under the selected configuration. None of the compared algorithms achieves this level of reliability, as evidenced by the non-zero variances observed for GBO, DMO, AOA, and SSA. For $F_{CEC2020-5}$, the superiority of QWS–GBO is reflected by both its reduced average fitness value and its markedly lower variance relative to DMO and AOA, which exhibit very large dispersions due to frequent divergence toward inferior local optima. The compact distribution obtained by QWS–GBO suggests improved robustness against the ruggedness of the search space. With respect to $F_{CEC2020-6}$, all algorithms approach similar average values, indicating a comparatively smoother optimization landscape. Nevertheless, QWS–GBO maintains the smallest standard deviation, confirming enhanced numerical stability and reduced stochastic variability. On the more challenging $F_{CEC2020-7}$, the advantage of QWS–GBO becomes more pronounced. The algorithm attains the lowest average fitness value together with the smallest standard deviation, whereas DMO, AOA, and SSA display substantially larger deviations, revealing difficulty in consistently escaping suboptimal regions. For $F_{CEC2020-8}$ and $F_{CEC2020-9}$, QWS–GBO again demonstrates superior convergence accuracy and robustness. Although the best values obtained by some competing methods appear comparable,

the mean and dispersion metrics consistently favor QWS–GBO, indicating that high-quality solutions are reached more reliably rather than sporadically. Finally, for $F_{CEC2020-10}$, QWS–GBO yields the smallest average fitness value with moderate variance, outperforming GBO, DMO, AOA, and SSA in both accuracy and stability. The reduced dispersion relative to AOA and DMO further confirms that the proposed modifications effectively suppress performance degradation in high-dimensional and irregular landscapes. Overall, the statistical evidence in Table 2 demonstrates that QWS–GBO consistently provides lower average objective values, smaller standard deviations, and competitive or superior best solutions across all benchmark functions. These outcomes confirm that the combined action of quadratic interpolation mutation and wavelet-based mutation substantially enhances the search dynamics of the original GBO. The improvements are particularly evident for functions characterized by high multimodality and complex variable interactions, which are representative of real-world engineering optimization problems. Consequently, the proposed QWS–GBO exhibits the robustness, stability, and efficiency required for subsequent application to controller tuning in dynamic systems.

5. Mathematical Model of DC Motor

Two commonly used DC motor configurations in industrial practice are the externally excited and self-excited types [52]. In this study, an externally excited DC motor is considered, and the shaft speed is regulated by manipulating the armature voltage. As indicated in Figure 2, the motor is represented by an armature electrical circuit coupled to the rotor's mechanical dynamics, where the field circuit is assumed to establish a constant magnetic flux under external excitation [53,54]. This representation enables the electromechanical interaction to be expressed through standard circuit and rotational-motion relations.

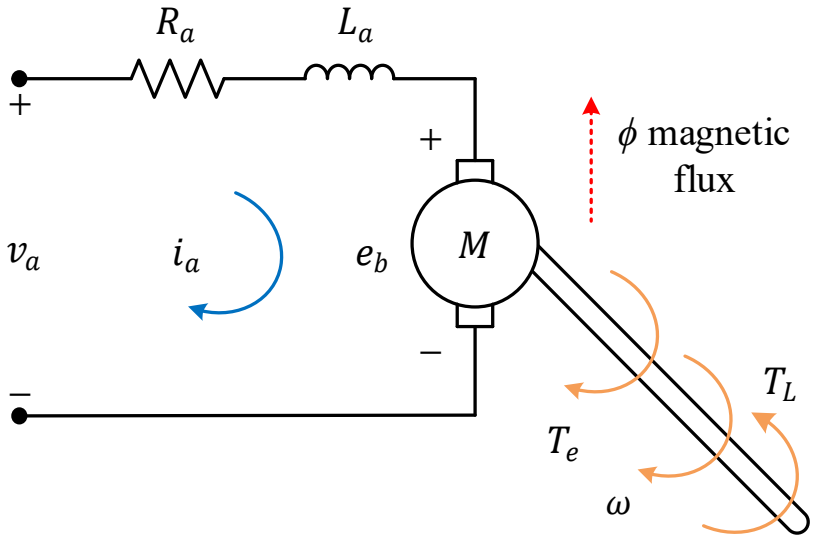


Figure 2. Equivalent circuit of DC motor.

To obtain a tractable analytical model suitable for controller design and a fair comparison with prior studies, several standard simplifying assumptions are adopted herein. The magnetic field flux is assumed to be constant due to external excitation, thereby neglecting magnetic saturation effects. Armature reaction is ignored, and the back electromotive force is considered linearly proportional to the rotor speed. Mechanical friction is modeled as viscous and linear, while Coulomb friction and dead-zone nonlinearities are neglected. In addition, the load torque is assumed to be constant during nominal operation, and parameter variations are addressed separately in the robustness study. These assumptions

are commonly employed in DC motor control literature and provide an adequate balance between model fidelity and analytical tractability.

In conventional armature-voltage control, an applied armature voltage v_a drives an armature current i_a, and an electromechanical conversion occurs through the developed electromagnetic torque and the back electromotive force (back-EMF). By applying Kirchhoff's voltage law to the armature circuit shown in Figure 2, the armature dynamics are written as:

$$v_a = i_a R_a + L_a \frac{di_a}{dt} + e_b \tag{26}$$

where R_a and L_a denote the armature resistance and armature inductance, respectively, and e_b denotes the back-EMF. Under the externally excited condition, the magnetic flux is treated as constant; hence, the back-EMF is proportional to the angular speed. This relationship is expressed as:

$$e_b = K_b \frac{d\theta}{dt} = K_b \omega \tag{27}$$

where θ is the rotor angular position, ω is the rotor angular velocity, and K_b is the back-EMF constant. On the mechanical side, the developed electromagnetic torque T_e is balanced by the load torque T_L and the torques associated with inertia and viscous friction. Using Newton's second law for rotational motion, the torque balance is obtained as:

$$T_e - T_L = J \frac{d\omega}{dt} + B\omega = K_m i_a \tag{28}$$

where J is the rotor moment of inertia, B is the viscous friction coefficient, and K_m is the torque constant. In Equation (28), T_e denotes the electromagnetic torque produced by the armature current and T_L denotes the external load torque applied to the shaft. By taking the Laplace transform of Equations (26)–(28) under zero initial conditions, the following relations are obtained.

$$V_a(s) = (L_a s + R_a) I_a(s) + E_b(s) \tag{29}$$

$$E_b(s) = K_b \omega(s) \tag{30}$$

$$T_e(s) - T_L(s) = (Js + B)\omega(s) = K_m I_a(s) \tag{31}$$

Substituting Equation (30) into Equation (29) and rearranging them yields the armature current in terms of the applied voltage and motor speed.

$$I_a(s) = \frac{V_a(s) - K_b \omega(s)}{(L_a s + R_a)} \tag{32}$$

From Equation (31), the speed can be written as a function of the torque difference, or equivalently as a function of the armature current.

$$\omega(s) = \frac{T_e(s) - T_L(s)}{(Js + B)} = \frac{K_m}{(Js + B)} I_a(s) \tag{33}$$

By combining Equations (32) and (33), the transfer function between the input armature voltage and the output angular speed is obtained. In the absence of load torque, i.e., $T_L(s) = 0$, the motor transfer function is expressed as follows.

$$G_{motor}(s) = \frac{K_m}{(L_a s + R_a)(Js + B) + K_b K_m} \tag{34}$$

For the simulations conducted in this work, the motor parameters listed in Appendix B were adopted, consistent with the values used in prior studies [26,27,29]. Substituting these parameters into Equation (34) results in the numerical plant model.

$$G_{plant}(s) = \frac{0.015}{0.00108s^2 + 0.0061s + 0.00163} \tag{35}$$

This transfer function is employed as the controlled plant in the subsequent controller design and comparative evaluations.

6. Recommended Novel Design Method for Improving the Performance of a DC Motor

This section presents the recommended design method for improving DC motor speed regulation performance. A reference–follower architecture is adopted, in which the follower is a fractional-order PID (FOPID) controller and the reference model conforms to Bode's ideal response. The FOPID parameters are tuned via the proposed QWS–GBO optimizer by minimizing an integral of squared error objective under practical bounds. The forthcoming subsections recall the FOPID fundamentals (Section 6.1), summarize the reference–follower rationale (Section 6.2), and formalize the optimization-based design used throughout the simulations (Section 6.3).

