Available *online* at : http://jnte.ft.unand.ac.id/

Jurnal Nasional Teknik Elektro

| ISSN (Print) 2302-2949 | ISSN (Online) 2407-7267 |

Natural Exponential Inertia Weight and Acceleration Coefficient Particle Swarm Optimization Algorithm tuned PID Controller for DC Motor Speed Control.

Dominic Adu-Buabeng, Yaw Opoku Mensah Sekyere, Francis Boafo Effah

Department of Electrical/Electronic Engineering, KNUST, Kumasi, Ghana.

ARTICLE INFORMATION

Received: July 15, 2025
Revised: September 08, 2025
Accepted: November 28, 2025
Available online: November 30, 2025

KEYWORDS

NExIWAC, Particle Swarm Optimization, PID Controller Tuning, Metaheuristic Algorithms, Step Response Analysis, Integral of Time-weighted Absolute Error

CORRESPONDENCE

E-mail: yaw.sekyere@knust.edu.gh

ABSTRACT

This paper presents a novel optimization algorithm, the NExIWAC (Natural Exponential Inertia Weight and Acceleration Coefficient) variant of Particle Swarm Optimization (PSO), for tuning PID controllers in DC motor speed control systems. The proposed NExIWAC algorithm improves control performance by dynamically adjusting the inertia weight and acceleration coefficients during optimization. To evaluate its effectiveness, the NExIWAC-tuned PID controller was compared against five established metaheuristic algorithms: Atomic Search Optimization (ASO), Sand Cat Swarm Optimization (SCSO), Grey Wolf Optimization (GWO), Invasive Weed Optimization (IWO), and Stochastic Fractal Search (SFS). The system's step response was analyzed under a reference speed demand of 1 p.u., with performance metrics including steady-state error, rise time, settling time, overshoot, and Integral of Time-weighted Absolute Error (ITAE). The NExIWAC algorithm demonstrated superior performance, achieving the fastest rise and settling times, zero steady-state error, and the lowest ITAE value among the tested algorithms. A robustness analysis was conducted by varying motor parameters, such as armature resistance and motor constant, by ±50%. The NExIWAC-PID controller exhibited stable and reliable performance under all conditions. Stability analysis through Bode plots and pole-zero mapping further confirmed the system's robust behavior, with a high phase margin and poles located in the left half of the complex plane. The results indicate that the NExIWAC algorithm is a powerful and reliable optimization tool for tuning PID controllers in DC motor applications, offering significant advantages in terms of precision, stability, and adaptability.

INTRODUCTION

DC motors are widely used in various industrial and commercial applications due to their simplicity, reliability, and ease of control. The speed control of DC motors is particularly crucial in applications where precision and adaptability are required, such as in robotics, conveyor systems, and electric vehicles [1]. A well-regulated speed ensures that the motor performs efficiently and meets the specific demands of the task [2]. Traditionally, Proportional-Integral-Derivative (PID) controllers have been the standard approach for controlling the speed of DC motors because of their straightforward design and effective response characteristics. However, optimizing PID parameters to achieve the desired speed control performance can be challenging due to the nonlinearities and dynamic behavior inherent in DC motors. As the complexity and performance demands of modern applications continue to grow, there is a pressing need for more refined methods to optimize PID controller parameters, ensuring that DC motors operate at peak efficiency while maintaining stability and responsiveness under varying load conditions.

PID controllers have been a fundamental tool in control systems across various industries due to their simplicity and efficacy in enhancing transient and steady-state performance [3], [4], [5]. However, if not properly tuned, PID controllers, while widely used, often fall short in addressing the complexities of modern systems such as DC motor control, where dynamics can be nonlinear and time-varying. Tuning the parameters of PID controllers is crucial for optimal performance, but traditional methods like Ziegler-Nichols [6], Cohen-Coon [6], and pole placement [7] often result in suboptimal solutions. These methods do not always provide the robustness needed for complex systems, leading to exploring alternative approaches, such as heuristic optimization algorithms [5].

