Electrical Power and Energy Systems 162 (2024) 110266

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Incorporating adaptive local search and experience-based perturbed learning into artificial rabbits optimizer for improved DC motor speed regulation

Rizk M. Rizk-Allah [a], Davut Izci [b,c], Serdar Ekinci [c], Ali Diabat [d,e,k], Absalom E. Ezugwu [f,*], Laith Abualigah [g,h,i,j,*]

[a] Basic Engineering Science Department, Faculty of Engineering, Menoufia University, Shebin El-Kom 32511, Egypt
[b] Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan
[c] Department of Computer Engineering, Batman University, Batman 72100, Turkey
[d] Division of Engineering, New York University Abu Dhabi, Saadiyat Island, 129188 Abu Dhabi, United Arab Emirates
[e] Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA
[f] Unit for Data Science and Computing, North-West University, 11 Hofman Street, Potchefstroom 2520, South Africa
[g] Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan
[h] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India
[i] MEU Research Unit, Middle East University, Amman 11831, Jordan
[j] Artificial Intelligence and Sensing Technologies (AIST) Research Center, University of Tabuk, Tabuk 71491, Saudi Arabia.
[k] Honorary Professor, School of Engineering, The University of Jordan, Amman, Jordan

ARTICLE INFO

Keywords:
DC motor
FOPID controller
Artificial rabbits optimizer
Intelligent optimization
Speed regulation

ABSTRACT

The widespread utilization of direct current (DC) motors in real-life engineering applications has led to the need for precise speed control, making controllers a crucial aspect of DC motor systems. Proportional-integral-derivative (PID) controllers have been widely adopted due to their simplicity and effectiveness. However, recent advancements have introduced fractional order PID (FOPID) controllers that offer improved control performance for complex systems with nonlinear dynamics. To fully leverage FOPID controller's benefits, an efficient tuning method is essential. In this study, we propose artificial rabbits optimization (ARO) algorithm with enhanced strategies, called IARO, to optimize the FOPID controller for DC motor speed regulation. The IARO algorithm incorporates an adaptive local search (ALS) mechanism and an experience-based perturbed learning (EPL) strategy, addressing the shortcomings of ARO and providing better exploration–exploitation balance. We validate the superiority of IARO over competitive algorithms on the CEC2020 benchmark functions, showcasing improved solution stability and consistency. The IARO algorithm is then applied to tune the FOPID controller for DC motor speed regulation. The problem is formulated as a constraint minimization task, optimizing the integral of time-weighted absolute error cost function while adhering to critical design requirements. Comparative simulations demonstrate the IARO algorithm's ability to achieve superior cost function values and faster convergence compared to other algorithms' based FOPID controllers. The IARO-based FOPID controller exhibits enhanced stability, smoother speed response, larger gain margin, and wider bandwidth compared to other reported algorithms. Additionally, a hardware implementation is also conducted to further validate the practical applicability of IARO based design method. The IARO-based FOPID controller showed remarkable accuracy in tracking multi-step reference inputs and robustly rejected external disturbances, outperforming other recent optimization-based controllers. Additionally, the IARO-based PID controller achieved better performance in key time-domain metrics, including lower overshoot, faster rise time, shorter settling time, and minimized peak time.

* Corresponding authors.
E-mail addresses: Absalom.ezugwu@nwu.ac.za (A.E. Ezugwu), aligah.2020@gmail.com (L. Abualigah).

https://doi.org/10.1016/j.ijepes.2024.110266
Received 8 August 2024; Received in revised form 22 September 2024; Accepted 25 September 2024
Available online 19 October 2024
0142-0615/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The application of direct current (DC) motors has played a pivotal role in a multitude of practical engineering scenarios. Their extensive adoption is evident in electric vehicles, machining tools, robotic arms, cranes, and a range of other industrial contexts [1]. These motors are preferred for their ease of control, exceptional durability, and cost-efficient attributes [2]. In numerous fields employing DC motors, precise speed control is a critical requirement. Achieving accurate regulation is essential given their role as primary actuation devices in dynamic systems [3]. To this end, various control structures have been adopted, including adaptive controllers [4,5] and fuzzy controllers [6,7], proportional-integral-derivative (PID) controllers [8], and fractional order PID (FOPID) controllers [9] to achieve optimal working conditions for DC motors.

PID controllers have consistently ranked as one of the most preferred types of controllers in various documented applications [10]. They have played a pivotal role not only in regulating the speed of DC motors but also in a diverse array of industrial contexts [11]. The widespread utilization of PID controllers is ascribed to their cost-effectiveness and user-friendly nature, underpinned by their straightforward yet dependable design. Nonetheless, recent progress has introduced FOPID controllers, demonstrating superior control performance compared to conventional PID controllers [12]. Leveraging advanced algorithms and optimization techniques, finely tuned parameters of optimized FOPID controllers are tailored to achieve specific control objectives. The adoption of these controllers enhances adaptability, proving particularly beneficial for complex systems with nonlinear dynamics or fluctuating parameters [13,14].

To harness the advantages of PID or FOPID controllers optimally, the adoption of an appropriate tuning methodology is imperative. In this regard, metaheuristic algorithms have risen as potent artificial intelligence optimization approaches [15]. Recent years have seen the development of several novel algorithms that enhance the performance of FOPID controllers across various engineering applications. For instance, an improved marine predators algorithm was proposed for tuning a fractional-order PID controller specifically designed for automatic voltage regulator systems, demonstrating enhanced control performance and stability [16]. Similarly, the application of a fractional-order actor-critic algorithm has shown optimal tuning of FOPID controllers by leveraging reinforcement learning principles, which offers robust adaptation in dynamic environments [17]. The walrus optimizer has been effectively employed for optimal FOPID control, significantly improving the performance of offshore wind farms by minimizing power fluctuations under varying conditions [18]. Other recent studies include the use of a grey wolf optimizer for the maximum power point tracking of grid-tied photovoltaic systems, which addresses the uncertainties in atmospheric conditions [19], and the modified harmony search optimization algorithm for precise modeling and auto-tuning of nonlinear systems [20]. Additionally, advanced algorithms such as the tree-seed optimization [21], aquila optimization [22], and hybrid cooperation search with Nelder-Mead [23] have been successfully applied to tune FOPID controllers in various complex scenarios, including industrial tank systems, hybrid power systems, and buck converters, respectively.

As listed in Table 1, various artificial intelligence techniques have been employed by researchers to tune controllers for DC motor systems (slime mould algorithm (SMA) [24], hybrid stochastic fractal search (HSFA) algorithm [25], manta ray foraging optimization (MRFO) algorithm and its hybrid version with opposition-based learning and simulated annealing (OBL-MRFO-SA) mechanisms [26], atom search optimization (ASO) algorithm and its chaotic version (ChASO) algorithm [27], grey wolf optimizer (GWO) [28], arithmetic optimization algorithm (AOA) and its enhanced version with Harris hawks optimizer (AOA-HHO) [29], covariance matrix adaptation based evolution strategy (CMA-ES) [29], improved model of marine predator algorithm (MMPA) [30], whale optimization algorithm (WOA) [30], equilibrium optimizer algorithm (EOA) [30], jaya algorithm [30], hybrid Lévy flight distribution and Nelder–Mead (LFDNM) algorithm [31], Henry gas solubility optimization (HGSO) and its opposition based version (OBL/HGSO) [32], gazelle simplex optimizer (GSO) [33], invasive weed optimization (IWO) algorithm [34] and particle swarm optimization (PSO) [34]). It is feasible to list more examples as reported in [35–41]. These methods have demonstrated improved performance to an extent as can be seen from the percent overshoot (%OS), rise time (t_r), settling time (t_s) and peak time (t_p) metrics presented in Table 1. However, the "no free lunch" theorem [42] reminds us that no single solution fits all scenarios, leaving ample space for further enhancements and developments.

Table 1
Literature review and comparative time-domain based metrics for same DC motor model.

Reference	Algorithm	Controller	%OS	t_r(s)	t_s(s)	t_p(s)
[24]	SMA	FOPID	0	0.0285	0.0451	0.0660
[25]	HSFS	FOPID	1.7674	0.0505	0.0766	0.1350
[26]	OBL-MRFO-SA	FOPID	0	0.0214	0.0339	0.0497
[26]	MRFO	FOPID	0.1546	0.0355	0.0562	0.0868
[27]	ChASO	FOPID	0	0.0253	0.0405	0.0598
[27]	ASO	FOPID	0	0.0376	0.0616	0.0987
[28]	GWO	FOPID	0.3145	0.0488	0.0814	0.1528
[29]	AOA-HHO	PID	2.8376	0.0743	0.2508	0.1832
[29]	AOA	PID	37.7541	0.0821	0.7138	0.1957
[29]	CMA-ES	PID	44.8522	0.0799	0.8893	0.1963
[30]	MMPA	PID	7.0059	0.0635	0.2793	0.1516
[30]	WOA	PID	5.9537	0.0627	0.2702	0.1503
[30]	EOA	PID	6.7823	0.0634	0.2758	0.1502
[30]	JAYA	PID	6.9365	0.0636	0.2766	0.1515
[31]	LFDNM	PID	0	0.0462	0.0813	0.1452
[32]	OBL/HGSO	PID	0	0.0546	0.0946	0.1749
[32]	HGSO	PID	0	0.0684	0.1186	0.2187
[33]	GSO	PID	0	0.0365	0.0650	0.1218
[34]	IWO	PID	6.9759	0.4189	1.2533	0.8544
[34]	PSO	PID	24.2406	0.3560	1.8028	0.8498

Despite the existence of numerous optimization algorithms in the literature, selecting an appropriate algorithm to effectively tune a controller remains a challenge. The artificial rabbits optimization (ARO) algorithm [43], a recent metaheuristic method, stands out due to its simplicity, effectiveness, and flexibility. Unlike many other algorithms, such as particle swarm optimization, atom search optimization, differential evolution, and others, ARO does not require additional tunable parameters beyond the population size and total iteration number. This characteristic reduces the complexity of implementation and mitigates the risk of improper parameter tuning, which can significantly affect optimization performance. Moreover, the ARO has demonstrated superior performance on various benchmark function sets and engineering design problems. Its algorithmic structure allows for easy integration with different enhancement mechanisms, making it a prime candidate for further improvement and application.

Similarly, the selection of the FOPID controller over other PID variants, such as neuroendocrine PID [44], sigmoid PID [45], and PID acceleration (PIDA) [46], is driven by its unique benefits. The FOPID controller's fractional derivative (λ) and integral (μ) orders provide a more robust and stable response across a wider frequency range, making it highly effective in handling complex systems with nonlinear dynamics or fluctuating parameters [47]. Additionally, it offers better stability under parameter variations and enhanced noise suppression capabilities compared to traditional PID controllers and other variants. Given these advantages, the FOPID controller, tuned with an enhanced ARO algorithm, offers a promising solution for optimizing performance in applications like DC motor speed regulation.

As part of this motivation, we aim to construct a system structure with better stability performance than the approaches in the literature

such that better performance metrics reported in Table 1 can be achieved. Therefore, in this study, the authors sought to apply a new intelligent optimization approach to design an efficient FOPID controller for a DC motor system. To achieve this, we adopted the ARO algorithm as the base optimization technique, further enhancing its capabilities with adaptive local search (ALS) and an experience-based perturbed learning (EPL) strategies. The ARO algorithm has its advantages, including simplicity of implementation and quick convergence to good solutions by identifying the best participants in the search region. However, it faces significant challenges when dealing with complex and multimodal problems, struggling to strike the right balance between exploration and exploitation [48]. A major issue lies in the rabbits' iterative process, where they randomly select burrows. While this acceleration can expedite convergence, it might compromise solution diversity and confine the algorithm within local optima. Furthermore, ARO lacks a strategic guidance system for traversing toward promising regions during the search, potentially resulting in suboptimal final solutions. Therefore, there exists a distinct necessity to augment the conventional ARO algorithm's efficacy in order to address these shortcomings. By incorporating ALS and EPL strategies as two essential components, the improved version called IARO algorithm aims to overcome the challenges posed by complex problems and improve the overall solution quality.

The performance of this improved ARO (IARO) algorithm was initially evaluated on challenging CEC2020 test suite in order to showcase its superiority over the basic form of the ARO algorithm. The experimental results on the CEC2020 benchmark functions indicate that the proposed IARO algorithm outperforms the ARO algorithm in terms of finding competitive solutions with improved stability and consistency. The IARO algorithm demonstrates its efficacy by achieving near optimal or optimal solutions for several benchmark functions while maintaining lower standard deviations compared to the ARO algorithm. Apart from original ARO [43], the IARO achieves better performance compared to other competitive and efficient approaches such as dwarf mongoose optimization [49], particle swarm optimization [50], differential evolution [51] and biogeography-based optimization [52], as well. These findings suggest that the IARO algorithm holds promise as a robust optimization technique for solving complex real-world problems.