6.1. Fundamental of FOPID Controller

When suitably tuned, a conventional PID controller is capable of delivering satisfactory closed-loop behavior over a broad operating range. Rapid transient recovery, acceptable robustness against parameter variations and modeling uncertainties, and effective attenuation of measurement noise can be achieved while preserving a simple structure and low computational cost [55]. Nevertheless, the fixed integer-order nature of the PID formulation inherently constrains the attainable dynamic flexibility, particularly in systems characterized by higher-order dynamics, nonlinear behavior, or significant uncertainty. To overcome these limitations, the fractional-order PID (FOPID) controller has been introduced as a generalized extension of the classical PID structure [56]. By allowing integral and derivative actions to assume non-integer orders, the FOPID controller provides an expanded design space and enables more refined shaping of both transient and frequency-domain characteristics. Several advantages over the conventional PID formulation have been reported [24]:

1. Five adjustable parameters are incorporated, providing two additional degrees of freedom in control design.
2. Iso-damping behavior can be achieved over a wider frequency range, thereby improving stability margins.
3. Enhanced robustness is obtained, yielding increased resilience to parameter variations and modeling errors.
4. Superior performance is attained in complex systems, including higher-order, nonlinear, non-minimum phase, and time-delay plants.

The transfer function of a FOPID controller is expressed as:

$$C_{FOPID}(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \tag{36}$$

where K_p, K_i, and K_d denote the proportional, integral, and derivative gains, respectively, while λ and μ represent the fractional orders of integration and differentiation. Through appropriate selection of λ and μ, classical control structures emerge as special cases of Equation (36). In particular, the P, PI, PD, and PID controllers are recovered for integer-

valued orders, illustrating that the conventional designs correspond to isolated points within the broader FOPID parameter space.

The fractional orders λ and μ play a decisive role in shaping the closed-loop dynamics. Unlike the integer-order PID, where phase and gain characteristics are tightly coupled, the fractional formulation enables independent adjustment of slope and phase behavior over a broader frequency range. In particular, varying μ modifies the high-frequency phase contribution and damping characteristics, while λ governs the low-frequency memory effect and steady-state error dynamics. This additional flexibility enlarges the feasible stability region and facilitates iso-damping behavior, which is difficult to achieve with integer-order controllers. Consequently, the FOPID structure is not merely a higher-parameter template but a fundamentally richer control architecture.

In practical simulation, the fractional powers of the Laplace operator in Equation (36) cannot be implemented directly in the time domain. In this study, the fractional differ-integrators were realized using the well-known Oustaloup recursive approximation. Specifically, the operator s^{β} was approximated by a band-limited rational transfer function of order $2N+1$ over the frequency range $[10^{-3}, 10^{3}]$ rad/s with $N = 5$. This approximation provides an accurate magnitude and phase match within the selected band while maintaining reasonable computational complexity. The same realization was applied consistently across all algorithms to ensure fair comparison.

The conceptual relationship between the classical and fractional-order controllers is illustrated in Figure 3a, where the traditional PID design is represented as a single point and the FOPID formulation spans an entire plane. The corresponding block diagram of the FOPID controller is shown in Figure 3b, clarifying its structural composition. Based on this configuration, the closed-loop transfer function of a plant controlled by a FOPID controller is written as follows.

$$TF(s) = \frac{C_{FOPID}(s) \times G_{plant}(s)}{1 + C_{FOPID}(s) \times G_{plant}(s)} \tag{37}$$

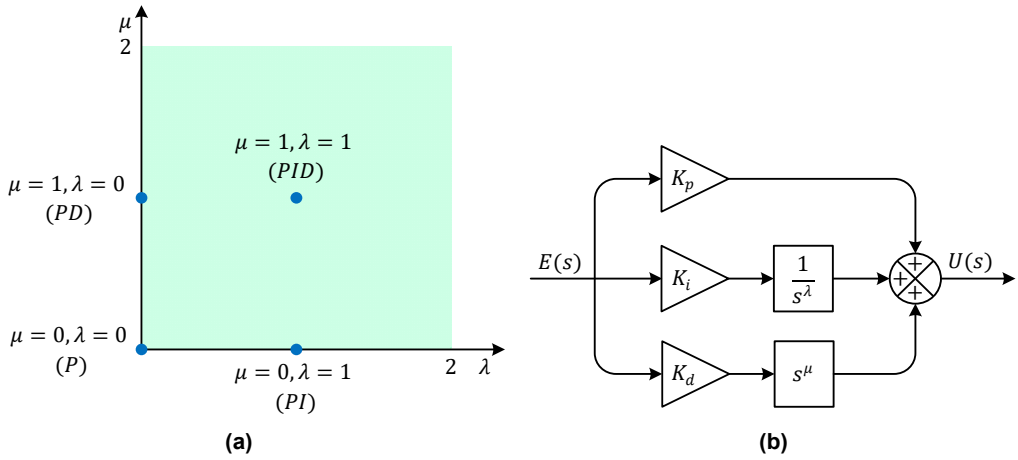


Figure 3. The plane (**a**) and block diagram (**b**) of FOPID controller.

This formulation provides the analytical foundation for the subsequent reference–follower design and optimization-based tuning procedure adopted in this work.

6.2. Master–Slave Approach

A master–slave control paradigm [57] is adopted in this study to impose a desired dynamic behavior on the DC motor while preserving robustness against gain variations and modeling uncertainties. Within this framework, an ideal reference model is employed as the master system [58] and the controlled plant equipped with a FOPID controller operates

as the slave. The role of the master is to define a target closed-loop behavior, whereas the slave is driven to reproduce this behavior as closely as possible. The task of the optimizer is therefore to adjust the FOPID parameters such that the motor response follows the reference dynamics prescribed by the master. The master system is constructed using Bode's ideal open-loop model, expressed as [43]:

$$L(s) = \left(\frac{\omega_c}{s}\right)^{\alpha} \tag{38}$$

where ω_c denotes the gain crossover frequency of $L(s)$, and α is a real-valued parameter satisfying $0 < \alpha < 2$. The parameter α governs both the slope of the magnitude curve and the phase characteristic on the Bode diagram. Specifically, the magnitude exhibits a constant slope of -20αdB/decade, while the phase assumes a horizontal line at $-\alpha\pi/2$radians. As a consequence, the resulting system possesses an iso-damping property: variations in loop gain shift only the crossover frequency ω_c, while the phase margin remains invariant at $\pi(1-\alpha/2)$radians. This property ensures that transient characteristics are preserved under gain perturbations, thereby imparting inherent robustness to the reference model. Assuming unit feedback, the corresponding ideal closed-loop transfer function of the reference model is written as follows.

$$TF_{IRM}(s) = \frac{L(s)}{1+L(s)} \tag{39}$$

In the present application, the parameters are selected as $\omega_c = 100$ and $\alpha = 1$, yielding a reference behavior that combines rapid dynamics with a constant phase margin. This ideal response serves as the master trajectory. The DC motor equipped with a FOPID controller constitutes the slave whose parameters are optimized by the proposed QWS–GBO algorithm. Through this arrangement, the closed-loop motor dynamics are compelled to emulate the ideal reference response, ensuring fast, well-damped transients and robustness against gain and parameter variations.

The use of a fractional-order controller is particularly compatible with the Bode ideal reference model. Because the fractional orders allow continuous phase adjustment, the FOPID structure can more effectively realize the constant phase-margin characteristic imposed by the iso-damping reference. This additional phase-shaping freedom is one of the key reasons for adopting the fractional-order formulation in the present master–slave framework.

6.3. Formulation of Master–Slave Model and QWS–GBO-Based FOPID Control System Design

To achieve effective and implementable speed regulation of the DC motor, the control design task is formulated as a constrained optimization problem. This formulation enables the direct use of metaheuristic search mechanisms for systematic parameter tuning. Within this framework, the decision vector is defined as $\mathrm{X} = [x_1, x_2, x_3, x_4, x_5] = [K_p, K_i, K_d, \lambda, \mu]$, where the elements correspond to the proportional, integral, and derivative gains and to the fractional integral and derivative orders of the FOPID controller. The optimization objective is established through the integral of squared error (ISE) criterion [47], expressed as:

$$ISE = \int_0^{\infty} [e(t)]^2 dt \tag{40}$$

where $e(t) = \omega^*_{\mathrm{ref}}(t) - \omega(t)$ denotes the tracking error between the ideal reference speed and the actual motor speed. The minimization of Equation (40) is carried out subject to practical bounds on the controller parameters, defined as $0 < K_p, K_i, K_d < 20$, $0.5 < \lambda, \mu < 1.5$.