Metaheuristic optimization algorithms, including the original Particle Swarm Optimization (PSO) [8], Genetic Algorithms (GA) [9], Simulated Annealing (SA) [10], and Bacterial Swarm Optimization (BSO) [11], have become popular for PID controller tuning. In applying metaheuristic algorithms to DC motor speed control, the Aquila optimizer [12], Sand Cat optimizer [2], and Atom Search optimizer [13] offer the advantage of searching a broader solution space, which can lead to better optimization outcomes. However, the performance of these algorithms is heavily influenced by their inability to escape local minima and their convergence speed. For instance, PSO is known for its simplicity and effectiveness but often suffers from

https://doi.org/10.25077/jnte.v14n3.1401.2025

Table 1. Comparison of Algorithms for Tunning PID Controller, Their Optimal Values and Functions for DC Motor Speed Control

S/N	Algorithms	Optimal Values			Objectives Functions
		Kp	Ki	Kd	
1	Grey Wolf Optimization (GWO) [28]	6.8984	0.5626	0.9293	ITAE
2	Invasive Weed Optimization (IWO) [26]	1.5782	0.4372	0.0481	ITSE
3	Stochastic Fractal Search (SFS) [27]	1.6315	0.2798	0.2395	ITSE
4	Sand Cat Swarm Optimization (SCSO) [2]	3.6182	10	0.5611	ITAE
5	Atomic Search Optimization (ASO)	1.383	1.461	0.546	ITSE
6	Chaotic Atomic Search Optimization [13]	19.7722	9.117	8.1189	ITAE

issues such as premature convergence and the need for proper parameter tuning [14].

Since 1995, the original Particle Swarm Optimization [8][15] has undergone numerous enhancements to address its inherent limitations and improve performance across diverse optimization problems. Hybrid PSO variants have been introduced, combining PSO with other optimization algorithms like genetic algorithms (PSO-GA) [9], Grey Wolf optimization (PSO-GWO) [16], differential evolution (PSO-DE) [17], or simulated annealing (hPSO-SA) [18] to leverage the strengths of each method and further enhance the quality of the solution. These hybrid approaches often incorporate mechanisms for dynamic learning, memory, or multi-population strategies, which contribute to more robust and versatile optimization performance. Furthermore, modifications like chaotic PSO, quantum-behaved PSO, and discrete PSO have expanded the applicability of PSO to a broader range of problems, including those in discrete or highly nonlinear spaces [19]. These advancements have made PSO a more powerful and flexible tool for solving complex optimization problems in various fields.

However, the increased computational demands of hybrid approaches often reduce their practicality compared to simpler variants that prioritize parameter regulation. A major advancement in this area has been the development of adaptive strategies for tuning key PSO parameters, particularly inertia weight and acceleration coefficients [14], to accelerate convergence and prevent premature stagnation. One of the most influential variants is the linearly decreasing time-varying inertia weight introduced in [20], which assumes that particles should begin the search with higher velocities to promote exploration and gradually reduce inertia weight to enhance exploitation as they approach optimal regions. This concept was later refined by [21], who proposed a natural exponential decay function as an alternative to the linear schedule, providing a more flexible mechanism for adjusting inertia dynamics.

In this study, a new PSO variant that integrates exponential inertia weight with exponential acceleration coefficients, referred to as the Natural Exponential Inertia Weight and Acceleration Coefficient PSO (NExIWAC-PSO), is proposed and applied to 16 benchmark optimization problems. Its performance is compared against several established metaheuristic algorithms. The proposed NExIWAC-PSO is then utilized to optimally tune a PID controller for DC motor speed regulation using the ITAE criterion as the objective function. Its transient-response performance is also compared with Grey Wolf Optimization (GWO) [38], Invasive Weed Optimization (IWO) [43], and Stochastic Fractal Search (SFS) [44] under identical ITAE-based evaluation. Furthermore, stability and robustness analyses are conducted under varying DC motor parameters to assess the reliability and practical applicability of the proposed NExIWAC-PID approach.

Related Works

Several optimization techniques for tuning PID controllers in DC motor speed control systems have been proposed in the literature, each with distinct advantages and limitations. Traditionally, PID controllers were tuned using methods such as Ziegler-Nichols, Cohen-Coon, and manual trial-and-error techniques, which, despite their simplicity, often resulted in poor performance, excessive time consumption, and overshooting issues. Metaheuristic algorithms have overcome these drawbacks and significantly enhanced the efficiency of PID tuning. For instance, Grey Wolf Optimization (GWO) [28], Invasive Weed Optimization (IWO) [26], and Stochastic Fractal Search (SFS) [27] algorithms have demonstrated improved transient response and accuracy in PID tuning. However, this research focuses on developing a novel Exponential Inertia Weight and Acceleration Coefficient (NExIWAC) PSO algorithm for PID tuning in DC motor speed control. The proposed algorithm aims to achieve superior performance in terms of transient response, stability, and robustness under varying motor parameters, compared to state-of-the-art metaheuristic approaches. Table 1 presents some outputs of metaheuristic algorithms for tuning PID controllers for DC motor speed, giving their optimal values and objective functions.