Subsequently, the IARO algorithm was employed to tune the FOPID controller for the DC motor speed regulation system. In this context, this paper conducts an in-depth exploration of the fusion of the FOPID controller with an externally excited DC motor, leveraging the capabilities of the IARO algorithm to achieve optimal performance. To attain a practical and effective solution for regulating the speed of the DC motor, the challenge is framed as a task of constraint minimization. The IARO algorithm, with its adaptive and iterative optimization approach, is employed to minimize the integral of time-weighted absolute error (ITAE) cost function [53], while adhering to critical design requirements such as settling time, overshoot, gain margin, and phase margin. The proposed IARO algorithm based FOPID controller is extensively evaluated through comparative simulations, with its performance statistically analyzed against the existing ARO algorithm. The simulations demonstrate the consistent ability of the IARO algorithm to achieve superior cost function values and convergence speed, signifying its efficiency and effectiveness in optimizing the FOPID controller parameters. Through a comprehensive stability analysis, the superiority of the IARO algorithm based FOPID controlled DC motor system is confirmed. It exhibits enhanced stability, smoother speed response, higher phase margin, larger gain margin, and significantly wider bandwidth compared to other reported algorithms' based FOPID controllers, including enhanced hybrid stochastic fractal search algorithm [25], manta ray foraging optimization [26], chaotic atom search optimization algorithm [27] and grey wolf optimizer [28]. Moreover, a comparative evaluation against PID controllers optimized by other algorithms (enhanced arithmetic optimization algorithm [29], opposition-based Henry gas solubility optimization PID [32], improved model of marine predator algorithm [30], hybrid Lévy flight distribution and Nelder–Mead algorithm [31]) demonstrates the exceptional performance of the IARO algorithm based FOPID controller, emphasizing its capacity for precise control and minimal oscillations. Additionally, a hardware implementation was conducted which further validated the practical applicability of IARO based design method. In addition to its optimization capabilities, the IARO algorithm was evaluated for its tracking and disturbance rejection abilities. This evaluation demonstrated the algorithm's robustness and adaptability in handling multi-step reference inputs and external disturbances, which are critical for real-world applications. To provide a comprehensive comparison, the IARO algorithm was also applied to tune a traditional PID controller, ensuring a fair assessment against other PID-based control methods. The findings demonstrated that the IARO-based PID controller surpassed many contemporary PID control methods, exhibiting enhanced time-domain metrics including reduced overshoot, accelerated rise time, decreased settling time, and minimized peak time. The results presented in this paper underscore the immense potential of the IARO algorithm in optimizing the FOPID controller for DC motor speed regulation. The IARO algorithm based FOPID controller showcases superior stability, dynamic performance, and precise control, making it an attractive and innovative solution for various real-world applications.

The subsequent sections of this paper are structured as follows: Section 2 provides a concise overview of the ARO algorithm, detailing its mathematical foundation. Section 3 elaborates on the newly introduced proposed IARO algorithm, explaining the integration of ALS and EPL strategies. Section 4 presents the experimental statistical results obtained using CEC2020 benchmark functions to demonstrate the superiority of the IARO algorithm. Section 5 focuses on the speed regulation of the DC motor using the optimized FOPID controller. Section 6 formulates the constrained optimization problem and describes the application of the IARO algorithm in this context. Section 7 provides a comprehensive comparison of the simulation results, including stability analysis, reference tracking, and disturbance rejection capabilities of the IARO-based FOPID controller, as well as the design and performance evaluation of the IARO-based PID controller for a fair comparison against other recent methods. This section also includes the hardware implementation, further validating the proposed methodology's effectiveness in real-world applications. Section 8 provides a detailed discussion of the results, analyzing the reasons for the observed improvements. Section 9 concludes the paper and outlines potential directions for future research. Finally, Section 10 concludes the paper and suggests future research

2. Artificial rabbits optimization

The ARO algorithm is a recent and robust metaheuristic method inspired by rabbit survival behaviors [43]. It incorporates foraging strategies such as random hiding and detour foraging. The algorithm simulates a foraging mode, where rabbits seek food near other burrows to mislead potential predators and safeguard their own dwellings. Instead of consuming nearby food sources, rabbits venture to distant locations in search of sustenance. The ARO swarm's population size is determined by the number of rabbits, with each rabbit possessing an eating zone containing plants and grass, as well as multiple burrows. During the foraging phase, rabbits randomly explore other rabbits' burrows for food, updating their positions based on their selected companions and introducing perturbations. This foraging action can be expressed mathematically as follows where ρ is a mathematical operator representing the rabbits' locomotion ($\rho = E.c$), $i, j = 1, 2, \cdots, M$ and $i \neq j$.

$$\vec{\Delta}_l(t+1) = \vec{z}_j(t) + \rho.\left(\vec{z}_l(t) - \vec{z}_j(t)\right) + round(0.5.(0.05 + g_1)).n_1 \quad (1)$$

$$E = \left(e - e^{\left(\frac{1-t}{T}\right)^2} \right) .sin(2\pi g_2) \tag{2}$$

$$c(k) = \begin{cases} 1 \text{ if } k = h(u) \\ 0 \quad else \end{cases}, k = 1, \cdots, d \text{ and } u = 1, 2, \cdots, \lceil g_3.d \rceil \tag{3}$$

Here, h is a random permutation function of the dimensionality of the problem, and n_1 follows a normal distribution. Additionally, g_1, g_2 and g_3 are uniform random numbers within the interval [0,1]. Terms such as $\vec{z}_l(t)$, $\vec{\Delta}_i(t+1)$, M, T, d, *round*, and E define the current and updated positions of candidate solutions, the population size, the number of iterations, the problem dimension, the rounding procedure to the nearest integer, and the duration of the foraging round, respectively. During the exploitation phase, rabbits employ a random hiding strategy to evade predators by creating burrows in close proximity to their original positions. In each iteration, a rabbit generates multiple burrows in each dimension and selects one for refuge. This process can be expressed mathematically as follows where $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, d$, and n_2 follows a normal distribution.

$$\vec{BU}_{ij}(t) = \vec{z}_l(t) + H.h.\vec{z}_l(t) \tag{4}$$

$$H = \frac{1 - t + T}{T} g_4 \tag{5}$$

$$h(k) = \begin{cases} 1 \text{ if } k = j \\ 0 \quad else \end{cases}, k = 1, \cdots, d \tag{6}$$

In the random hiding mode, the rabbit's position is updated as follows where, $i = 1, 2, \cdots, M$.

$$\vec{\Delta}_l(t+1) = \vec{z}_l(t) + \rho.\left(g_4.BU_{ir}(t) - \vec{z}_l(t)\right) \tag{7}$$

$$h_r(k) = \begin{cases} 1 \text{ if } k = \lceil g_5.d \rceil \\ 0 \quad else \end{cases}, k = 1, \cdots, d \tag{8}$$

$$BU_{ir}(t) = \vec{z}_l(t) + H.h_r.\vec{z}_l(t) \tag{9}$$

The l^{th} rabbit's position is updated either after a detour foraging mode or a random hiding process based on fitness comparisons.

$$\vec{z}_l(t+1) = \begin{cases} \vec{z}_l(t) & f(\vec{z}_l(t)) \le f(\vec{\Delta}_l(t+1)) \\ \vec{\Delta}_l(t+1) & f(\vec{z}_l(t)) > f(\vec{\Delta}_l(t+1)) \end{cases} \tag{10}$$

If a rabbit's candidate fitness is superior to its current position's fitness, it relocates to the candidate's location as determined by either Eq. (1) or Eq. (7). The energy of the rabbits diminishes as the iteration progresses, facilitating the transition from exploratory to exploitative modes. This transition is described as:

$$EA(t) = 4\left(1 - \frac{t}{T}\right) ln\left(\frac{1}{\alpha}\right) \tag{11}$$

where α is a random number. When $EA(t) > 1$, the algorithm focuses on global exploration, while $EA(t) \le 1$ directs the algorithm toward local exploitation.

The standard ARO algorithm exhibits certain structural weaknesses that contribute to its tendency to get trapped in local optima and its imbalanced approach to exploration and exploitation. One major cause is the lack of a mechanism to adaptively adjust the balance between exploration and exploitation phases throughout the optimization process. In the standard ARO, rabbits randomly select burrows during the search, which can accelerate convergence but often leads to premature convergence to suboptimal solutions due to insufficient exploration of the solution space. Furthermore, the random hiding strategy used in ARO does not incorporate sufficient diversity in the population, making it challenging for the algorithm to escape local optima. The reliance on a relatively simple foraging and hiding process means that the ARO lacks a strategic guidance system to effectively traverse toward more promising regions. As a result, the algorithm may repeatedly explore the same regions without adequate diversification, leading to stagnation around local optima. These structural aspects (lack of adaptive mechanisms, limited population diversity, and simplistic exploration–exploitation balance) are the primary causes that limit the standard ARO's effectiveness in handling complex and multimodal optimization problems.

3. Artificial rabbits optimizer with enhanced strategies

In this section, we introduce IARO, an enhanced iteration of the ARO algorithm, aimed at addressing complexities that ARO struggles with while retaining its simplicity and fast convergence. IARO incorporates two key strategies: adaptive local search (ALS) and experience-based perturbed learning (EPL). ALS targets promising regions to avoid local optima, while EPL strikes a balance between exploration and exploitation, ultimately enhancing the effectiveness of IARO for optimization tasks. The IARO algorithm introduces a novel exploration strategy wherein rabbits follow another rabbit from the population. However, this update scheme can sometimes lead to invasive diversification patterns. To overcome this, the algorithm integrates the EPL approach, which seeks to identify more promising regions within the feasible search space and enhance exploration. The EPL process begins by calculating the mean (Δ^t_{mean}) and deviation (Δ^t_{dev}) of a randomly chosen solution concerning the best solution found so far (Δ_{best}). The updated solution can be expressed as follows.

$$\Delta^t_{mean} = (z_{best} + z^t_I)/2 \tag{12}$$

$$\Delta^t_{dev} = abs(z_{best} - z^t_I) \tag{13}$$

$$\Delta^t_C = \Delta^t_{mean} + rand_1.\Delta^t_{dev} \tag{14}$$

$$z^t_{new} = \Delta^t_C + rand_2 \cdot (z_{best} - \Delta^t_C) + 0.95 \cdot (rand_3 - 0.5) \cdot \left| z_{max,j} - z_{min,j} \right| \tag{15}$$

In this algorithmic framework, z^t_I represents any randomly chosen solution, and $rand_1, rand_2$, and $rand_3$ denote three random numbers following a uniform distribution within the interval [0,1]. Specifically, the third term in Eq. (15) introduces a perturbed solution within dynamic boundaries (z_{max} and z_{min}). The incorporation of EPL into the IARO algorithm seamlessly integrates exploration and exploitation, leading to enhanced search performance and improved convergence to optimal solutions. The pseudocode of the EPL strategy is presented in Algorithm 1.

Algorithm 1 Pseudocode of the EPL strategy

Input: z_I, $rand_1, rand_2$, and $rand_3$
$z_I \leftarrow$ a solution selected randomly from the current population
$rand_1, rand_2$, and $rand_3 \leftarrow$ three uniformly distributed random numbers from [0,1]
Calculate the mean between z_I and z_{best} by Eq. (12)
Calculate the deviation between z_I and z_{best} by Eq. (13)
For the diversity aspect, the position of the mean solution slightly changes according to Eq. (14)
Update the current position in terms of the z_{best} and the dynamic search space by Eq. (15)
Output: z^t_{new} (obtained solution by EPL strategy)

To refine accuracy throughout the iterative process and steer exploitation towards promising areas, we introduce the ALS strategy. This guidance mechanism leverages shared information within an elite group of rabbits, considering their best individual (z_{best}) and worst individual (z_{worst}). Initially, the elite group is identified based on the fitness function, and the worst and best individuals (z^P and z^W) are determined within this group. Subsequently, three distinct movements are executed for the updating step: (1) pushing z^W in the direction of z^P, (2) pushing z^W towards z_{best}, and (3) pushing z^W in the direction of the average of z^P

and z_{best}. These movements are executed sequentially, and the process concludes when one of them attains superior fitness. The updating phase of this strategy is expressed as follows.

$$z_l^{t+1} = \begin{cases} z_1^{ALS} = 2 \times r_1 \times \left(z^P - z^W\right) + z^W & iff\left(z_1^{ALS}\right) \le f\left(z_l^t\right) \\ z_2^{ALS} = 2 \times r_2 \times \left(z_{best} - z^W\right) + z^W & elseiff\left(z_2^{ALS}\right) \le f\left(z_l^t\right) \\ z_3^{ALS} = 2 \times r_3 \times \left(\left(z^P + z_{best}\right)/2 - z^W\right) + z^W & otherwise \end{cases} \tag{16}$$

Here, z_l^t and z_l^{t+1} represent the present and updated solutions of the i^{th} solution within the elite class. Algorithm 2 presents the pseudocode of ALS strategy while Algorithm 3 and Fig. 1, respectively, illustrate the pseudocode and flowchart of the proposed IARO algorithm.