The ISE index was selected because its quadratic structure imposes a strong penalty on large deviations from the reference trajectory. As a result, excessive overshoot and abrupt transients are discouraged, which is consistent with the reference–follower objective of closely emulating the ideal model response. It is acknowledged that ISE does not explicitly weight the temporal persistence of the error, in contrast to indices such as ITAE. However, in the present study, duration-related characteristics are explicitly evaluated through the reported time-domain metrics, including rise time, settling time, peak time, and the robustness scenarios. These measures provide direct insight into the temporal quality of the response and therefore complement the magnitude-sensitive nature of the ISE objective. Moreover, the use of a single-objective ISE framework ensures a fair and uniform comparison among all competing optimizers under an equal-budget protocol. Nevertheless, ITAE-based or multi-objective formulations that explicitly incorporate time-weighted errors constitute an important and promising direction for future research.

Within the proposed architecture, the ideal reference model acts as the master, defining the desired closed-loop dynamics, while the DC motor equipped with a FOPID controller forms the slave. The QWS–GBO algorithm iteratively updates the decision vector X in order to minimize the ISE subject to the imposed bounds. Through this process, the controller parameters are driven toward values that enforce close adherence of the motor response to the reference model. The complete configuration of the master–slave system, together with the QWS–GBO-based tuning loop, is illustrated in Figure 4.

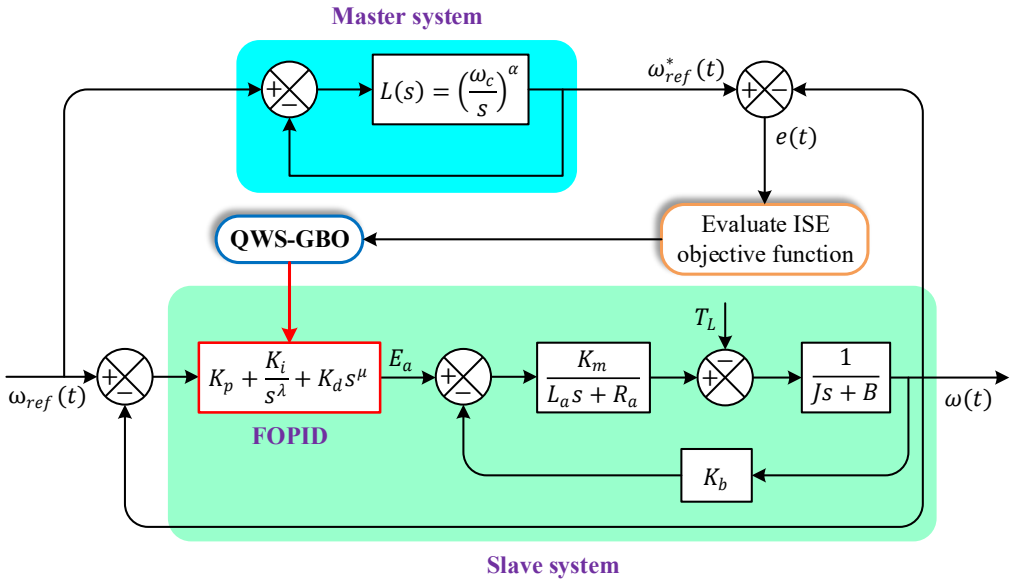


Figure 4. Application of proposed master–slave approach driven and QWS–GBO-tuned FOPID-controlled DC motor system.

7. Comparative Simulation Results and Discussion

This section reports the comparative simulation study for the DC motor case under a uniform protocol. The analysis considers QWS–GBO alongside representative baselines (GBO, DMO, AOA, SSA) under equal computational budgets. Results are organized as follows: (i) statistical summaries over multiple runs, (ii) convergence profiles of the ISE objective, (iii) nominal time response metrics, (iv) robustness across operating scenarios, and (v) comparisons with published controllers. Units and performance indices are stated explicitly and figures and tables are referenced where they substantiate the discussion. All simulations in this study were conducted using MATLAB Simulink R2025a on a machine equipped with a 12th Gen Intel® Core™ i7-1260P processor running at 2.10 GHz and 16.0 GB RAM.

7.1. Statistical Performance

For the DC motor system, the evaluation of the proposed QWS–GBO begins with a statistical assessment of its performance in minimizing the ISE cost function. The minimization process involves a population size of 30, a total of 50 iterations, and 25 runs for each algorithm. Figure 5 presents the obtained values of the cost function across all runs for QWS–GBO, GBO, DMO, AOA, and SSA. The scatter plots in this figure reveal that QWS–GBO consistently yields lower values in each run compared to its competitors. Furthermore, the boxplot analysis depicted in Figure 6 reinforces the efficacy of QWS–GBO, showcasing its consistent ability for cost function minimization within a narrow band, signifying its robust performance characteristics.

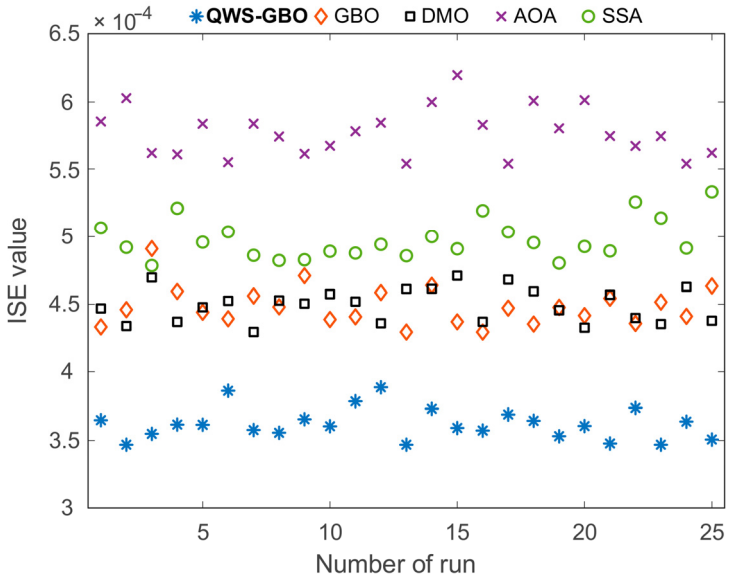


Figure 5. The ISE cost function values obtained from all runs.

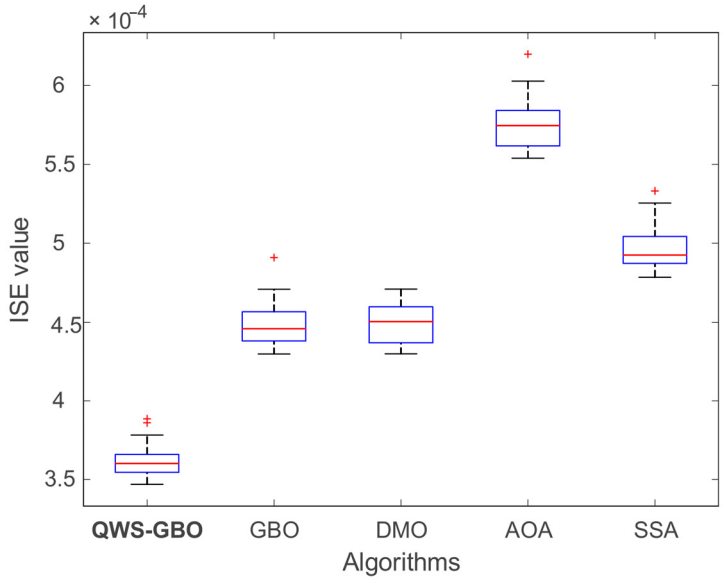


Figure 6. Comparative boxplot analysis.

Table 3 presents the statistical performance of the algorithms in terms of the ISE cost function, highlighting the effectiveness of each algorithm. The performance metrics include the best, worst, average, and standard deviation values of the cost function across multiple runs. Notably, the QWS–GBO algorithm demonstrates superior performance with

the lowest best and average values and a relatively low standard deviation, signifying consistently efficient optimization results. Comparatively, GBO, DMO, AOA, and SSA exhibit slightly higher cost function values.