METHODS

Overview of Original Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based algorithm where each particle represents a candidate solution and updates its velocity based on inertia, personal best, and global best components [15]. The dynamics of velocity and position are governed by the standard PSO equations (1) and (2), where w is the inertia weight. c_1, c_2 are the acceleration coefficients, $x_i(t)$ being the current position of a particle, $x_i(t+1)$ is the updated position of the particle, $p_i(t)$ = the personal best of a particle, $g(t)$ = the global best of a particle, $v_i(t)$ is the velocity of a particle and $v_i(t+1)$ is updated velocity of the particle with the updated position $x_i(t+1)$ [51]. The cognitive term guides a particle toward its own best experience, whereas the social term encourages movement toward the globally best solution. In the original PSO, both inertia weight and acceleration coefficients remain constant, which limits adaptability and affects the balance between exploration and exploitation. This limitation motivates the

https://doi.org/10.25077/jnte.v14n3.1401.2025

development of parameter-adjustment strategies, including the variant proposed in this study.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{1}$$

$$v_i(t+1) = wv_i(t) + c_1(p_i(t) - x_i(t)) + c_2(g(t) - x_i(t)) \tag{2}$$

Proposed NExIWAC PSO Variant

In the proposed PSO variant named Natural Exponential Inertia Weight and Acceleration Coefficient (NExIWAC), the natural exponential function given by (3) and (4) are applied to replace the constants used for inertia weight and acceleration coefficients of the original PSO. The use of exponentially decaying functions for both the inertia weight and the cognitive acceleration coefficient in PSO creates a highly dynamic variant that enhances the algorithm's performance.

$$w(t) = W_{min} + (W_{max} - W_{min})e^{-\left(\frac{t}{T/4}\right)^2} \tag{3}$$

$$c_1(t) = C_{min} + (C_{max} - C_{min})e^{-\left(\frac{t}{T/4}\right)^2} \tag{4}$$

Where *w* is the inertia weight, *c₁* is the cognitive acceleration coefficient, *t* is the current iteration number, and *T* is the maximum number of iterations.

The proposed (3) and (4) for inertia weight and acceleration coefficient ensure that the inertia weight decreases rapidly at the beginning of the iterations, promoting broad exploration of the search space, allowing the swarm to escape local minima and explore diverse potential solutions. As the iterations progress, the exponential decay in these parameters reduces the emphasis on exploration and enhances exploitation, guiding the particles toward more precise and stable convergence to optimal solutions. As the iterations progress, the inertia weight decreases more slowly as compared to the linearly decreasing, allowing the particles to focus on fine-tuning their positions in the search space and improving the convergence towards the optimal solution. Similarly, the cognitive acceleration coefficient given by (4) follows the same exponential decay, initially encouraging particles to explore their best-found positions vigorously. Over time, as *c₁* decreases, the particles reduce their self-confidence, allowing the swarm's collective experience to guide them more effectively towards the global optimum. This combination of dynamic inertia weight and acceleration coefficient helps maintain a balance between exploration and exploitation, thereby improving the overall effectiveness of the PSO variant.

DC Motor model

Figure 1 illustrates the model of a separately excited DC motor, a common type of motor used in various applications due to its precise speed control capabilities. In this model, e_a represents the applied armature voltage, i_a is the armature current, R_a is the armature resistance, and L_a is the armature inductance. The back electromotive force (emf), e_b is generated as the motor's armature rotates, opposing the applied voltage and proportional to the motor's speed ω. The field current, i_f, is kept constant, ensuring a constant magnetic field. The motor produces torque T which drives a load with inertia J and friction B as depicted in the transfer function representation of the DC motor presented in Figure 2. The dynamic behavior of the DC motor which is mathematically described in equations give the relationship between electrical and mechanical parameters, making it a fundamental model in control system design and analysis.

$$e_b = K_b \frac{d\theta}{dt} = K_b \omega \tag{5}$$

$$e_a = L_a \frac{di_a}{dt} + R_a i_a + e_b \tag{6}$$

$$T = J\frac{d\omega}{dt} + B\omega = Ki_a \tag{7}$$

$$E_b(s) = K_b \omega(s) \tag{8}$$

$$E_b(s) = (L_a + R_a)I_a(s) \tag{9}$$

$$T(s) = (Js + B)\omega(s) = KI_a(s) \tag{10}$$

i_f: field current, A
e_a: applied armature voltage, V
e_b: back electromotive force, V
T: motor torque, N.m
ω: angular speed of motor shaft, rad/s
J: inertia torque of motor, kg.m2
K_b: electromotive force constant, N.m/A
B: motor friction constant, N.m.s/rad