Algorithm 2 Pseudocode of the ALS strategy

Input: z_{best}, z^P, z^W, r_1, r_2, *and* r_3

Intialize the solution for ALS phase: $z_1^{ALS} \leftarrow z_l$
$z_{best} \leftarrow$ *best individual obtained so far*
z^P, $z^W \leftarrow$ *best and worst individuals with the elite group* (*i.e., contains top ranked individuals*)
r_1, r_2, *and* $r_3 \leftarrow$ *three uniformly distributed random numbers from* [0,1]
Perform the updating step to push z^W *towards the promising regions by Eq.* (16):
Push z^W *towards* z^P : $z_1^{ALS} = 2 \times r_1 \times \left(z^P - z^W\right) + z^W$
if $f\left(z_1^{ALS}\right) < f(z_l)$
$z_l^{t+1} = z_1^{ALS}$
else
Push z^W *towards* z_{best} : $z_2^{ALS} = 2 \times r_2 \times \left(z_{best} - z^W\right) + z^W$
if $f\left(z_2^{ALS}\right) < f(z_l)$
$z_l^{t+1} = z_2^{ALS}$
else
Push z^W *towards the average of* z^P *and* z_{best} : $z_3^{ALS} = 2 \times r_3 \times \left(\left(z^P + z_{best}\right)/2 - z^W\right) + z^W$
if $f\left(z_3^{ALS}\right) < f(z_l)$
$z_l^{t+1} = z_3^{ALS}$
end
end
end
Output: z_l^{t+1} (*obtained solution by ALS strategy*)

The ALS and EPL strategies are designed to address the fundamental weaknesses of the standard ARO in dealing with local optima and the imbalance between exploration and exploitation. The ALS strategy is specifically tailored to enhance the exploitation capability of the optimizer. By dynamically focusing the search around promising regions, ALS reduces the risk of premature convergence to local optima. This is achieved through an adaptive mechanism that adjusts the search radius based on the fitness landscape, allowing the algorithm to more effectively fine-tune high-quality solutions. This approach ensures a thorough exploration of the local space around potential optima, thus reducing the likelihood of getting trapped in suboptimal regions. The EPL strategy directly addresses the balance between exploration and exploitation by introducing controlled perturbations based on the experiences of the best solutions found so far. By maintaining diversity in the population and avoiding convergence to a limited set of solutions, EPL ensures that the algorithm can explore new regions of the solution space while retaining the capability to exploit known good solutions. This mechanism prevents stagnation and encourages a more balanced search, thereby increasing the probability of finding the global optimum. Together, these strategies create a more dynamic and adaptive optimization process that effectively mitigates the problems of local optima and imbalanced search efforts. The pseudocodes provided in this section illustrate the operational details of ALS and EPL, highlighting their role in enhancing the overall search capabilities of the improved ARO.Algorithm 3 Pseudocode of the IARO

Configure the parameters of the IARO
Generate the initial swarm of rabbits in a random fashion within the search bounds.
Evaluate the fitness function and elect the best individual (z_{best})
While $t < T$ ***do***
for $i = 1 : M$
Compute the energy of rabbit (*EA*) *by Eq.* (11)
if *EA* > 1 **% *Exploration stage***
Select a rabbit randomly from the swarm
Compute the running operators
Perform detour foraging process by Eq. (1)
else **% *Exploitation stage***
Create the d burrows and randomly pick one as hiding by Eq. (9)
Carry out the random hiding by Eq. (7)
end if
Evaluate the fitness function
Update the current position of the i^{th} *rabbit in* $(t+1)^{th}$ *iteration by Eq.* (10)
Update the best individual obtained so far (z_{best})
end for
% *Invoke the EPL and ALS improvements*
Intialize the required inputs for the experience-based perturbed learning (*EPL*) *phase*
Perform the EPL phase as in Algorithm 1
Intialize the required inputs for the adaptive local search (*ALS*) *phase*
Perform the ALS phase as in Algorithm 2
Evaluate the fitness function
Update the rabbit's position using the EPL/ALS solution.
Update the best individual obtained so far (z_{best})
$t = t + 1$
end while
Output:z_{best}

4. Experimental results on CEC2020 benchmark functions

In this section, we present the experimental results of the proposed IARO algorithm on the CEC2020 benchmark functions. The benchmark functions used in this study are as presented in Table 2. The CEC2020 benchmark functions (F1 −F10) are a set of ten challenging optimization problems that are commonly used to evaluate the efficiency and robustness of optimization algorithms. Each function is characterized by its specific mathematical structure, dimensionality, and global minimum point. The objective of optimization algorithms is to find the global minimum of these functions within a given search space [54]. The parameters listed in Table 2 represent the key attributes of the CEC2020 benchmark functions used in this study. These include lower bound (*LB*) which represents the minimum boundary of the search space for each variable in the function, upper bound (*UB*) which represents the maximum boundary of the search space for each variable in the function, dimension (*D*) which indicates the number of variables or dimensions in the benchmark function, reflecting the complexity of the optimization problem and finally global optimum (F_{min}^*) that denotes the known global minimum value of the benchmark function, which the optimization algorithms aim to achieve. These parameters are critical for understanding the characteristics of the optimization landscape and for comparing the performance of different algorithms in achieving the optimal solutions.

The effectiveness and performance of the IARO algorithm are compared with the original ARO [43], dwarf mongoose optimization (DMO) algorithm [49], particle swarm optimization (PSO) [50], differential evolution (DE) algorithm [51], biogeography-based optimization (BBO) [52]. Each algorithm was executed 30 times on each of the CEC2020 benchmark functions. The total number of iterations for each run was set to 1000, and a population size of 50 individuals was used in each run.

Table 3 presents the statistical evaluation of algorithms on the CEC2020 benchmark functions. The evaluation metrics include the best fitness value, worst fitness value, average fitness value, standard deviation and rank across multiple runs of each algorithm for each function.

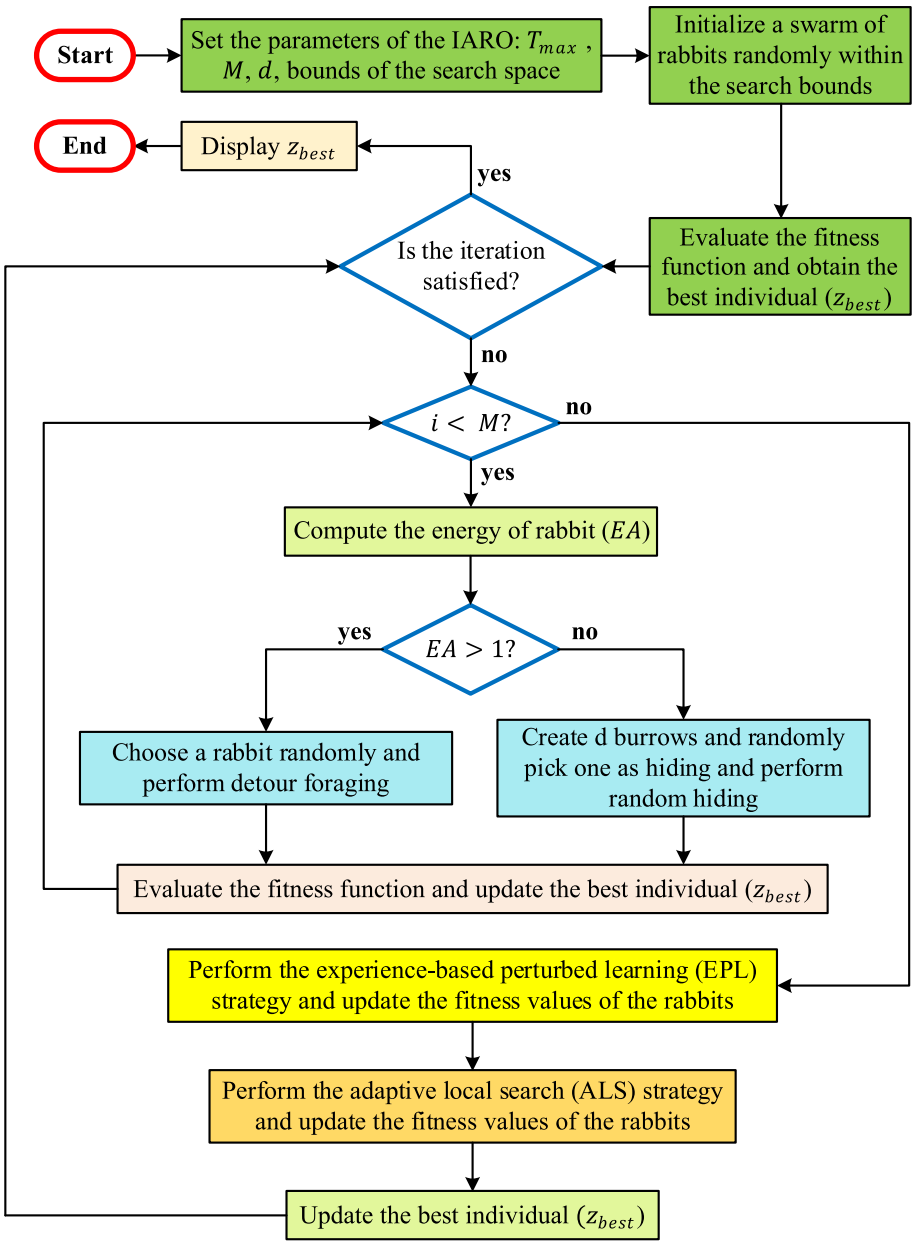


Fig. 1. The framework of the improved version of artificial rabbits optimization.

Across individual benchmark functions, IARO consistently outperformed its counterparts, delivering lower average and standard deviation values. Notably, in functions like the shifted and rotated bent cigar (F1) and Schwefel's function (F2), IARO exhibited the best solutions, showcasing its robustness. Additionally, for hybrid and composition functions (F5-F10), IARO maintained its dominance.

The comprehensive evaluation, reflected in Friedman's mean rank, reaffirms the efficacy of IARO, obtaining the lowest mean rank (1.0) and Friedman's mean rank (1.6) across all algorithms as demonstrated in Table 4. This implies that IARO not only consistently outperformed other algorithms on individual functions but also exhibited superior overall performance. The results suggest that the IARO algorithm holds promise for real-world applications requiring accurate and efficient optimization solutions.

To assess the statistical significance of the improvements achieved by the IARO algorithm compared to the standard ARO, the Wilcoxon signed-rank test was performed for each benchmark function. The results are presented in Table 5. The p-values obtained from the test indicate a statistically significant difference between the performance of IARO and ARO for most of the benchmark functions. Specifically, the p-values for functions F1, F2, F3, F5, F6, F7, F8, F9, and F10 are all below

Table 2
CEC2020 benchmark functions.

Function	Function name	*LB*	*UB*	*D*	F_{min}^{*}
F1	Shifted and rotated bent cigar function	−100	100	10	100
F2	Shifted and rotated Schwefel's function	−100	100	10	1100
F3	Shifted and rotated Lunacek bi-Rastrigin function	−100	100	10	700
F4	Expanded Rosenbrock's plus Griewank's function	−100	100	10	1900
F5	Hybrid function 1 ($N = 3$)	−100	100	10	1700
F6	Hybrid function 2 ($N = 4$)	−100	100	10	1600
F7	Hybrid function 3 ($N = 5$)	−100	100	10	2100
F8	Composition function 1 ($N = 3$)	−100	100	10	2200
F9	Composition function 2 ($N = 4$)	−100	100	10	2400
F10	Composition function 3 ($N = 5$)	−100	100	10	2500

Table 3
Statistical evaluation of the algorithms on CEC2020 benchmark functions.

Function	Algorithm	Best	Worst	Average	Standard deviation	Rank
F1	IARO	100	219.88	131.02	40.466	1
	ARO	100	5360	1246.5	1468.8	2
	DMO	409.33	22,626	5538.9	6041.8	6
	PSO	107.58	5618	1676.6	1566.3	3
	DE	326.91	5794.6	1992.1	1501.2	4
	BBO	101.45	9479.3	2057.6	2482.6	5
F2	IARO	1115.2	1381.7	1268.1	79.338	1
	ARO	1122	1956	1473.1	202.46	3
	DMO	1499.5	2439.7	2064.3	206.86	6
	PSO	1115.1	1905.4	1466.3	187.56	2
	DE	1308.2	1754.7	1552.7	111.6	4
	BBO	1250.5	2390.9	1783.9	263.22	5
F3	IARO	714.97	728.4	721.73	3.7229	3
	ARO	715.3	743.25	726.25	6.6119	5
	DMO	724.85	744.13	733.3	4.6801	6
	PSO	712.76	732.39	721.39	4.9189	2
	DE	716.08	727.08	722.16	2.245	4
	BBO	714.08	729.92	720.9	4.36	1
F4	IARO	1900	1900	1900	0	1
	ARO	1900	1900	1900	0	1
	DMO	1900.5	1902.7	1901.3	0.56783	4
	PSO	1900.5	1901.9	1901.1	0.37742	3
	DE	1900.8	1901.9	1901.6	0.28284	5
	BBO	1900.1	1904.6	1901.7	0.94168	6
F5	IARO	1703.9	1939.7	1812.7	78.918	1
	ARO	1718	12,288	2813.5	2265.9	4
	DMO	2689.5	9413.5	4343.8	1491.5	6
	PSO	2261.1	8108.7	4208	1651.5	5
	DE	4677.1	98,655	27,794	19,248	3
	BBO	2208	121,810	25,290	33,507	2
F6	IARO	1600	1612.5	1601.9	2.828	3
	ARO	1600.1	1839	1652.8	69.068	6
	DMO	1600.6	1601.1	1600.9	0.12188	2
	PSO	1600.5	1659	1612.9	11.984	5
	DE	1600.5	1600.9	1600.7	0.083246	1
	BBO	1600.5	1619.3	1603.3	5.9223	4
F7	IARO	2100	2117.1	2106.7	7.906	1
	ARO	2100	2358.7	2133.1	57.258	2
	DMO	2360	2968.9	2579.1	134.97	3
	PSO	2134.6	4599.7	2701.9	537.8	4
	DE	2239.1	5027.4	3169.1	790.72	5
	BBO	2125.9	22,806	8606.8	7009.3	6
F8	IARO	2221.8	2301.6	2293.6	22.216	2
	ARO	2223	2303.9	2296.5	19.041	3
	DMO	2239.3	2303.5	2287.8	19.046	1
	PSO	2200	2304.8	2298.7	18.67	4
	DE	2274.2	2301.6	2299.9	4.9212	5
	BBO	2300.6	2309.7	2302.6	1.6989	6
F9	IARO	2400	2745.4	2639.8	124.33	2
	ARO	2500	2761.1	2679.3	110.1	3
	DMO	2500	2614.1	2522.1	29.925	1
	PSO	2500	2764.2	2723.9	55.138	5
	DE	2560.5	2753.8	2725.6	44.927	6
	BBO	2500	2767.4	2715.7	86.578	4
F10	IARO	2600.1	2899.6	2888.7	54.515	1
	ARO	2897.7	2949.1	2917.5	22.926	5
	DMO	2898	2940.7	2903.9	9.8305	2
	PSO	2600.1	2948	2907.5	62.419	3
	DE	2899.9	2947.5	2913.9	13.589	4
	BBO	2897.8	2949.4	2932.1	22.393	6

Table 4
Friedman's mean rank for CEC2020 benchmark functions.