Table 3. Statistical performance of algorithms.

Algorithm	Best	Worst	Average	Standard Deviation
QWS–GBO	3.4690×10^{-4}	3.8870×10^{-4}	3.6190×10^{-4}	1.1451×10^{-5}
GBO	4.2966×10^{-4}	4.9102×10^{-4}	4.4829×10^{-4}	1.4215×10^{-5}
DMO	4.2974×10^{-4}	4.7112×10^{-4}	4.4952×10^{-4}	1.2518×10^{-5}
AOA	5.5384×10^{-4}	6.1980×10^{-4}	5.7674×10^{-4}	1.7927×10^{-5}
SSA	4.7856×10^{-4}	5.3336×10^{-4}	4.9767×10^{-4}	1.4869×10^{-5}

Table 4 provides insights into the average time complexity per run for each algorithm. Here, the QWS–GBO exhibits a time complexity of 35.8972 s, which is slightly higher than the GBO's 34.0048 s. However, this marginal increase in time complexity for QWS–GBO is negligible and can be attributed to the integration of QIM and WMS strategies into the GBO. Despite this minimal increase, QWS–GBO still outperforms DMO, AOA, and SSA in terms of time complexity, showcasing its efficiency in achieving high-quality solutions.

Table 4. Average time complexity per run.

QWS–GBO	GBO	DMO	AOA	SSA
35.8972 s	34.0048 s	44.5267 s	40.5424 s	38.2567 s

7.2. Convergence Evolutions of Objective Function

Figure 7 shows the convergence profiles of the algorithms as they minimize the ISE cost function for the DC motor system. A visual analysis of the graph clearly demonstrates that QWS–GBO outperforms the other algorithms in terms of convergence behavior. It consistently achieves the lowest ISE cost function value, highlighting its exceptional ability to rapidly converge to the optimal solution. This performance underscores the efficiency and effectiveness of QWS–GBO, making it a highly promising choice for optimizing the ISE cost function in the design of the FOPID controller for the DC motor system.

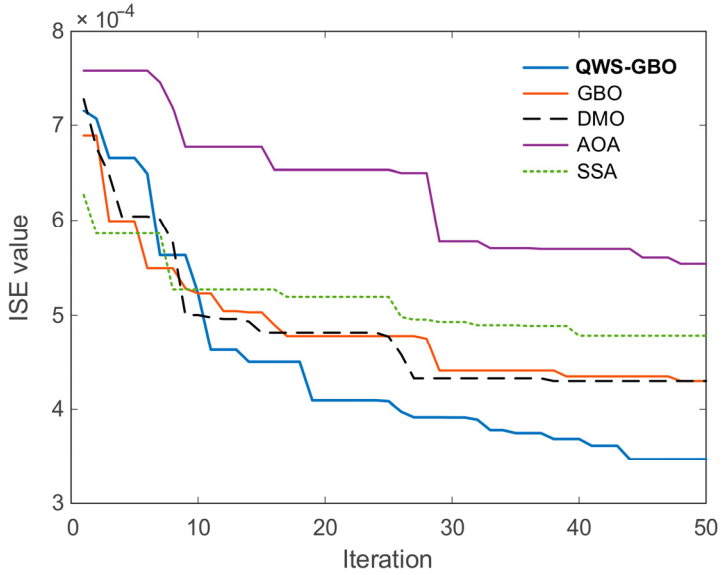


Figure 7. Changes of ISE objection function.

The accelerated convergence of QWS–GBO can be attributed to the complementary roles of its two enhancement mechanisms. The quadratic interpolation mutation guides the population toward promising unexplored regions using fitness-informed estimates, thereby improving global search efficiency. Meanwhile, the wavelet mutation strategy introduces progressively shrinking perturbations that enhance local exploitation and prevent premature stagnation. The cooperative action of these mechanisms explains the consistently lower objective values observed in Figure 7.

Upon minimizing the ISE cost function, the controller parameters outlined in Table 5 are obtained. Utilizing these values, along with the DC motor parameters from Appendix B, and substituting them into Equation (37), the transfer functions of the FOPID-controlled DC motor system designed with QWS–GBO, GBO, DMO, AOA, and SSA are derived as presented in Equations (41)–(45).

$$TF_{QWS-GBO}(s) = \frac{0.13055s^{1.9343} + 0.29318s^{1.0214} + 0.29525}{0.00108s^{3.0214} + 0.0061s^{2.0214} + 0.13055s^{1.9343} + 0.29481s^{1.0214} + 0.29525} \tag{41}$$

$$TF_{GBO}(s) = \frac{0.11668s^{1.8946} + 0.27157s^{0.9685} + 0.071507}{0.00108s^{2.9685} + 0.0061s^{1.9685} + 0.11668s^{1.8946} + 0.2732s^{0.9685} + 0.071507} \tag{42}$$

$$TF_{DMO}(s) = \frac{0.1348s^{2.174} + 0.24285s^{1.2616} + 0.080955}{0.00108s^{3.2616} + 0.0061s^{2.2616} + 0.1348s^{2.174} + 0.24448s^{1.2616} + 0.080955} \tag{43}$$

$$TF_{AOA}(s) = \frac{0.11796s^{1.7985} + 0.18876s^{0.8831} + 0.2989}{0.00108s^{2.8831} + 0.0061s^{1.8831} + 0.11796s^{1.7985} + 0.19039s^{0.8831} + 0.2989} \tag{44}$$

$$TF_{SSA}(s) = \frac{0.10092s^{1.9066} + 0.25883s^{0.9562} + 0.13758}{0.00108s^{2.9562} + 0.0061s^{1.9562} + 0.10092s^{1.9066} + 0.26046s^{0.9562} + 0.13758} \tag{45}$$

Table 5. Obtained controller parameters via different algorithms.

Algorithm	K_p	K_i	K_d	λ	μ
QWS–GBO	19.5451	19.6834	8.7031	1.0214	0.9129
GBO	18.1044	4.76713	7.7785	0.9685	0.9261
DMO	16.1897	5.3970	8.9867	1.2616	0.9124
AOA	12.5842	19.9270	7.8642	0.8831	0.9154
SSA	17.2552	9.1723	6.7281	0.9562	0.9504

It is important to note that the five-dimensional FOPID tuning problem produces a more intricate and multimodal search landscape than classical PID tuning. The enhanced exploration capability provided by QIM helps the optimizer escape shallow local minima associated with fractional-order interactions, while the adaptive refinement of WMS improves sensitivity to narrow optimal regions. This characteristic makes QWS–GBO particularly well suited to the FOPID parameter optimization problem.

7.3. Time Response Analysis

This section presents a detailed stability analysis of the FOPID-controlled DC motor system, designed using the QWS–GBO, GBO, DMO, AOA, and SSA algorithms, with a particular focus on time response. Figure 8 compares the speed responses of the DC motor systems designed with these different algorithms, while Figure 9 provides a closer, zoomed-in view of the QWS–GBO's performance. The visual data in these figures clearly

show that the DC motor system designed with QWS–GBO delivers a smoother and more precise speed response. In contrast, systems designed with competing algorithms exhibit less consistent behavior. This highlights the superior stability of the QWS–GBO-based FOPID-controlled system, ensuring a more controlled and stable speed profile.

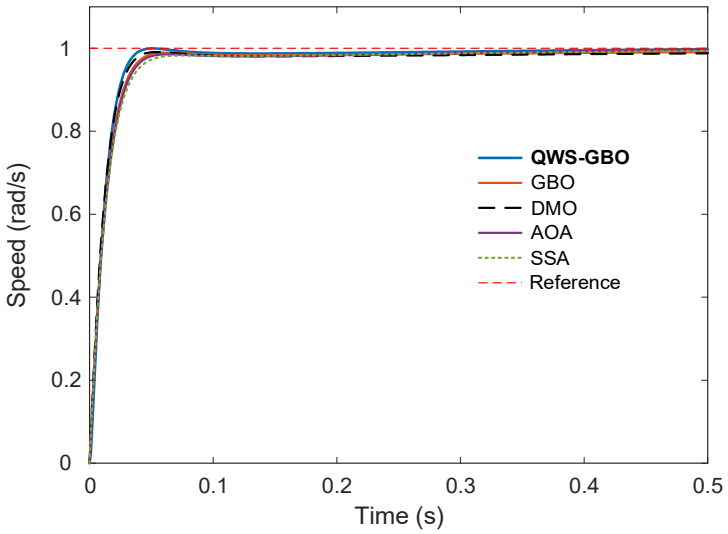


Figure 8. Speed comparison of FOPID-controlled DC motor systems designed via QWS–GBO, GBO, DMO, AOA, and SSA.