Metric	IARO	ARO	DMO	PSO	DE	BBO
Friedman's mean rank	1.6	3.4	3.7	3.6	4.1	4.5
Mean rank	**1**	2	4	3	5	6

0.05, confirming that the IARO algorithm significantly outperforms the ARO in these cases. Only for function F4, the p-value is equal to 1, indicating a tie between the two algorithms, which suggests that there is no significant difference in their performances for this specific function. Overall, the results demonstrate that the enhancements introduced in the IARO algorithm contribute to its superior optimization capabilities across various complex and multimodal problems. This statistical analysis validates the robustness and effectiveness of the proposed IARO algorithm in finding optimal solutions more consistently compared to the standard ARO.

Table 5
Wilcoxon signed-rank test results for IARO versus ARO.

Function	p-value	Winner
F1	1.6394E − 05	IARO
F2	3.4053E − 05	IARO
F3	0.0053197	IARO
F4	1	Tie
F5	0.013975	IARO
F6	0.00048969	IARO
F7	0.0077309	IARO
F8	0.0098421	IARO
F9	0.0098421	IARO
F10	0.00011499	IARO

5. DC motor speed regulation with FOPID controller

The DC motor is a widely used electromechanical device that converts electrical energy into mechanical energy through the interaction of a magnetic field and current-carrying conductors. It finds applications in various industries, such as robotics, automation, and electric vehicles, due to its simplicity, controllability, and high torque-to-inertia ratio. To enhance the performance and precision control of DC motors, advanced control strategies are employed. The fractional order proportional integral derivative (FOPID) controller is a modern control technique that utilizes fractional calculus to improve the stability and robustness of the control system. In this section, we explore the integration of a FOPID controller with an externally excited DC motor to achieve superior control characteristics.

The externally excited DC motor consists of a rotor (armature) and a stator (field winding) that are externally powered. When a voltage is applied to the field winding, a magnetic field is generated, interacting with the current in the armature winding, resulting in a torque that drives the motor's rotation. The speed of the motor can be controlled by varying the voltage applied to the armature winding. To develop a mathematical model of this system, the following assumptions and simplifications are considered: (1) the DC motor is modeled as a linear time-invariant system, (2) the mechanical load is modeled as a constant torque and (3) the motor is fed with a constant voltage input. Under these assumptions, the DC motor system can be represented through a system of differential equations governing the motor's speed and torque dynamics. Let ω denote the motor's angular velocity, and τ denote the motor's torque. The system's dynamics are captured by the following interconnected differential equations:

$$J(d\omega/dt) = \tau - B\omega \tag{17}$$

$$L_a(di/dt) = E_a - R_aI_a - K_e\omega \tag{18}$$

Here, J represents the motor's moment of inertia, B is the motor's damping coefficient, L_a stands for the motor's inductance, R_a denotes the motor's resistance, K_e characterizes the motor's back electromotive constant, and E_a represents the applied voltage to the motor. I_a signifies the motor's current, which is linked to torque through $\tau = K_t{\cdot}I_a$, where K_t denotes the motor's torque constant. These equations can be further simplified by assuming that the motor's internal dynamics operate significantly faster than the mechanical load's dynamics. Under this simplification, the motor's angular velocity ω can be considered constant and equal to the commanded speed. With this simplification, the above equations can be reduced to the following equation for the

current:

$$I_a = (E_a - K_e\omega)/R_a \tag{19}$$

The presented equation illustrates that the current in the motor is directly proportional to the applied voltage and inversely proportional to the motor's resistance. This relationship enables precise control of the motor's torque through voltage adjustments. Utilizing this model, it becomes possible to simulate the system's behavior under various control strategies and load conditions. Furthermore, control theory methodologies can be employed to devise a control system that regulates the motor's speed and torque via voltage modulation. The design of such a control system will be tailored to meet the specific demands and constraints of the given application. Considering the provided explanation, we can derive the open-loop transfer function for a DC motor as follows.

$$G_{motor}(s) = \frac{K_t}{(L_a s + R_a)(Js + B) + K_e K_t} \tag{20}$$

In order to comply with earlier reported studies [27,28,30,32], the DC motor parameters are selected as 0.4Ω for R_a, 2.7 H for L_a, $0.0004kg \bullet m^2$ for J, $0.0022N \bullet m \bullet s/rad$ for B, $0.015N \bullet m/A$ for K_t, and $0.05V \bullet s/rad$ for K_e. The DC motor can be controlled via a FOPID controller which is defined as follows [55].

$$C_{FOPID}(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \tag{21}$$

where K_p, K_i, and K_d are proportional, integral, and derivative gains, while λ and μ stand for fractional integral and derivative orders, respectively.

Fig. 2 presents the block diagram of an externally excited DC motor that is controlled with a FOPID controller. The implementations for this work were carried out on MATLAB/Simulink environment that is installed on a Windows computer with 12th Gen Intel i5-12400, 2.50 GHz processor and 16.00 GB RAM.

6. Constrained optimization problem and application of IARO algorithm

To find a feasible and efficient solution for the DC motor's speed regulation, the problem can be represented as a constraint minimization problem so that the optimization algorithm can be used. Regarding the delineation of the minimization problem, this study adopts the subsequent procedure for optimizing the FOPID controller. Firstly, the problem is represented as $X = [x_1, x_2, x_3, x_4, x_5] = [K_p, K_i, K_d, \lambda, \mu]$ and secondly the following *ITAE* cost function [56] is adopted for appropriate minimization via IARO algorithm.

$$ITAE = \int_0^{\infty} t \cdot |e(t)| \cdot dt \tag{22}$$

In here, $e(t)$ denotes the error signal. In this study, a penalty-based approach is used to handle constraints related to the design requirements such as settling time ($t_s \le 0.1s$), overshoot ($OS \le 1\%$), gain margin ($G_m \ge 100dB$), and phase margin ($P_m \ge 60^{\circ}$). In this regard, the optimization objective is considered based on the minimization of:

$$ITAE + \sum_{i=1}^{4} Penalty_i \tag{23}$$

where the penalty terms are defined as $Penalty_i = \rho[G_i(X)]^2 \cdot R$. Here, $i = 1, 2, 3, 4$ and R is defined as follows.:

$$R = \begin{cases} 0, \text{ if } G_1(X),\ G_2(X),\ G_3(X) \text{ and } G_4(X) \le 0 \\ 1, \text{ otherwise} \end{cases} \tag{24}$$

with constraints of $G_1(X) = t_s - 0.1 \le 0$; $G_2(X) = OS - 1 \le 0$; $G_3(X) = 100 - G_m \le 0$; $G_4(X) = 60 - P_m \le 0$ and penalty factor $\rho = 10^{20}$. The variable range of $0 < K_p, K_i, K_d < 20$ and $0.5 < \lambda, \mu < 1.5$ are also included for the design requirement. The penalty terms are incorporated into the ITAE objective function to penalize candidate solutions that violate these constraints, guiding the optimization process towards feasible solutions that meet all design criteria [57–59]. This method effectively balances minimizing the ITAE and adhering to the specified design requirements, converting the problem into a constrained optimization task. Fig. 3 illustrates the application of the IARO algorithm for the design of FOPID controlled DC motor.

7. Comparative simulation results and discussion

7.1. Statistical analysis and obtained FOPID parameters

For the DC motor system, the proposed IARO is initially assessed for its statistical performance of minimizing the ITAE cost function. For the minimization of this constraint engineering task, a population size of 30, total number of iterations of 50 are considered and the algorithm performs 30 runs. Besides, the default parameters of the incorporated algorithms are used in order to provide a fair comparison. The obtained cost function values with respect to all runs of IARO and ARO algorithms are provided in Fig. 4. As seen from the scatter plots of this figure, IARO algorithm consistently produces lower values compared to the ARO algorithm. It has a consistent minimization ability within a narrow band indicating its good performance characteristics.

The visualized performance of the IARO algorithm is numerically supported with the data presented in Table 6. The statistical metrics listed in this table show that the IARO algorithm is capable of reaching the best value of 8.3270E − 03 which is significantly better than the best value (8.9139E − 03) achieved via ARO algorithm. Similarly, the IARO has reached the minimum average value (8.4690E − 03) with lower standard deviation (1.0887E − 04) compared to the values achieved via ARO algorithm, demonstrating its superior capacity of minimizing the ITAE cost function for the design of FOPID controlled DC motor speed regulation system. The results of the Wilcoxon signed-rank test, shown in Table 6, indicate a statistically significant difference in performance

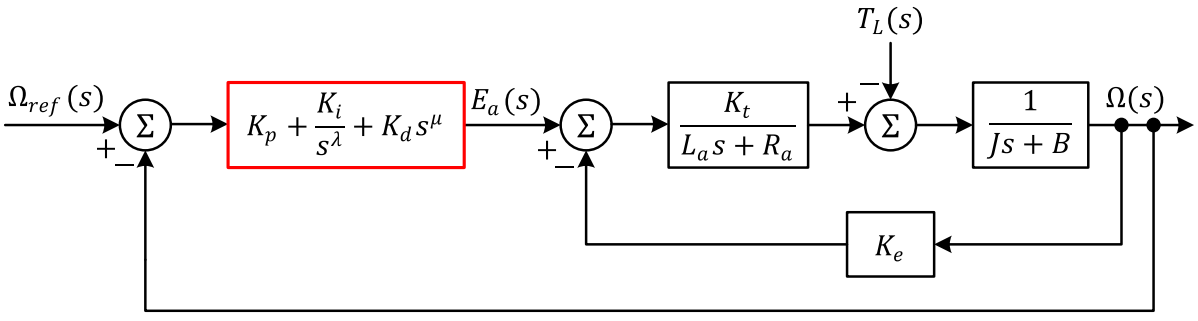


Fig. 2. Block diagram of FOPID controlled DC motor speed system.

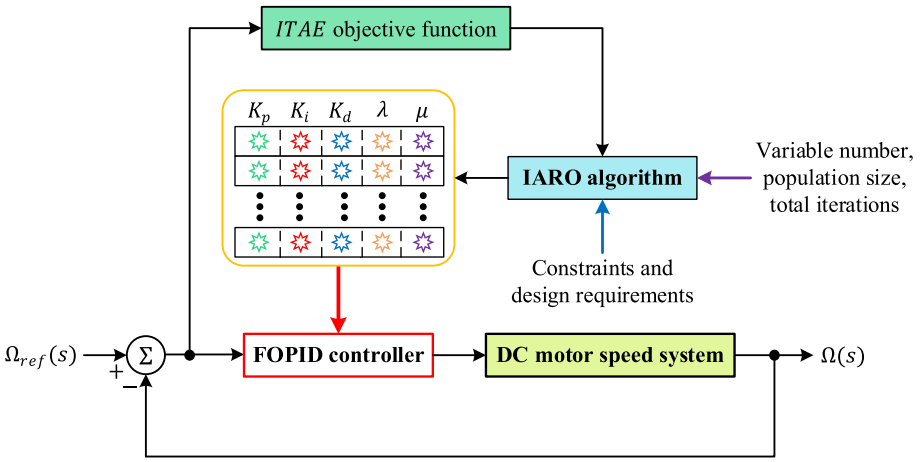


Fig. 3. Application of the IARO algorithm for FOPID controlled DC motor system.

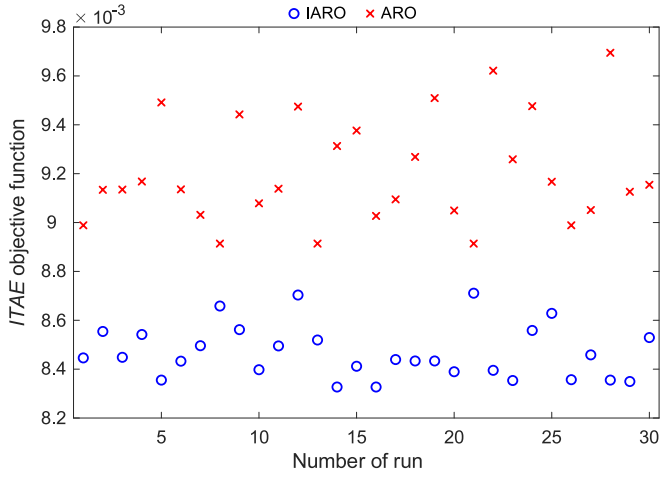


Fig. 4. The ITAE cost function values obtained from all 30 runs.

between the IARO and ARO algorithms for the FOPID tuning problem, with the IARO consistently demonstrating superior optimization capabilities across multiple runs. The p-value of 1.7344E − 06 confirms the effectiveness of the IARO in comparison to the ARO, validating the robustness and reliability of the proposed approach.