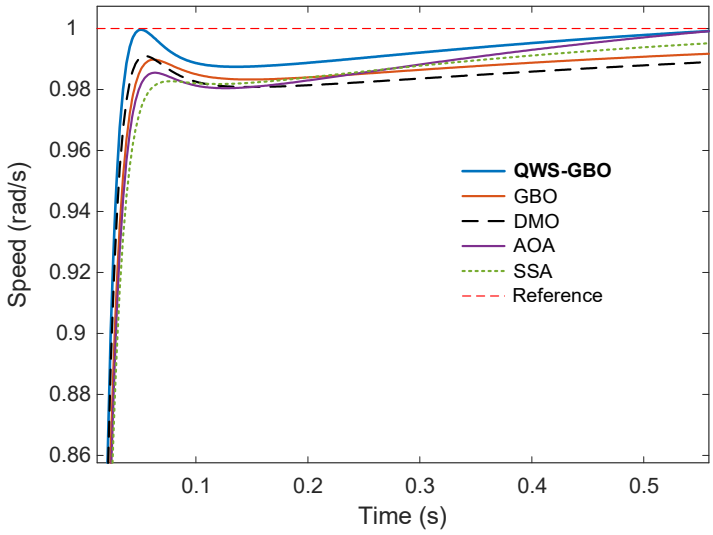


Figure 9. Zoomed illustration of the speed comparison presented in Figure 8.

The numerical results obtained for time response analysis are summarized in Table 6. As is evident from the data, the QWS–GBO algorithm leads to a minimal rise time of 0.0216 s, a settling time of 0.035 s, and zero overshoot. The proposed QWS–GBO-tuned FOPID controller achieves a smooth time response with zero overshoot and rapid settling time, which are clear indicators of the system's stability in practical scenarios. In comparison, the competitors, including GBO, DMO, AOA, and SSA, exhibit slightly longer rise times, longer settling times, and varying degrees of overshoot. This reinforces the effectiveness of the QWS–GBO algorithm in achieving a more rapid response and superior stability, contributing to enhanced performance in time response analysis.

7.4. Robustness of Proposed QWGB-Based FOPID Controller

To assess the robustness of the proposed QWS–GBO-based FOPID controller, various operating scenarios of the DC motor system were considered, and their corresponding time response results were analyzed. The key parameters for different scenarios are provided in Table 7, wherein operating points represented by the torque constant (K_m) and armature resistance (R_a) were altered to simulate different conditions. The corresponding numerical time response results for each scenario are presented in Table 8, focusing on rise time, settling time, overshoot percentage, and peak time.

Table 6. Numerical results obtained for time response analysis.

Algorithm	t_r (s)	t_s (s)	$\%OS$	t_p (s)
QWS–GBO	0.0216	0.0350	0	0.0509
GBO	0.0258	0.0449	0	0.0625
DMO	0.0234	0.0397	0	0.0546
AOA	0.0269	0.0491	0.2439	0.6728
SSA	0.0284	0.0594	0.0859	0.9971

Table 7. Operating points of DC motor.

Scenario	K_m	R_a
Nominal	0.015	0.4
(I)	0.012	0.3
(II)	0.018	0.3
(III)	0.012	0.5
(IV)	0.018	0.5

Table 8. Numerical time response results obtained for different scenarios.

Scenario	t_r (s)	t_s (s)	$\%OS$	t_p (s)
Nominal	0.0216	0.0350	0	0.0509
(I)	0.0273	0.0458	0	0.0640
(II)	0.0178	0.0282	0.4650	0.0424
(III)	0.0274	0.0463	0	0.0640
(IV)	0.0178	0.0283	0.3951	0.0424

The nominal scenario, reflecting standard operating conditions with $K_m = 0.015$ and $R_a = 0.4$, serves as a baseline for comparison. Under these conditions, the QWS–GBO-based FOPID controller demonstrates a minimal rise time of 0.0216 s, settling time of 0.035 s, zero overshoot, and a peak time of 0.0509 s. Considering all the scenarios together, for the rise time, the maximum deviation is 0.0058 s, for the settling time, the maximum deviation is 0.0113 s, for the overshoot, the maximum deviation is 0.465%, and for the peak time, the maximum deviation is 0.0131 s. The significantly small magnitudes of these numerical values serve as an indicator of the robustness of the proposed controller even under diverse operating scenarios. The minimal deviations in key performance metrics across different operating scenarios highlight the robust stability of the proposed controller, even under significant parameter variations.

It is worth noting that the favorable robustness observed in the DC motor study is consistent with the strong performance of QWS–GBO on the high-dimensional CEC2020 benchmark suite. This consistency suggests that the proposed optimizer is not limited to the present application and can be extended to more complex control design problems.

7.5. Comparisons of Time Response Metrics with Other Published Works

In this section, the performance of the proposed QWS–GBO-based FOPID controller is compared with several reported works utilizing different optimization algorithms for the design of FOPID and PID controllers. The considered approaches in the literature include an enhanced hybrid stochastic fractal search (HSFS) based FOPID controller [25], a manta ray foraging optimization (MRFO) based FOPID controller [23], an atom search optimization (ASO) based FOPID controller [20], a grey wolf optimizer (GWO) based FOPID controller [22], an enhanced arithmetic optimization algorithm (AOA-HHO) based PID controller [26], an opposition-based Henry gas solubility optimization (OBL/HGSO) based PID controller [29], an improved model of a marine predator algorithm (MMPA) based PID controller [27], and a hybrid Lévy flight distribution and Nelder–Mead algorithm (LFDNM) based PID controller [6]. Table 9 provides a summary of the reported PID and FOPID parameters in the published works, offering insights into the controller configurations used by different optimization algorithms.

Table 9. Reported PID and FOPID parameters in the published works.

System Type	Ref.	K_p	K_i	K_d	λ	μ
HSFS-based FOPID	[25]	19.23	6.001	3.0159	0.9934	0.9732
MRFO-based FOPID	[23]	19.0527	6.3585	5.3293	0.9466	0.9222
ASO-based FOPID	[20]	19.3282	7.9728	4.7805	0.9755	0.9428
GWO-based FOPID	[22]	18.328	4.9418	3.2612	0.9998	0.9845
AOA-HHO-based PID	[26]	14.435	0.1636	1.7620	1	1
OBL/HGSO-based PID	[29]	16.9327	0.9508	2.8512	1	1
MMPA-based PID	[27]	20	0.7448	1.7395	1	1
LFDNM-based PID	[6]	19.5593	5.1566	3.4097	1	1

Figure 10 illustrates a comparative analysis of speed responses between the proposed QWS–GBO-based FOPID controller and the counterparts reported in the literature. The comparison serves as a visual representation of the controllers' efficiency in regulating the speed of the DC motor. Figure 11 presents a more detailed and zoomed-in illustration of the speed comparison shown in Figure 10. This figure provides a closer examination of the speed response characteristics, emphasizing the distinctive features of each controller.

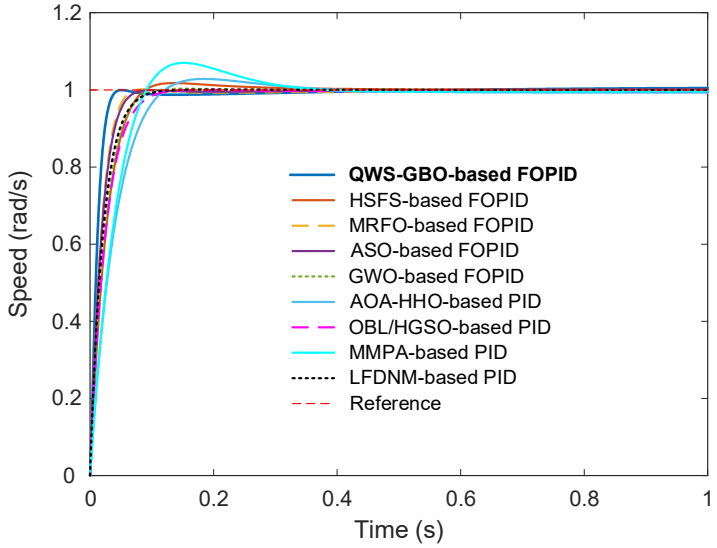


Figure 10. Comparative speed comparisons with published works.

Table 10 quantifies the performance evaluation metrics, including rise time, settling time, overshoot, and peak time, for each system type considered in the comparison. The QWS–GBO-based FOPID controller demonstrates superior performance with a minimal rise time, settling time, overshoot, and peak time, highlighting its efficacy in comparison to the reported works. The comprehensive comparison showcases the competitive edge of the proposed QWS–GBO-based FOPID controller, emphasizing its efficiency in regulating the speed of the DC motor when compared to existing optimization-based controllers reported in the literature.

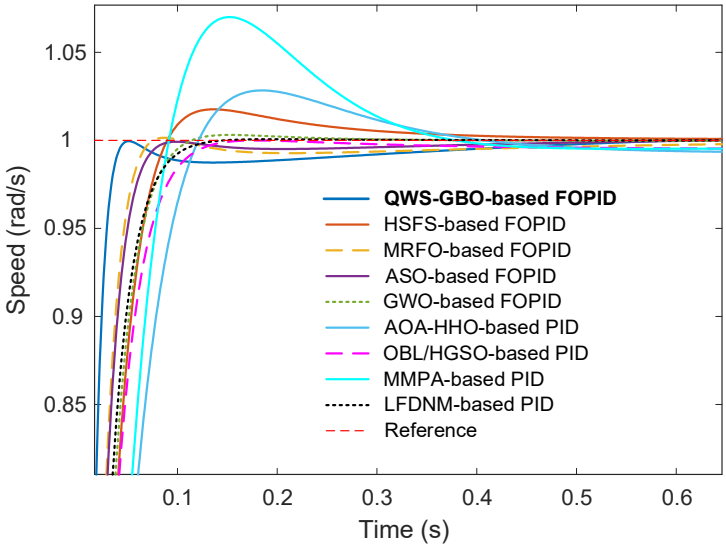


Figure 11. Zoomed illustration of the speed comparison presented in Figure 10.

Table 10. Comparative performance evaluation with reported works.

System Type	t_r (s)	t_s (s)	$\%OS$	t_p (s)
QWS–GBO-based FOPID (proposed)	0.0216	0.0350	0	0.0509
HSFS-based FOPID	0.0505	0.0766	1.7674	0.1350
MRFO-based FOPID	0.0355	0.0562	0.1546	0.0868
ASO-based FOPID	0.0376	0.0616	0	0.0987
GWO-based FOPID	0.0488	0.0814	0.3145	0.1528
AOA-HHO-based PID	0.0743	0.2508	2.8376	0.1832
OBL/HGSO-based PID	0.0546	0.0946	0	0.1749
MMPA-based PID	0.0635	0.2793	7.0059	0.1516
LFDNM-based PID	0.0462	0.0813	0	0.1452

7.6. Limitations and Threats to Validity

Several limitations delimit the scope of the present study and frame the interpretation of its results. First, the controller parameters were tuned offline under fixed operating assumptions. Consequently, sustained shifts in conditions may require retuning or an adaptive/online extension of the optimizer. Second, robustness was examined through parameter perturbations representative of typical DC motor variability, whereas unmodeled effects—such as pronounced friction nonlinearities, dead zones, actuator saturation, or time delays—were not explicitly included and could dominate in certain regimes. Third, the fractional operators in the FOPID were realized via standard approximations with fixed order and frequency band. These introduce bounded magnitude/phase errors and may slightly alter closed-loop behavior under different selections. Fourth, the chosen ISE objective strongly penalizes large errors but does not explicitly weight their duration. Consequently, alternative indices such as IAE, ITAE, or ITSE may yield different trade-offs

when temporal performance is prioritized. Incorporating such criteria within a unified or multi-objective framework remains an important topic for future investigation. Fifth, the QWS–GBO introduces a modest runtime overhead relative to baseline GBO due to its added mutation mechanisms. While acceptable for offline tuning, this cost may be non-trivial for strict real-time or embedded deployments without further engineering. Sixth, the empirical evidence is limited to single-objective, unconstrained testbeds (CEC functions and a DC motor). Constrained, multi-objective, or MIMO settings—and a broader baseline set—remain outside the present scope. Seventh, the reference–follower architecture presumes an idealized (Bode-type) reference model. In plants with strong non-minimum-phase behavior, tight input constraints, or substantial unmodeled dynamics, achievable tracking can be limited. Finally, the validation is simulation-based; hardware-in-the-loop and real-time experiments will be pursued to corroborate the findings under sensor noise, quantization, and actuator constraints. These considerations point to concrete avenues for future work, including adaptive/online tuning, duration-weighted or multi-objective design, constrained and MIMO extensions, and hardware validation.

8. Conclusions

This study presented a unified optimization–control framework for high-performance DC motor speed regulation, centered on the development of the QWS–GBO and its integration within a reference–follower control architecture. The proposed optimizer extends the original gradient-based optimizer by incorporating quadratic interpolation mutation and a wavelet mutation strategy, thereby strengthening population diversity, mitigating stagnation, and establishing a more effective balance between global exploration and local exploitation. The effectiveness of QWS–GBO was first verified on the CEC2020 benchmark suite under a uniform experimental protocol. Across all test functions, the proposed algorithm demonstrated consistently superior convergence behavior and statistical robustness when compared with representative contemporary optimizers. Lower mean fitness values and reduced dispersion across independent runs confirmed that the introduced mechanisms enhance both accuracy and repeatability while incurring only a marginal increase in computational cost. The optimizer was subsequently embedded in a master–slave control structure for DC motor speed regulation wherein a Bode-based ideal reference model defined the desired dynamics and a fractional-order PID controller acted as the follower. The controller parameters were tuned by QWS–GBO using the integral of squared error as the objective. In this application, the proposed framework achieved fast rise and settling times, zero overshoot under nominal conditions, and stable peak behavior. Robustness assessments conducted under multiple operating scenarios revealed only minor deviations in transient indices, indicating that the reference-tracking objective was preserved despite parameter variations and disturbances. Comparative analyses against both baseline optimizers and published optimization-based controllers further confirmed that the proposed approach delivers competitive, and often superior, dynamic performance. Taken together, the results demonstrate that the coordinated integration of quadratic interpolation and wavelet-based mutation within the GBO framework yields a reliable and efficient optimizer for complex control-oriented problems. When coupled with a reference–follower architecture and a fractional-order controller, this optimizer provides a systematic and reproducible pathway to high-quality control design in systems characterized by nonlinear dynamics and stringent performance requirements. From a practical perspective, the proposed QWS–GBO framework is not restricted to DC motor systems. Because the optimizer operates in a model-free manner and the reference–follower structure is plant-independent, the methodology can be directly extended to other servo drives, electromechanical actuators, and process-control loops that require high-precision

tracking. The modest computational overhead observed in this study further supports its suitability for offline industrial tuning workflows.

Future work may extend the present framework in several directions. Alternative performance indices, such as ITAE or multi-objective formulations, may be incorporated to explicitly balance error magnitude and duration. The applicability of QWS–GBO to other classes of plants, including multivariable and strongly nonlinear systems, may also be investigated. In addition, real-time and hardware-in-the-loop implementations would offer further insight into computational feasibility and practical deployment. These avenues are expected to broaden the scope of the proposed method and reinforce its relevance in advanced control applications.

Author Contributions: D.I. and S.E.: Conceptualization, Methodology, Software, Visualization, Investigation and Writing—Original draft preparation; R.M.R.-A. and M.A.A.: Data curation, Validation, and Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MTUN Strategic Collaboration Research Grant, RDU243501.

Data Availability Statement: All related data are presented within the manuscript.

Acknowledgments: The highest gratitude is especially extended to the Universiti Malaysia Pahang Al-Sultan Abdullah for the financial assistance provided under the MTUN Strategic Collaboration Research Grant, RDU243501.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Details of CEC2020 benchmark functions.