With the minimization of the ITAE cost function, the IARO algorithm achieved the FOPID controller parameters as $K_p = 16.3942$, $K_i = 12.5099$; $K_d = 9.5688$, $\lambda = 1.0652$, and $\mu = 0.9018$. Using those values together with the employed DC motor parameters and the information from Fig. 2, the following transfer function can be obtained for FOPID controlled DC motor designed with the IARO.

$$T_{IARO-FOPID}(s) = \frac{143.53s^{1.967} + 245.91s^{1.0652} + 187.65}{1.08s^{3.0652} + 6.1s^{2.0652} + 143.53s^{1.967} + 247.54s^{1.0652} + 187.65} \quad (25)$$

Similarly, with the minimization of the ITAE cost function, the original ARO algorithm achieved the FOPID controller parameters as $K_p = 14.7396$, $K_i = 4.5697$; $K_d = 8.7170$, $\lambda = 1.0398$, and $\mu = 0.8839$. Using those values together with the employed DC motor parameters and the information from Fig. 2, the following transfer function can be obtained for FOPID controlled DC motor designed with the original ARO.

$$T_{ARO-FOPID}(s) = \frac{130.76s^{1.9237} + 221.09s^{1.0398} + 68.545}{1.08s^{3.0398} + 6.1s^{2.0398} + 130.76s^{1.9237} + 222.72s^{1.0398} + 68.545} \quad (26)$$

The obtained transfer functions provided in Eqs. (23) and (24) can be used to perform the analyses presented in the subsequent sections for FOPID controlled DC motor system designed via IARO and ARO algorithms, respectively.

7.2. Convergence evolutions of cost function

Fig. 5 presents a visual representation of the comparative convergence profiles of the IARO and ARO algorithms, both applied to minimize the ITAE cost function for the DC motor system. By examining this graph, it becomes evident that the IARO algorithm exhibits a distinct advantage over the ARO algorithm in terms of convergence performance. The IARO algorithm consistently reaches the lowest ITAE cost function value in more iterations when compared to the ARO algorithm. This characteristic highlight the IARO algorithm's exceptional capability to effectively converge towards the optimal solution, making it an efficient and promising choice for optimizing the ITAE cost function in the context of the FOPID controller design for the DC motor system.

7.3. Stability analysis

In this section, we conduct a stability analysis of FOPID controlled DC motor systems designed using the IARO and ARO algorithms. Fig. 6 provides a speed comparison between the two FOPID controlled DC motor systems. As shown in the figure, the DC motor system designed with the IARO algorithm exhibits a smoother and more precise speed response, while the system designed with the ARO algorithm shows some oscillations and overshoot. This observation highlights the superior stability achieved by the IARO based FOPID controlled system, leading to a better-controlled speed profile during operation.

The Bode diagram, depicted in Fig. 7, presents the frequency

Table 6
Statistical results obtained from minimization of the ITAE cost function.

Algorithm	Best	Worst	Average	Standard deviation	p-value	Winner
IARO	8.3270E − 03	8.7109E − 03	8.4690E − 03	1.0887E − 04	1.7344E − 06	IARO
ARO	8.9139E − 03	9.6948E − 03	9.2046E − 03	2.1617E − 04		

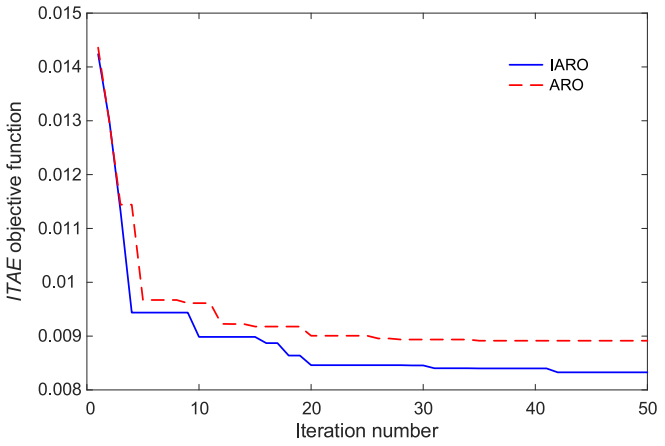


Fig. 5. Convergence curves of ITAE for IARO and ARO algorithms.

response of the open-loop FOPID controlled DC motor systems designed using the IARO and ARO algorithms. From the figure, we can observe that the IARO based FOPID controlled system exhibits a higher phase margin compared to the ARO based FOPID controlled system. This increase in phase margin indicates improved stability and robustness in the IARO based FOPID controlled system, providing a greater safety margin before reaching instability.

Table 7 provides a comprehensive comparison of various time and frequency domain metrics for both FOPID controlled DC motor systems. The IARO based FOPID controlled system showcases a percent overshoot (%OS) of 0%, while the ARO based FOPID controlled system has a higher percent overshoot of 0.2229%. Additionally, the IARO based FOPID controlled system demonstrates better performance in terms of rise time ($t_r = 0.0204s$), settling time ($t_s = 0.033s$), and peak time ($t_p = 0.0472s$) compared to the ARO based FOPID controlled system (rise time: 0.0236s, settling time: 0.037s, peak time: 0.0528s). Moreover, the IARO based FOPID controlled system exhibits a significantly high gain margin (G_m) of infinity and a larger phase margin (P_m) of 78.776° compared to the ARO based FOPID controlled system (gain margin: infinity, phase margin: 78.1326°). This implies that the IARO based FOPID controlled system has a better capacity to handle uncertainties and disturbances while maintaining stability. Another critical aspect of stability is the bandwidth (B_w) of the control system. The bandwidth indicates the range of frequencies at which the system can effectively respond to changes in the input signal. For the DC motor system, a wider

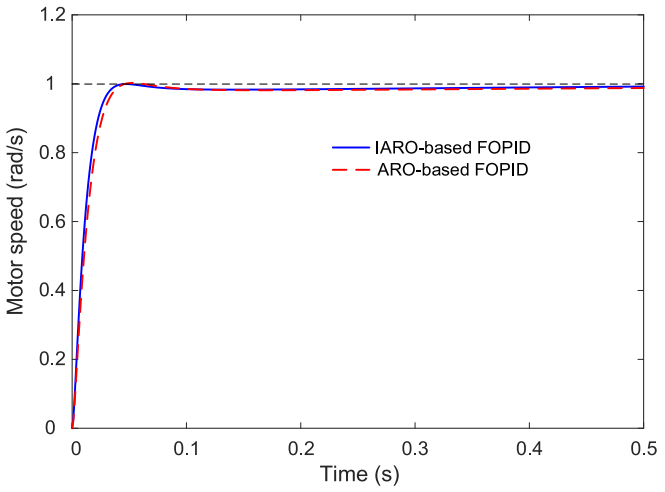


Fig. 6. Speed comparison of FOPID controlled DC motor systems designed with IARO and ARO algorithms.

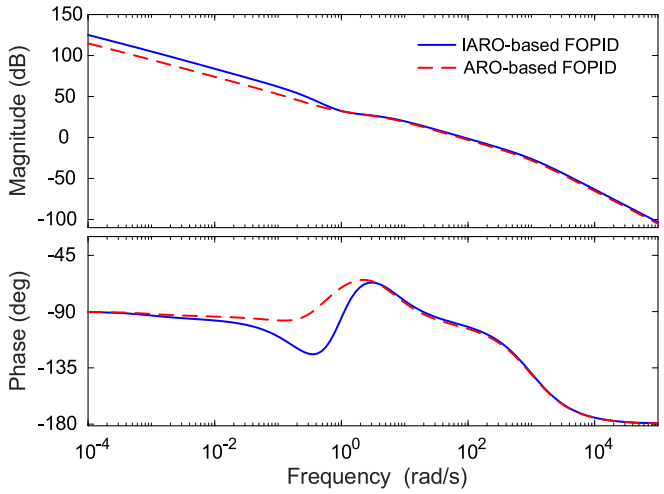


Fig. 7. Bode diagram of open-loop FOPID controlled DC motor systems designed with IARO and ARO algorithms.

bandwidth typically translates to faster and more accurate responses to control commands.

From Table 7, it is evident that the IARO based FOPID controlled system possesses a significantly larger bandwidth of 104.3071rad/s compared to the ARO based FOPID controlled system's bandwidth of 90.059rad/s. The higher bandwidth of the IARO based FOPID controlled system implies better responsiveness and superior dynamic performance, allowing the system to follow reference commands with increased precision and reduced tracking errors.

The comprehensive stability analysis, including time and frequency domain metrics, has consistently demonstrated that the FOPID controlled DC motor system designed with the IARO algorithm exhibits better stability characteristics compared to the one designed with the ARO algorithm. The IARO-FOPID controlled system showcases smoother speed response, higher phase margin, larger gain margin, and a significantly wider bandwidth. These findings further emphasize the efficacy of the proposed IARO algorithm in achieving a well-controlled and highly stable DC motor system, making it a compelling choice for applications where stability and dynamic performance are crucial.

7.4. Evaluation of time-domain based metrics against other reported algorithms-based FOPID controllers

In this section, we assess the performance of the proposed IARO algorithm-based FOPID controller by comparing it with other recently reported algorithms-based FOPID controllers, including enhanced hybrid stochastic fractal search FOPID controller (HSFS based FOPID [25]), manta ray foraging optimization based FOPID controller (MRFO based FOPID [26]), chaotic atom search optimization algorithm based FOPID controller (ChASO based FOPID [27]), and grey wolf optimizer based FOPID controller (GWO based FOPID [28]). Table 8 presents the reported controller parameters achieved via each method.

Using those parameters, the transfer functions presented in Eqs. (27), (28), (29) and (30) can be obtained for HSFS based FOPID [25], MRFO based FOPID [26], ChASO based FOPID [27] and GWO based FOPID [28], respectively. Using those transfer functions, one can perform the following comparative analyses.

$$T_{HSFS-FOPID}(s) = \frac{45.238s^{1.9666} + 288.45s^{0.9934} + 90.015}{1.08s^{2.9934} + 6.1s^{1.9934} + 45.238s^{1.9666} + 290.08s^{0.9934} + 90.015} \quad (27)$$

Table 7
Numerical results for time and frequency domain metrics.

System type	%OS	t_r(s)	t_s(s)	t_p(s)	G_m(dB)	P_m(°)	B_w(rad/s)
IARO-FOPID	0	0.0204	0.0330	0.0472	Inf	78.7760	104.3071
ARO-FOPID	0.2229	0.0236	0.0370	0.0528	Inf	78.1326	90.0590

Table 8
FOPID controller parameters obtained via different reported approaches.

Reported Works	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 [26]	19.0527	6.3585	5.3293	0.9466	0.9222
ChASO based FOPID [27]	19.7722	9.1117	8.1189	0.8401	0.9112
GWO based FOPID [28]	18.328	4.9418	3.2612	0.9998	0.9845

$$T_{MRFO-FOPID}(s) = \frac{79.939s^{1.8688} + 285.79s^{0.9466} + 95.378}{1.08s^{2.9466} + 6.1s^{1.9466} + 79.939s^{1.8688} + 287.42s^{0.9466} + 95.378} \quad (28)$$

$$T_{ChASO-FOPID}(s) = \frac{121.78s^{1.7513} + 296.58s^{0.8401} + 136.68}{1.08s^{2.8401} + 6.1s^{1.8401} + 121.78s^{1.7513} + 298.21s^{0.8401} + 136.68} \quad (29)$$

$$T_{GWO-FOPID}(s) = \frac{48.918s^{1.9843} + 274.92s^{0.9998} + 74.127}{1.08s^{2.9998} + 6.1s^{1.9998} + 48.918s^{1.9843} + 276.55s^{0.9998} + 74.127} \quad (30)$$

Fig. 8 displays a comparative analysis of speed responses for different FOPID controlled systems. The proposed IARO based FOPID controller showcases the most desirable behavior with smooth and precise speed responses. On the other hand, the speed responses of FOPID controllers designed using other reported algorithms show varying degrees of oscillations and overshoots, indicating less effective control.

Table 9 presents a detailed performance evaluation of the FOPID controllers based on various time-domain metrics, namely percent overshoot, rise time, settling time, and peak time. The time-domain metrics comparison clearly demonstrates the exceptional performance of the proposed IARO based FOPID controller. The IARO based FOPID system achieves zero percent overshoot, the shortest rise time of 0.0204s, and the quickest settling time of 0.033s compared to the other

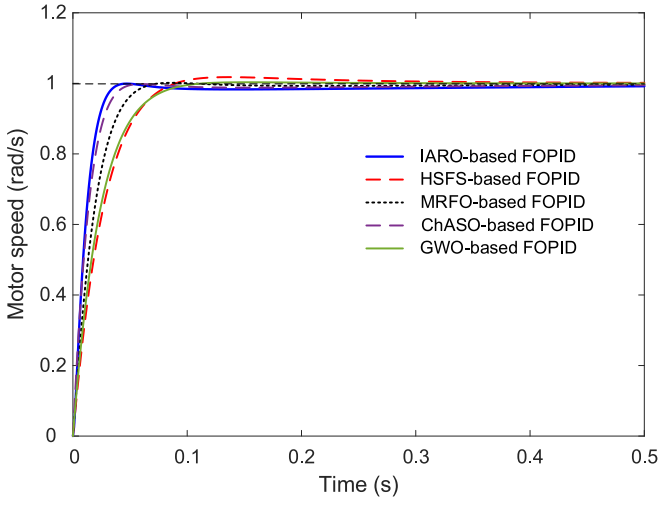


Fig. 8. Speed comparisons of reported recent FOPID controlled systems and proposed approach.

Table 9
Comparative performance evaluation of different algorithms based FOPID controllers in terms of time-domain based metrics.

System type	%OS	t_r(s)	t_s(s)	t_p(s)
IARO based FOPID (proposed)	**0**	**0.0204**	**0.0330**	**0.0472**
HSFS based FOPID [25]	1.7674	0.0505	0.0766	0.1350
MRFO based FOPID [26]	0.1546	0.0355	0.0562	0.0868
ChASO based FOPID [27]	0	0.0253	0.0405	0.0598
GWO based FOPID [28]	0.3145	0.0488	0.0814	0.1528

reported algorithms. It also exhibits a reasonably low peak time of 0.0472s. In contrast, the other FOPID controllers display varying degrees of overshoot, longer rise and settling times, and higher peak times. Notably, the FOPID controllers designed with the HSFS, and GWO algorithms exhibit relatively higher percent overshoot, indicating less stability and potentially undesired oscillations in the system. The evaluation of time-domain based metrics against other reported algorithms-based FOPID controllers unequivocally establishes the superiority of the proposed IARO based FOPID controller. Its remarkable ability to minimize overshoot, achieve faster rise and settling times, and maintain a reasonable peak time sets it apart as a highly effective control solution for the DC motor system. These results highlight the efficacy and practicality of the IARO algorithm in designing stable and well-performing FOPID controllers, making it a promising choice for DC motor speed regulation systems.

7.5. Evaluation of time-domain based metrics against other reported algorithms-based PID controllers

In this section, we assess the performance of the proposed IARO algorithm-based FOPID controller by comparing it with other recently reported algorithms-based PID controllers, including enhanced arithmetic optimization algorithm based PID controller (AOA-HHO based PID [29]), opposition based Henry gas solubility optimization based PID controller (OBL/HGSO based PID [32]), improved model of marine predator algorithm based PID controller (MMPA based PID [30]), hybrid Lévy flight distribution and Nelder–Mead algorithm based PID controller (LFDNM based PID [31]). Table 10 presents the reported controller parameters achieved via each method.

Using those parameters, the transfer functions presented in Eqs. (31), (32), (33) and (34) can be obtained for AOA-HHO based PID [29], OBL/HGSO based PID [32], MMPA based PID [30]), LFDNM based PID [31]. Using those transfer functions, one can perform the following comparative analyses.

$$T_{AOA-HHO-PID}(s) = \frac{26.43s^2 + 216.5s + 2.454}{1.08s^3 + 32.53s^2 + 218.2s + 2.454} \quad (31)$$

$$T_{OBL/HGS-PID}(s) = \frac{42.77s^2 + 254s + 14.26}{1.08s^3 + 48.87s^2 + 255.6s + 14.26} \quad (32)$$

Table 10
PID controller parameters obtained via different reported approaches.

Reported Works	K_p	K_i	K_d
AOA-HHO based PID [29]	14.435	0.1636	1.7620
OBL/HGSO based PID [32]	16.9327	0.9508	2.8512
MMPA based PID [30]	20	0.7448	1.7395
LFDNM based PID [31]	19.5593	5.1566	3.4097

$$T_{MMPA-PID}(s) = \frac{26.09s^2 + 300s + 11.17}{1.08s^3 + 32.19s^2 + 301.6s + 11.17} \quad (33)$$

$$T_{LFDNM-PID}(s) = \frac{51.15s^2 + 293.4s + 77.35}{1.08s^3 + 57.25s^2 + 295s + 77.35} \quad (34)$$

Fig. 9 displays a comparative analysis of speed responses for different PID controlled systems plotted against IARO based FOPID controller. The proposed IARO based FOPID controller showcases the most desirable behavior with smooth and precise speed responses. On the other hand, the speed responses of PID controllers designed using other reported algorithms show varying degrees of oscillations and overshoots, indicating less effective control.

Table 11 provides a detailed performance assessment of the PID controllers based on various time-domain metrics. The comparative analysis based on time-domain metrics reveals notable differences in the performance of the PID controllers. The proposed IARO based FOPID controller stands out with zero percent overshoot, indicating excellent control and minimal oscillations in the system. It also achieves the shortest rise time of 0.0204s and settling time of 0.033s, demonstrating its rapid response to reach the desired output. Among the reported PID controllers, the OBL/HGSO based PID controller also achieves zero percent overshoot, but it exhibits slightly longer rise and settling times compared to the IARO based FOPID controller. The LFDNM based PID controller shares similar characteristics with zero percent overshoot, but it has a marginally longer rise time and settling time compared to the IARO based FOPID controller. On the other hand, the AOA-HHO based PID controller exhibits the highest percent overshoot of 2.8376%, indicating less stable and more oscillatory behavior in the system. The MMPA based PID controller displays the highest percent overshoot of 7.0059%, indicating the least stable performance among all the controllers evaluated. The evaluation of time-domain based metrics against other reported algorithms-based PID controllers demonstrates the exceptional performance of the proposed IARO based FOPID controller. Its ability to achieve zero percent overshoot, along with the shortest rise and settling times, distinguishes it as a highly effective control solution. Compared to the other PID controllers, the IARO based FOPID controller offers superior stability and precise control over the system response. These results further underscore the efficacy and practicality of the IARO algorithm in designing stable and high-performing controllers, positioning it as a promising choice for DC motor speed regulation systems.

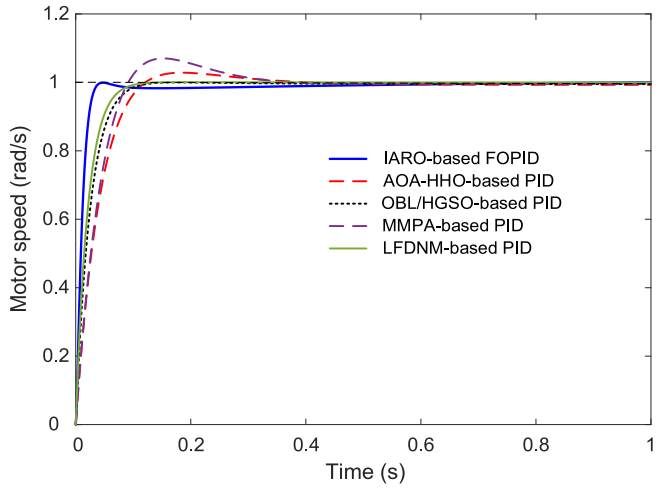


Fig. 9. Speed comparisons of reported recent PID controlled systems and proposed approach.

Table 11
Comparative performance evaluation of different algorithms based PID controllers in terms of time-domain based metrics.

System type	%OS	t_r(s)	t_s(s)	t_p(s)
IARO based FOPID (proposed)	**0**	**0.0204**	**0.0330**	**0.0472**
AOA-HHO based PID [29]	2.8376	0.0743	0.2508	0.1832
OBL/HGSO based PID [32]	0	0.0546	0.0946	0.1749
MMPA based PID [30]	7.0059	0.0635	0.2793	0.1516
LFDNM based PID [31]	0	0.0462	0.0813	0.1452

7.6. Hardware implementation

As part of the validation process, an experimental phase was conducted in this investigation. The experimental setup utilized an existing physical system [26], illustrated in Fig. 10(a). The system's hardware comprises a DC motor with a driver and power supply, alongside a data acquisition card (National Instruments) and a magnetic encoder. The controller was implemented within the Simulink environment, following the block diagram outlined in Fig. 10(b). To specify the speed command in revolutions per minute (rpm), the Simulink environment was employed. A corresponding pulse-width-modulation (PWM) signal was generated at the analog output of the data acquisition device by converting the commanded speed value. This facilitated the creation of the corresponding voltage level for speed control with the motor driver's assistance. The magnetic encoder read the motor's output speed in terms of logic signals, and the related data was transmitted back to the Simulink environment through the digital input of the data acquisition card. A second conversion was performed to obtain the speed in terms of rpm. The commanded and acquired speeds were then compared to provide input to the FOPID controller, enabling the regulation of the speed as required. The FOPID parameters obtained through various algorithms were applied to analyze the transient response behavior of the respective systems. Experimental results were obtained for a commanded speed of 35 rpm, as demonstrated in Fig. 11. The comparative analysis of experimental and simulation results in the respective figure is well-matched, confirming the superior performance of the proposed FOPID controller based on IARO for DC motor speed regulation.

8. Reference tracking

In this section, we evaluate the tracking capability of the proposed IARO-based FOPID controller in response to multi-step reference inputs. The results, illustrated in Fig. 12, show the input signal tracking ability of the IARO-based FOPID controller. As depicted, the controller effectively tracks the reference inputs with minimal delay and no noticeable overshoot, demonstrating a high level of precision and responsiveness. The controller smoothly follows the reference changes, confirming its ability to maintain accurate speed regulation across varying input conditions. This performance is crucial in practical applications where rapid adaptation to different operational commands is required, validating the robustness and adaptability of the IARO-based FOPID controller in dynamic environments.

8.1. External disturbance rejection

To assess the robustness of the proposed controller under disturbance conditions, we introduce external disturbances into the system and evaluate its performance. The disturbance rejection ability of the IARO-based FOPID controller is presented in Fig. 13. The results illustrate that the controller can quickly and effectively mitigate the impact of external disturbances, maintaining the desired speed with minimal deviation. This demonstrates the enhanced disturbance rejection capabilities of the IARO-based FOPID controller, which is critical for applications where the system may be exposed to unpredictable environmental changes or load variations. The controller's ability to reject disturbances while

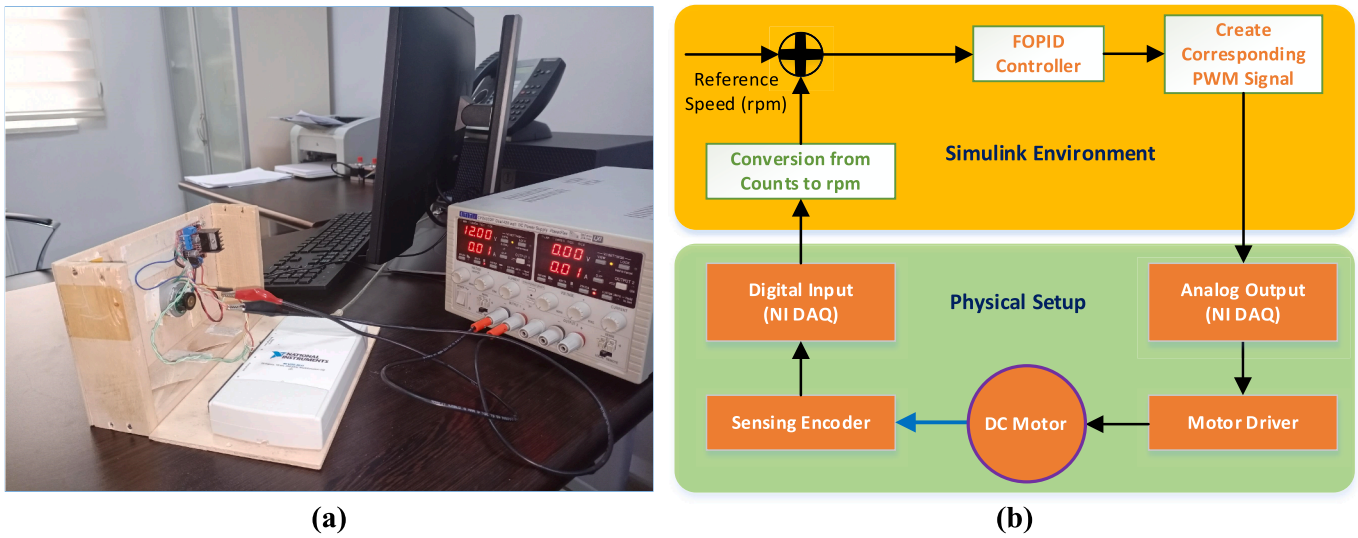


Fig. 10. The hardware implementation (a) and the block diagram showing the experimental procedure (b).

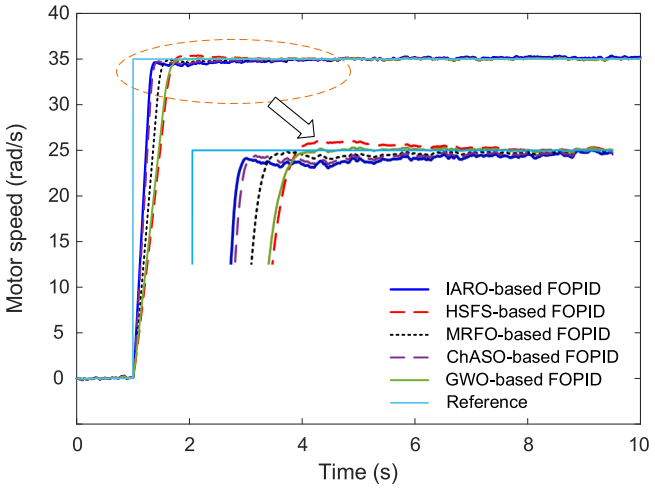


Fig. 11. The experimental results obtained from hardware implementation.