Function Definition	Range	Dimension	Global
$F_{CEC2020-1}$ (Shifted and rotated bent cigar function)	$[-100, 100]$	10	100
$F_{CEC2020-2}$ (Shifted and rotated Schwefel's function)	$[-100, 100]$	10	1100
$F_{CEC2020-3}$ (Shifted and rotated Lunacek bi-Rastrigin function)	$[-100, 100]$	10	700
$F_{CEC2020-4}$ (Expanded Rosenbrock's plus Griewank's function)	$[-100, 100]$	10	1900
$F_{CEC2020-5}$ (Hybrid function 1 ($N = 3$))	$[-100, 100]$	10	1700
$F_{CEC2020-6}$ (Hybrid function 2 ($N = 4$))	$[-100, 100]$	5	1600
$F_{CEC2020-7}$ (Hybrid function 3 ($N = 5$))	$[-100, 100]$	10	2100
$F_{CEC2020-8}$ (Composition function 1 ($N = 3$))	$[-100, 100]$	10	2200
$F_{CEC2020-9}$ (Composition function 2 ($N = 4$))	$[-100, 100]$	10	2400
$F_{CEC2020-10}$ (Composition function 3 ($N = 5$))	$[-100, 100]$	10	2500

Appendix B

Table A2. Parameters of DC motor.

Parameter	Value
R_a	0.4 Ω
L_a	2.7 H
J	0.0004 kg × m^2
B	0.0022 N × m × s/rad
K_m	0.015 N × m/A
K_b	0.05 V × s/rad

References

1. Ayinla, S.L.; Amosa, T.I.; Ibrahim, O.; Rahman, S.; Bahashwan, A.A.; Mostafa, M.G.; Yusuf, A.O. Optimal Control of DC Motor Using Leader-Based Harris Hawks Optimization Algorithm. *Frankl. Open* **2024**, *6*, 100058. [CrossRef]
2. Xie, W.; Wang, J.S.; Wang, H.B. PI Controller of Speed Regulation of Brushless DC Motor Based on Particle Swarm Optimization Algorithm with Improved Inertia Weights. *Math. Probl. Eng.* **2019**, *2019*, 2671792. [CrossRef]
3. Ibrahim, E.K.; Issa, A.H.; Gitaffa, S.A. Optimization and Performance Analysis of Fractional Order PID Controller for DC Motor Speed Control. *J. Eur. Des Systèmes Autom.* **2022**, *55*, 723–739. [CrossRef]
4. Jabari, M.; Ekinci, S.; Izci, D.; Bajaj, M.; Zaitsev, I. Efficient DC Motor Speed Control Using a Novel Multi-Stage FOPD(1 + PI) Controller Optimized by the Pelican Optimization Algorithm. *Sci. Rep.* **2024**, *14*, 22442. [CrossRef] [PubMed]
5. Manuel, N.L.; İnanç, N.; Lüy, M. Control and Performance Analyses of a DC Motor Using Optimized PIDs and Fuzzy Logic Controller. *Results Control Optim.* **2023**, *13*, 100306. [CrossRef]
6. Izci, D. Design and Application of an Optimally Tuned PID Controller for DC Motor Speed Regulation via a Novel Hybrid Lévy Flight Distribution and Nelder–Mead Algorithm. *Trans. Inst. Meas. Control* **2021**, *43*, 3195–3211. [CrossRef]
7. Sharma, V.; Kumar, V.; Naresh, R.; Kumar, V. Robust Control of DC Motor Using Model Predictive Controller Strategy. In *Proceedings of the Third International Conference on Advances in Physical Sciences and Materials: ICAPSM 2022, Coimbatore, India, 18–19 August 2022*; AIP Publishing LLC: Melville, NY, USA, 2023; p. 070008.
8. Vanchinathan, K.; Selvaganesan, N. Adaptive Fractional Order PID Controller Tuning for Brushless DC Motor Using Artificial Bee Colony Algorithm. *Results Control Optim.* **2021**, *4*, 100032. [CrossRef]
9. Alkrwy, A.; Hussein, A.A.; Atyia, T.H.; Khamees, M. Adaptive Tuning of PID Controller Using Crow Search Algorithm for DC Motor. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1076*, 012001. [CrossRef]
10. Eltoum, M.A.M.; Hussein, A.; Abido, M.A. Hybrid Fuzzy Fractional-Order PID-Based Speed Control for Brushless DC Motor. *Arab. J. Sci. Eng.* **2021**, *46*, 9423–9435. [CrossRef]
11. Farahani, G.; Rahmani, K. Speed Control of a Separately Excited DC Motor Using New Proposed Fuzzy Neural Algorithm Based on FOPID Controller. *J. Control Autom. Electr. Syst.* **2019**, *30*, 728–740. [CrossRef]
12. Puangdownreong, D. Fractional Order PID Controller Design for DC Motor Speed Control System via Flower Pollination Algorithm. *Trans. Electr. Eng. Electron. Commun.* **2019**, *17*, 14–23. [CrossRef]
13. Viola, J.; Angel, L.; Sebastian, J.M. Design and Robust Performance Evaluation of a Fractional Order PID Controller Applied to a DC Motor. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 304–314. [CrossRef]
14. Potnuru, D.; Alice Mary, K.; Sai Babu, C. Experimental Implementation of Flower Pollination Algorithm for Speed Controller of a BLDC Motor. *Ain Shams Eng. J.* **2019**, *10*, 287–295. [CrossRef]
15. Kumarasamy, V.; Ramasamy, V.K.; Chandrasekaran, G.; Chinnaraj, G.; Sivalingam, P.; Kumar, N.S. A Review of Integer Order PID and Fractional Order PID Controllers Using Optimization Techniques for Speed Control of Brushless DC Motor Drive. *Int. J. Syst. Assur. Eng. Manag.* **2023**, *14*, 1139–1150. [CrossRef]
16. Torabi, E.; Ghobaei-Arani, M.; Shahidinejad, A. Data Replica Placement Approaches in Fog Computing: A Review. *Clust. Comput.* **2022**, *25*, 3561–3589. [CrossRef]
17. Saleem, O.; Ali, S.; Iqbal, J. Robust MPPT Control of Stand-Alone Photovoltaic Systems via Adaptive Self-Adjusting Fractional Order PID Controller. *Energies* **2023**, *16*, 5039. [CrossRef]
18. Reiss-Mirzaei, M.; Ghobaei-Arani, M.; Esmaeili, L. A Review on the Edge Caching Mechanisms in the Mobile Edge Computing: A Social-Aware Perspective. *Internet Things* **2023**, *22*, 100690. [CrossRef]
19. Etemadi, M.; Ghobaei-Arani, M.; Shahidinejad, A. A Cost-Efficient Auto-Scaling Mechanism for IoT Applications in Fog Computing Environment: A Deep Learning-Based Approach. *Clust. Comput.* **2021**, *24*, 3277–3292. [CrossRef]
20. Hekimoglu, B. Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm. *IEEE Access* **2019**, *7*, 38100–38114. [CrossRef]
21. Jain, R.V.; Aware, M.; Junghare, A. Tuning of Fractional Order PID Controller Using Particle Swarm Optimization Technique for DC Motor Speed Control. In *Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 4–6 July 2016*; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2016; pp. 1–4.
22. Agarwal, J.; Parmar, G.; Gupta, R.; Sikander, A. Analysis of Grey Wolf Optimizer Based Fractional Order PID Controller in Speed Control of DC Motor. *Microsyst. Technol.* **2018**, *24*, 4997–5006. [CrossRef]
23. Ekinci, S.; Izci, D.; Hekimoğlu, B. Optimal FOPID Speed Control of DC Motor via Opposition-Based Hybrid Manta Ray Foraging Optimization and Simulated Annealing Algorithm. *Arab. J. Sci. Eng.* **2021**, *46*, 1395–1409. [CrossRef]
24. Izci, D.; Ekinci, S.; Zeynelgil, H.L.; Hedley, J. Fractional Order PID Design Based on Novel Improved Slime Mould Algorithm. *Electr. Power Compon. Syst.* **2021**, *49*, 901–918. [CrossRef]
25. Saini, R.; Parmar, G.; Gupta, R. An Enhanced Hybrid Stochastic Fractal Search FOPID for Speed Control of DC Motor. In *Fractional Order Systems and Applications in Engineering*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 51–67.