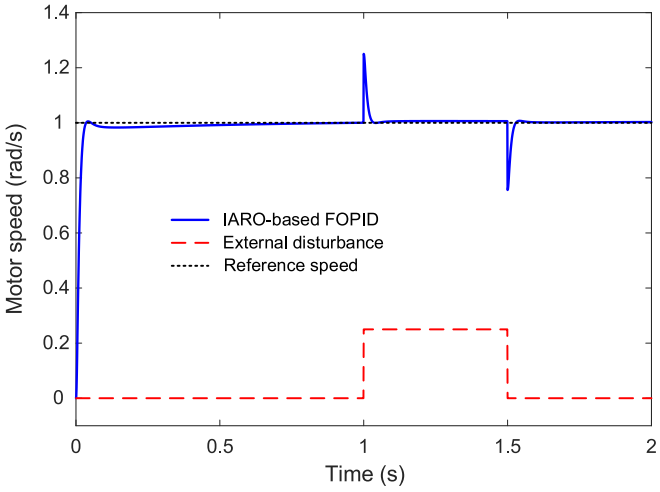


Fig. 13. Disturbance rejection ability of IARO-based FOPID controller.

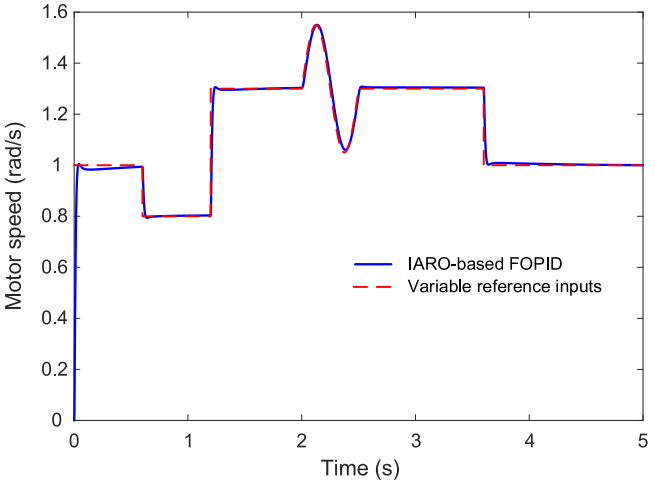


Fig. 12. Input signal tracking ability of IARO-based FOPID controller.

maintaining stability underlines its superiority in ensuring reliable performance in real-world conditions.

8.2. Design of IARO based PID controller

To provide a fair comparison, the IARO algorithm was also applied to tune a traditional PID controller, and its performance was evaluated against other recent PID-controlled systems. The optimized PID parameters achieved using the IARO are $K_p = 19.9913$, $K_i = 10.4694$ and $K_d = 3.6278.$, resulting in the transfer function:

$$T_{IARO-PID}(s) = \frac{54.42s^2 + 299.9s + 157}{1.08s^3 + 60.52s^2 + 301.5s + 157} \quad (35)$$

The speed response comparisons, shown in Fig. 14, indicate that the IARO-based PID controller outperforms other recently reported PID-controlled systems, displaying a smoother speed profile with less overshoot and faster response times. The comparative performance evaluation presented in Table 12 further confirms that the IARO-based PID controller achieves a 0 % overshoot, a rise time of 0.0439 s, a settling time of 0.0786 s, and a peak time of 0.1448 s, outperforming other methods in terms of key time-domain metrics. This result demonstrates that the proposed IARO method is not only effective for FOPID

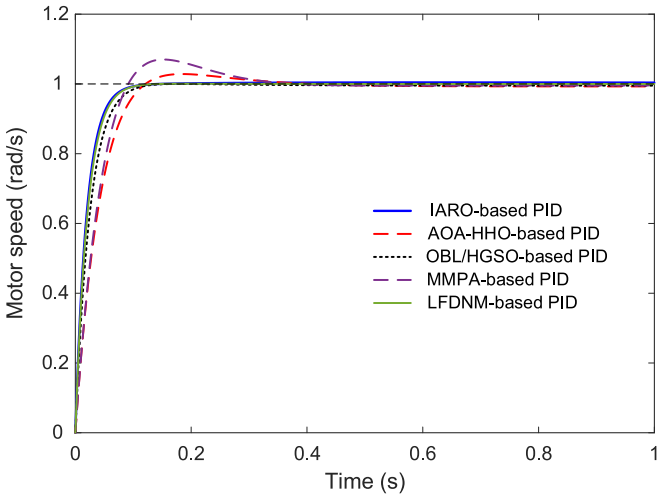


Fig. 14. Speed comparisons of reported recent PID controlled systems and IARO based PID controlled system.

Table 12
Comparative performance evaluation of IARO and different algorithms based PID controllers in terms of time-domain based metrics.

System type	%*OS*	t_r(s)	t_s(s)	t_p(s)
IARO based PID	**0**	**0.0439**	**0.0786**	**0.1448**
AOA-HHO based PID [29]	2.8376	0.0743	0.2508	0.1832
OBL/HGSO based PID [32]	0	0.0546	0.0946	0.1749
MMPA based PID [30]	7.0059	0.0635	0.2793	0.1516
LFDNM based PID [31]	**0**	0.0462	0.0813	0.1452

controllers but also enhances the performance of traditional PID controllers, providing a versatile and robust solution for various control applications.

9. Discussion

In this section, we analytically discuss the reasons for the improvements observed with the proposed IARO in optimizing the FOPID controller for DC motor speed regulation. The significant performance gains demonstrated by the IARO over the standard ARO and other competitive algorithms can be attributed to the specific enhancements introduced in its design: the ALS mechanism and the EPL strategy. The significant improvements observed with the proposed IARO-based FOPID controller can be attributed to several key factors, as reflected in the achieved simulation and experimental results.

The ALS mechanism improves the exploitation capability of the IARO by directing the search towards promising regions in the solution space. It dynamically adjusts the local search process based on the fitness landscape, preventing premature convergence to local optima. This enhancement is particularly effective in tuning the FOPID controller parameters, as it ensures a more thorough exploration of the solution space around high-quality solutions. Consequently, the IARO achieves a more precise and stable controller design, as evidenced by the reduced overshoot, faster settling time, and lower ITAE values.

The EPL strategy further enhances the IARO's global search ability by maintaining diversity in the population and avoiding stagnation around suboptimal solutions. It does this by perturbing the solutions based on the experiences of the best solutions found so far, thus balancing exploration and exploitation. This is particularly beneficial for complex and multimodal optimization problems like FOPID tuning, where the control landscape is highly nonlinear. By leveraging EPL, the IARO effectively navigates the solution space, leading to consistently superior performance metrics such as wider bandwidth, higher gain margins, and smoother speed responses.

The integration of ALS and EPL strategies into the IARO contributes substantially to its superior performance. The ALS mechanism enhances the optimizer's exploitation capability by dynamically guiding the search toward promising regions while preventing premature convergence to local optima. This is evident in the lower ITAE cost function values achieved by the IARO algorithm (best value of 8.3270E − 03) compared to the original ARO algorithm (best value of 8.9139E − 03). Similarly, the EPL strategy maintains diversity within the population, improving global search capabilities and allowing the optimizer to escape local minima. The convergence profile presented in Fig. 5 shows that the IARO consistently reaches optimal solutions more quickly than the ARO, underscoring its effectiveness in balancing exploration and exploitation.

The superiority of the IARO algorithm was validated through extensive tests on the CEC2020 benchmark functions, which represent a wide range of complex optimization challenges. The IARO demonstrated its ability to find near-optimal or optimal solutions consistently, with lower standard deviations and better stability compared to other algorithms. These findings suggest that the IARO is well-suited for solving real-world optimization problems, such as tuning FOPID controllers, where both robustness and precision are crucial.

The IARO-based FOPID controller exhibits markedly better stability and dynamic response characteristics than the ARO-based FOPID controller. As shown in Table 7, the IARO-based system achieves a 0 % overshoot, faster rise time (0.0204 s), and shorter settling time (0.033 s) compared to the ARO-based system (0.2229 % overshoot, 0.0236 s rise time, and 0.037 s settling time). Additionally, the IARO-based system demonstrates a higher phase margin (78.776°) and a significantly wider bandwidth (104.3071 rad/s) than the ARO-based system (78.1326° phase margin and 90.059 rad/s bandwidth), indicating enhanced robustness and stability in handling uncertainties and disturbances.

The practical applicability of the IARO-based FOPID controller was validated through hardware implementation. The experimental results, depicted in Fig. 11, align well with the simulation outcomes, confirming the proposed controller's ability to maintain precise speed regulation under real-world conditions. The smoother and more stable speed response achieved with the IARO-based FOPID controller compared to other methods reinforces its potential for deployment in diverse engineering applications where robust and precise motor control is essential.

A comprehensive evaluation against other recently reported algorithms-based FOPID and PID controllers further highlights the exceptional performance of the IARO-based FOPID controller. As shown in Tables 9 and 11, the IARO-based controller achieves the lowest percent overshoot, fastest rise and settling times, and the lowest peak time among the evaluated methods. These results demonstrate that the IARO-based FOPID controller not only outperforms the standard ARO-based controller but also surpasses other state-of-the-art algorithms, such as the HSFS, MRFO, ChASO and GWO algorithms, in terms of both time-domain and frequency-domain performance metrics.

Furthermore, the IARO-based FOPID controller demonstrated excellent reference tracking capability, effectively following multi-step reference inputs with minimal delay and no noticeable overshoot. The controller's performance in rejecting external disturbances was also superior, maintaining the desired speed with minimal deviation, which is essential for applications exposed to unpredictable environmental changes or load variations. The IARO algorithm's adaptability was further validated by its application to a traditional PID controller, where it outperformed recent PID-controlled systems in terms of overshoot, rise time, settling time, and peak time, highlighting its versatility across different control structures. In conclusion, the improvements achieved by the proposed IARO-based FOPID controller stem from the advanced search strategies incorporated into the IARO algorithm. The enhanced balance between exploration and exploitation, coupled with the superior stability and dynamic response characteristics, make the IARO-based method a highly effective solution for optimizing FOPID

controllers in DC motor speed regulation applications. The results underscore the potential of the IARO algorithm as a robust and practical optimization technique, offering significant advantages over existing methods in both simulated and real-world environments.

10. Conclusion and future work

In this study, we explored the optimization of FOPID controllers for DC motor speed regulation using the IARO algorithm. To make the most of the advantages offered by FOPID controllers, an efficient tuning method was necessary. We proposed the IARO algorithm, an enhanced version of the ARO algorithm, which incorporates an ALS mechanism and an EPL strategy. The IARO algorithm addresses the limitations of ARO, striking a better balance between exploration and exploitation. We validated the effectiveness of the IARO algorithm by comparing it to the original version of ARO and other competitive algorithms using the challenging CEC2020 benchmark functions. The results demonstrated the superiority of the IARO algorithm in terms of solution stability and consistency over its competitors. Subsequently, we applied the IARO algorithm to optimize the FOPID controller for DC motor speed regulation, formulating the problem as a constraint minimization task. Through comprehensive simulations, we evaluated the performance of the IARO-based FOPID controller and compared it with various other algorithms-based FOPID controllers and PID controllers. The simulation results confirmed that the IARO algorithm outperformed the other methods in terms of cost function values and convergence profile, displaying enhanced stability, smoother speed response, larger gain margin, and wider bandwidth. We also conducted hardware implementation, further validating the practical applicability of our proposed method. Moreover, the IARO-based FOPID controller proved to be highly effective in practical scenarios, showing excellent performance in reference tracking, where it maintained high precision with minimal overshoot and delay, and in disturbance rejection, where it quickly mitigated the impact of external disturbances. These results confirm the controller's robustness and adaptability in dynamic environments. Additionally, the proposed IARO was implemented on a traditional PID controller to provide a fair comparison with other recent PID-based control strategies. The IARO-based PID controller showed significant improvements over existing methods, achieving better time-domain performance metrics, such as lower overshoot, faster rise time, shorter settling time, and reduced peak time. This demonstrates the algorithm's versatility and effectiveness across different control paradigms.

Looking ahead, several avenues for further exploration and improvement in the field of DC motor speed regulation with FOPID controllers and IARO algorithm emerge. Firstly, the extension of this research to explore the applicability of the IARO algorithm in other control engineering domains beyond DC motor systems holds promise. Investigating the performance of the IARO algorithm in other motor types (e.g., AC motor systems) and different engineering applications will provide valuable insights into its versatility and adaptability. Furthermore, enhancing the IARO algorithm to handle even more complex and dynamic systems with multiple constraints can lead to more robust control solutions. This could involve incorporating advanced learning and adaptive mechanisms to ensure the algorithm's effectiveness in challenging scenarios. Lastly, combining the IARO algorithm with other optimization techniques or control strategies may offer synergistic benefits. Exploring hybrid algorithms or control schemes can potentially boost the algorithm's performance and make it even more effective in various real-world applications.

CRediT authorship contribution statement

All authors contributed equally.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data are available within the paper.

References

[1] Xie W, Wang JS, Wang HB. 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. https://doi.org/10.1155/2019/2671792.
[2] Ibrahim EK, Issa AH, Gitaffa SA. Optimization and performance analysis of fractional order PID controller for DC motor speed control. Journal Européen Des Systèmes Automatisés 2022;55:723–39. https://doi.org/10.18280/jesa.550605.
[3] Eker E, Kayri M, Ekinci S, Izci D. A new fusion of ASO with SA algorithm and its applications to MLP training and DC motor speed control. Arab J Sci Eng 2021;46:3889–911. https://doi.org/10.1007/s13369-020-05228-5.
[4] Vanchinathan K, Selvaganesan N. Adaptive fractional order PID controller tuning for brushless DC motor using artificial bee colony algorithm. Results in Control and Optimization 2021;4:100032. https://doi.org/10.1016/j.rico.2021.100032.
[5] Alkrwy A, Hussein AA, Atyia TH, Khamees M. Adaptive tuning of PID controller using crow search algorithm for DC motor. IOP Conf Ser Mater Sci Eng 2021;1076:012001. https://doi.org/10.1088/1757-899X/1076/1/012001.
[6] Mohammed Eltoum A, Hussein A, Abido MA. Hybrid fuzzy fractional-order PID-based speed control for brushless DC motor. Arab J Sci Eng 2021;46:9423–35. https://doi.org/10.1007/s13369-020-05262-3.
[7] Farahani G, Rahmani K. Speed control of a separately excited DC motor using new proposed fuzzy neural algorithm based on FOPID controller. Journal of Control, Automation and Electrical Systems 2019;30:728–40. https://doi.org/10.1007/s40313-019-00485-8.
[8] Pongfai J, Su X, Zhang H, Assawinchaichote W. A novel optimal PID controller autotuning design based on the SLP algorithm. Expert Syst 2020;37:e12489. https://doi.org/10.1111/exsy.12489.
[9] Puangdownreong D. Fractional order PID controller design for DC motor speed control system via flower pollination algorithm. Transactions on Electrical Engineering, Electronics, and Communications 2019;17:14–23. https://doi.org/10.37936/ecti-eec.2019171.215368.
[10] Izci D, Ekinci S. Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller. Electrica 2021;21:151–9. https://doi.org/10.5152/electrica.2021.20077.
[11] Izci D, Ekinci S, Hussien AG. Effective PID controller design using a novel hybrid algorithm for high order systems. PLoS One 2023;18:e0286060. https://doi.org/10.1371/journal.pone.0286060.
[12] Izci D, Ekinci S. A novel improved version of hunger games search algorithm for function optimization and efficient controller design of buck converter system. E-Prime - Advances in Electrical Engineering, Electronics and Energy 2022;2:100039. https://doi.org/10.1016/j.prime.2022.100039.
[13] Kumarasamy V, KarumanchettyThottam Ramasamy V, Chandrasekaran G, Chinnaraj G, Sivalingam P, Kumar NS. A review of integer order PID and fractional order PID controllers using optimization techniques for speed control of brushless DC motor drive. International Journal of System Assurance Engineering and Management 2023;14:1139–50. https://doi.org/10.1007/s13198-023-01952-x.
[14] Munagala VK, Jatoth RK. A novel approach for controlling DC motor speed using NARXnet based FOPID controller. Evol Syst 2023;14:101–16. https://doi.org/10.1007/s12530-022-09437-1.
[15] Potnuru D, Alice Mary K, Sai BC. Experimental implementation of flower pollination algorithm for speed controller of a BLDC motor. Ain Shams Eng J 2019;10:287–95. https://doi.org/10.1016/j.asej.2018.07.005.
[16] Mohd Tumari MZ, Ahmad MA, Suid MH, Hao MR. An improved marine predators algorithm-tuned fractional-order PID controller for automatic voltage regulator system. Fractal and Fractional 2023;7:561. https://doi.org/10.3390/fractalfract7070561.
[17] Shalaby R, El-Hossainy M, Abo-Zalam B, Mahmoud TA. Optimal fractional-order PID controller based on fractional-order actor-critic algorithm. Neural Comput Appl 2023;35:2347–80. https://doi.org/10.1007/s00521-022-07710-7.
[18] Shaheen MAM, Hasanien HM, Mekhamer SF, Talaat HEA. Walrus optimizer-based optimal fractional order PID control for performance enhancement of offshore wind farms. Sci Rep 2024;14:17636. https://doi.org/10.1038/s41598-024-67581-x.
[19] Mazumdar D, Biswas PK, Sain C, Ahmad F, Al-Fagih L. A comprehensive analysis of the optimal GWO based FOPID MPPT controller for grid-tied photovoltaics system under atmospheric uncertainty. Energy Rep 2024;12:1921–35. https://doi.org/10.1016/j.egyr.2024.08.013.
[20] Rajendran A, Karthikeyan M, Saravanakumar G. Implementation of FOPID controller with modified harmony search optimization for precise modelling and auto-tuning of nonlinear systems. Automatika 2024;65:881–93. https://doi.org/10.1080/00051144.2024.2307227.

[21] Kottayathu Rajagopalan AG, Mahapatra S, Mahapatro SR. Advanced tree-seed optimization based fractional-order <scp>PID</scp> controller design for simplified decoupled industrial tank systems. Int J Numer Model Electron Networks Devices Fields 2024;37. https://doi.org/10.1002/jnm.3228.
[22] Gupta DK, Dei G, Soni AK, Jha AV, Appasani B, Bizon N, et al. Fractional order PID controller for load frequency control in a deregulated hybrid power system using Aquila optimization. Results in Engineering 2024;23:102442. https://doi.org/10.1016/j.rineng.2024.102442.
[23] Ersali C, Hekimoğlu B. A novel opposition-based hybrid cooperation search algorithm with Nelder-Mead for tuning of FOPID-controlled buck converter. Trans Inst Meas Control 2024;46:1924–42. https://doi.org/10.1177/01423312231214593.
[24] Izci D, Ekinci S, Zeynelgil HL, Hedley J. Fractional order PID design based on novel improved slime mould algorithm. Electr Power Compon Syst 2021;49:901–18. https://doi.org/10.1080/15325008.2022.2049650.
[25] Saini R, Parmar G, Gupta R. An enhanced hybrid stochastic fractal search FOPID for speed control of DC motor. Fractional Order Systems and Applications in Engineering, Elsevier 2023:51–67. https://doi.org/10.1016/B978-0-32-390953-2.00011-6.
[26] 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;46:1395–409. https://doi.org/10.1007/s13369-020-05050-z.
[27] Hekimoglu B. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. IEEE Access 2019;7:38100–14. https://doi.org/10.1109/ACCESS.2019.2905961.
[28] 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. https://doi.org/10.1007/s00542-018-3920-4.
[29] Issa M. Enhanced arithmetic optimization algorithm for parameter estimation of PID controller. Arab J Sci Eng 2023;48:2191–205. https://doi.org/10.1007/s13369-022-07136-2.
[30] Ramezani M, Bahmanyar D, Razmjooy N. A New Improved Model of Marine Predator Algorithm for Optimization Problems. Arab J Sci Eng 2021;46:8803–26. https://doi.org/10.1007/s13369-021-05688-3.
[31] 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;43:3195–211. https://doi.org/10.1177/01423312211019633.
[32] Ekinci S, Hekimoğlu B, Izci D. Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. Engineering Science and Technology, an International Journal 2021;24:331–42. https://doi.org/10.1016/j.jestch.2020.08.011.
[33] Ekinci S, Izci D, Yilmaz M. Efficient speed control for DC motors using novel gazelle simplex optimizer. IEEE Access 2023;11:105830. https://doi.org/10.1109/ACCESS.2023.3319596.
[34] Khalilpour M, Razmjooy NHH, Moallem P. Optimal control of DC motor using invasive weed optimization (IWO) algorithm. Majlesi Conference on Electrical Engineering, Isfahan 2011:1–6.
[35] Widya Suseno E, Ma'arif A. Tuning of PID Controller Parameters with Genetic Algorithm Method on DC Motor. International Journal of Robotics and Control Systems 2021;1:41–53. https://doi.org/10.31763/ijrcs.v1i1.249.
[36] Shamseldin MA. Optimal Coronavirus Optimization Algorithm Based PID Controller for High Performance Brushless DC Motor. Algorithms 2021;14:193. https://doi.org/10.3390/a14070193.
[37] Saini R, Parmar G, Gupta R, Sikander A. An Enhanced Tuning of PID Controller via Hybrid Stochastic Fractal Search Algorithm for Control of DC Motor, 2022, p. 185–94. DOI: 10.1007/978-981-16-7274-3_16.
[38] Munagala VK, Jatoth RK. Design of Fractional-Order PID/PID Controller for Speed Control of DC Motor Using Harris Hawks Optimization. In: Kumar R, Singh VP, Mathur A, editors., Singapore: Springer Singapore; 2021, p. 103–13. DOI: 10.1007/978-981-15-8045-1_11.
[39] Ekinci S, Hekimoğlu B, Demirören A, Eker E. Speed Control of DC Motor Using Improved Sine Cosine Algorithm Based PID Controller. In: 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT); 2019. p. 1–7. https://doi.org/10.1109/ISMSIT.2019.8932907.
[40] Razmjooy N, Vahedi Z, Estrela VV, Padilha R, Monteiro ACB. Speed Control of a DC Motor Using PID Controller Based on Improved Whale Optimization Algorithm 2021:153–67. https://doi.org/10.1007/978-3-030-56689-0_8.
[41] Qi Z, Shi Q, Zhang H. Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-Based DC motor subject to stochastic delays. IEEE Trans Ind Electron 2020;67:5637–46. https://doi.org/10.1109/TIE.2019.2934030.
[42] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:67–82. https://doi.org/10.1109/4235.585893.
[43] Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 2022;114:105082. https://doi.org/10.1016/j.engappai.2022.105082.
[44] Mohd Tumari MZ, Ahmad MA, Suid MH, Ghazali MR, Tokhi MO. An improved marine predators algorithm tuned data-driven multiple-node hormone regulation neuroendocrine-PID controller for multi-input–multi-output gantry crane system. Journal of Low Frequency Noise, Vibration and Active Control 2023;42:1666–98. https://doi.org/10.1177/14613484231183938.
[45] Suid MH, Ahmad MA. Optimal tuning of sigmoid PID controller using nonlinear sine cosine algorithm for the automatic voltage regulator system. ISA Trans 2022;128:265–86. https://doi.org/10.1016/j.isatra.2021.11.037.
[46] Sivanandhan A, Thriveni G. Optimal design of controller for automatic voltage regulator performance enhancement: a survey. Electr Eng 2024. https://doi.org/10.1007/s00202-023-02196-5.
[47] Siva Krishna P, Rao GK, Pv.. Fractional-order PID controller for blood pressure regulation using genetic algorithm. Biomed Signal Process Control 2024;88:105564. https://doi.org/10.1016/j.bspc.2023.105564.
[48] Izci D, Rizk-Allah RM, Snášel V, Ekinci S, Hashim FA, Abualigah L. A novel control scheme for automatic voltage regulator using novel modified artificial rabbits optimizer. E-Prime - Advances in Electrical Engineering, Electronics and Energy 2023;6:100325. https://doi.org/10.1016/j.prime.2023.100325.
[49] Agushaka JO, Ezugwu AE, Abualigah L. Dwarf Mongoose Optimization Algorithm. Comput Methods Appl Mech Eng 2022;391:114570. https://doi.org/10.1016/j.cma.2022.114570.
[50] Eberhart RC, Shi Y. Guest Editorial Special Issue on Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation 2004;8:201–3. DOI: 10.1109/TEVC.2004.830335.
[51] Price K V., Storn RM, Lampinen JA. Differential Evolution. Berlin/Heidelberg: Springer-Verlag; 2005. DOI: 10.1007/3-540-31306-0.
[52] Simon D. Biogeography-Based Optimization. IEEE Trans Evol Comput 2008;12:702–13. https://doi.org/10.1109/TEVC.2008.919004.
[53] Ekinci S, Izci D, Abu Zitar R, Alsoud AR, Abualigah L. Development of Lévy flight-based reptile search algorithm with local search ability for power systems engineering design problems. Neural Comput Appl 2022;34:20263–83. https://doi.org/10.1007/s00521-022-07575-w.
[54] Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH. Prairie dog optimization algorithm. Neural Comput Appl 2022;34:20017–65. https://doi.org/10.1007/s00521-022-07530-9.
[55] Izci D, Ekinci S. A novel-enhanced metaheuristic algorithm for FOPID-controlled and Bode's ideal transfer function–based buck converter system. Trans Inst Meas Control 2023;45:1854–72. https://doi.org/10.1177/01423312221140671.
[56] Snášel V, Rizk-Allah RM, Izci D, Ekinci S. Weighted mean of vectors optimization algorithm and its application in designing the power system stabilizer. Appl Soft Comput 2023;136:110085. https://doi.org/10.1016/j.asoc.2023.110085.
[57] Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH. Aquila optimizer: A novel meta-heuristic optimization algorithm. Comput Ind Eng 2021;157:107250. https://doi.org/10.1016/j.cie.2021.107250.
[58] Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M. Spider wasp optimizer: a novel meta-heuristic optimization algorithm. Artif Intell Rev 2023;56:11675–738. https://doi.org/10.1007/s10462-023-10446-y.
[59] Ghasemi M, Golalipour K, Zare M, Mirjalili S, Trojovský P, Abualigah L, et al. Flood algorithm (FLA): an efficient inspired meta-heuristic for engineering optimization. J Supercomput 2024;80:22913–3017. https://doi.org/10.1007/s11227-024-06291-7.