26. Issa, M. Enhanced Arithmetic Optimization Algorithm for Parameter Estimation of PID Controller. *Arab. J. Sci. Eng.* **2023**, *48*, 2191–2205. [CrossRef]
27. Ramezani, M.; Bahmanyar, D.; Razmjooy, N. A New Improved Model of Marine Predator Algorithm for Optimization Problems. *Arab. J. Sci. Eng.* **2021**, *46*, 8803–8826. [CrossRef]
28. Khalilpour, M.; Razmjooy, N.; Hosseini, H.; Moallem, P. Optimal Control of DC Motor Using Invasive Weed Optimization (IWO) Algorithm. In *Proceedings of the Majlesi Conference on Electrical Engineering, Isfahan, Iran, 24 August 2011*; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2011; pp. 1–6.
29. Ekinci, S.; Hekimoğlu, B.; Izci, D. Opposition Based Henry Gas Solubility Optimization as a Novel Algorithm for PID Control of DC Motor. *Eng. Sci. Technol. Int. J.* **2021**, *24*, 331–342. [CrossRef]
30. Ekinci, S.; Izci, D.; Yilmaz, M. Efficient Speed Control for DC Motors Using Novel Gazelle Simplex Optimizer. *IEEE Access* **2023**, *11*, 105830–105842. [CrossRef]
31. Lan, Y.; Zheng, S. Improved Fractional Model-Free Adaptive Control and Parameters Optimization for a Class of Nonlinear Discrete Systems. *J. Frankl. Inst.* **2026**, *363*, 108377. [CrossRef]
32. Izci, D.; Artun, F.; Ekinci, S.; Bajaj, M.; Blazek, V.; Zaitsev, I. Enhanced Hiking Optimization Algorithm for Robust PID Control in Doubly-Fed Induction Generator Systems for Wind Energy Applications. *IET Control Theory Appl.* **2025**, *19*, e70071. [CrossRef]
33. Nejlaoui, M.; Alghafis, A.; Alqahtani, N.A. Optimal Fractional Order PID Controller Design for Hydraulic Turbines Using a Multi-Objective Imperialist Competitive Algorithm. *Fractal Fract.* **2026**, *10*, 46. [CrossRef]
34. Munagala, V.K.; Jatoth, R.K. A Novel Approach for Controlling DC Motor Speed Using NARXnet Based FOPID Controller. *Evol. Syst.* **2023**, *14*, 101–116. [CrossRef]
35. Mujammal, M.A.H.; Moualdia, A.; Boulkhrachef, S.; Wira, P.; Boudana, D.; Albasheri, M.A. Advancing Wind Energy Conversion: Smart Maximum Power Point Based on M5-Pruned Algorithm for Enhanced Wind Energy Production. *Prod. Eng.* **2025**, *19*, 347–368. [CrossRef]
36. Mujammal, M.A.H.; Moualdia, A.; Boudana, D.; Wira, P.; Cherifi, A. Implementation of Smart Direct Power Control Strategy Based on M5-Pruned Algorithm for Three-Phase Pulse Width Modulation Rectifier. *Energy* **2025**, *315*, 134417. [CrossRef]
37. Mujammal, M.A.H.; Moualdia, A.; Wira, P.; Albasheri, M.A.; Cherifi, A. Novel Direct Power Control Based on Grid Voltage Modulated Strategy Using Artificial Intelligence. *Smart Grids Sustain. Energy* **2024**, *9*, 37. [CrossRef]
38. Silaa, M.Y.; Barambones, O.; Cortajarena, J.A.; Alkorta, P.; Bencherif, A. PEMFC Current Control Using a Novel Compound Controller Enhanced by the Black Widow Algorithm: A Comprehensive Simulation Study. *Sustainability* **2023**, *15*, 13823. [CrossRef]
39. Silaa, M.Y.; Barambones, O.; Bencherif, A. A Novel Adaptive PID Controller Design for a PEM Fuel Cell Using Stochastic Gradient Descent with Momentum Enhanced by Whale Optimizer. *Electronics* **2022**, *11*, 2610. [CrossRef]
40. Silaa, M.Y.; Barambones, O.; Bencherif, A. Robust Adaptive Sliding Mode Control Using Stochastic Gradient Descent for Robot Arm Manipulator Trajectory Tracking. *Electronics* **2024**, *13*, 3903. [CrossRef]
41. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
42. Ahmadianfar, I.; Bozorg-Haddad, O.; Chu, X. Gradient-Based Optimizer: A New Metaheuristic Optimization Algorithm. *Inf. Sci.* **2020**, *540*, 131–159. [CrossRef]
43. Izci, D.; Rizk-Allah, R.M.; Ekinci, S.; Hussien, A.G. Enhancing Time-Domain Performance of Vehicle Cruise Control System by Using a Multi-Strategy Improved RUN Optimizer. *Alex. Eng. J.* **2023**, *80*, 609–622. [CrossRef]
44. Rizk-Allah, R.M.; Zineldin, M.I.; Mousa, A.A.A.; Abdel-Khalek, S.; Mohamed, M.S.; Snášel, V. On a Novel Hybrid Manta Ray Foraging Optimizer and Its Application on Parameters Estimation of Lithium-Ion Battery. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 62. [CrossRef]
45. Barbosa, R.S.; Machado, J.A.T.; Ferreira, I.M. Tuning of PID Controllers Based on Bode's Ideal Transfer Function. *Nonlinear Dyn.* **2004**, *38*, 305–321. [CrossRef]
46. Bode, H.W. *Network Analysis and Feedback Amplifier Design*; Van Nostrand: Nordstrand, Germany, 1945.
47. Ekinci, S.; Izci, D.; Bajaj, M.; Blazek, V. Novel Application of Sinh Cosh Optimizer for Robust Controller Design in Hybrid Photovoltaic-Thermal Power Systems. *Sci. Rep.* **2025**, *15*, 2825. [CrossRef] [PubMed]
48. Agushaka, J.O.; Ezugwu, A.E.; Abualigah, L. Dwarf Mongoose Optimization Algorithm. *Comput. Methods Appl. Mech. Eng.* **2022**, *391*, 114570. [CrossRef]
49. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The Arithmetic Optimization Algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
50. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
51. Agushaka, J.O.; Ezugwu, A.E.; Abualigah, L. Gazelle Optimization Algorithm: A Novel Nature-Inspired Metaheuristic Optimizer. *Neural Comput. Appl.* **2022**, *35*, 4099–4131. [CrossRef]

52. Izci, D.; Jabari, M.; Çelik, E.; Ekinci, S.; Bajaj, M.; Rubanenko, O.; Prokop, L. Designing a Cascaded Exponential PID Controller via Starfish Optimizer for DC Motor and Liquid Level Systems. *Sci. Rep.* **2025**, *15*, 44504. [CrossRef]
53. Izci, D.; Ekinci, S.; Rizk-Allah, R.M.; Alribdi, N.I.; Smerat, A.; Alzahrani, A.; Alwadain, A.; Snasel, V.; Abualigah, L. Master–Slave Architecture Enhanced and Improved GBO Tuned Cascaded PI-PDN Controller for Speed Regulation of DC Motors. *Optim. Control Appl. Methods* **2025**, *46*, 2137–2152. [CrossRef]
54. Ekinci, S.; Izci, D.; Almomani, M.H.; Saleem, K.; Zitar, R.A.; Smerat, A.; Snasel, V.; Ezugwu, A.E.; Abualigah, L. Advanced Control Parameter Optimization in DC Motors and Liquid Level Systems. *Sci. Rep.* **2025**, *15*, 1394. [CrossRef]
55. Raizada, V.; Singh, N.J.; Chopra, V.; Kaur, J.; Lamba, R. Fractional Order PID Controller Design Using Predator Effect Inspired Artificial Bee Colony Algorithm. *J. Eng. Res.* **2026**. [CrossRef]
56. Tepljakov, A.; Alagoz, B.B.; Yeroglu, C.; Gonzalez, E.A.; Hosseinnia, S.H.; Petlenkov, E.; Ates, A.; Cech, M. Towards Industrialization of FOPID Controllers: A Survey on Milestones of Fractional-Order Control and Pathways for Future Developments. *IEEE Access* **2021**, *9*, 21016–21042. [CrossRef]
57. Pradhan, R.; Majhi, S.K.; Pradhan, J.K.; Pati, B.B. Antlion Optimizer Tuned PID Controller Based on Bode Ideal Transfer Function for Automobile Cruise Control System. *J. Ind. Inf. Integr.* **2018**, *9*, 45–52. [CrossRef]
58. Alagoz, B.B.; Ates, A.; Yeroglu, C. Auto-Tuning of PID Controller According to Fractional-Order Reference Model Approximation for DC Rotor Control. *Mechatronics* **2013**, *23*, 789–797. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.