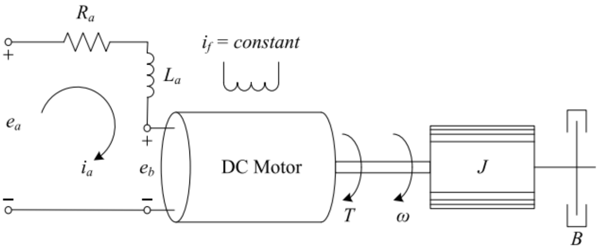

Figure 1: Model Of a Separately Excited DC Motor

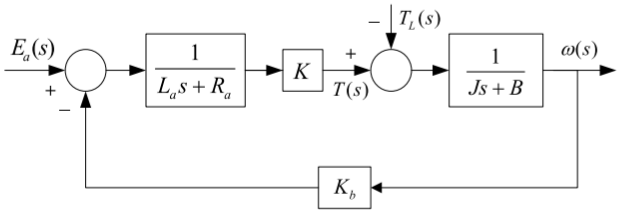

Figure 2: Transfer function model representation of DC motor system

Figure 2 illustrates the DC motor system, showing the relationship between the applied armature voltage $E_a(s)$ and the output speed of the motor $\omega(s)$. The open-loop transfer function of the DC motor system, which links these two variables, is derived from the motor's electrical and mechanical dynamics. The transfer function $G(s)$ is given by (11).

$$G(s) = \frac{\omega(s)}{E_a(s)} = \frac{K}{(L_a s + R_a)(Js + B) + K_b K} \tag{11}$$

This represents how the motor's speed responds to changes in the armature voltage, considering the effects of armature resistance, armature inductance, moment of inertia, viscous friction coefficient (*B*), back EMF constant (K_b), and the motor torque constant (*K*). This transfer function is crucial for analyzing the motor's dynamic behavior and designing control strategies to achieve desired performance. The values used for simulation works performed in this research are given in Table 2.

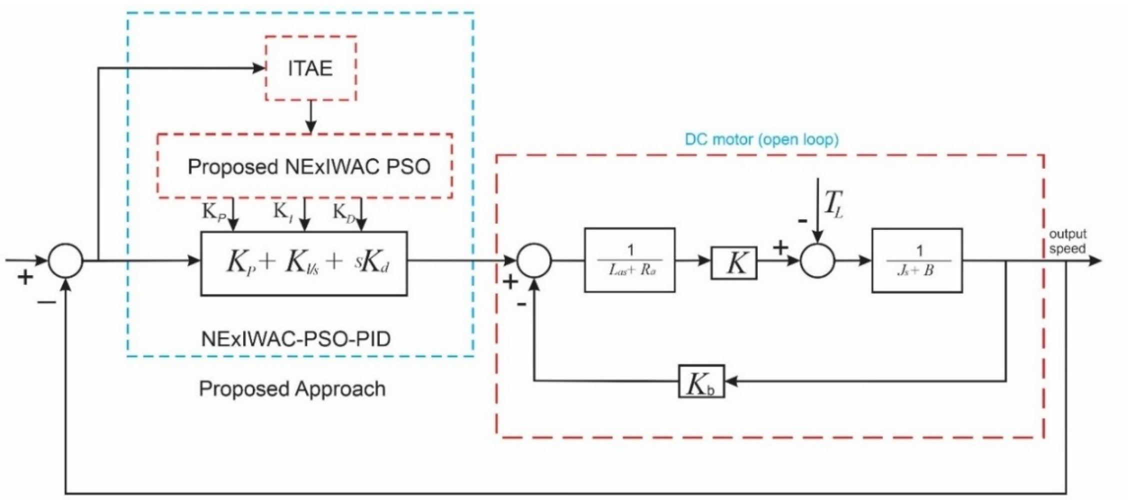

Figure 3: Transfer Function Representation of a Closed-Loop DC Motor System with The PID Controller Integrated

Table 2: Motor Parameters

Parameter	Value
Armature resistance R_a	0.4 Ω
Inductance L_a	2.7 H
Moment of inertia, J	0.0004 kgm^2
Friction Coefficient, B	0.0022 Nms/ rad
Motor torque constant, K	0.015 N. m/ A
Back emf constant K_b	0.05 Vs

PID Controller

PID controllers are widely regarded as the most commonly used controllers in industrial applications. When implemented in a DC motor speed control system, a PID controller processes the error signal, which is the difference between the desired reference speed and the actual measured speed. The controller then applies proportional, integral, and derivative actions to minimize this error and achieve precise speed control. The performance of the PID controller is heavily dependent on the optimal tuning of its proportional, integral, and derivative gains. In this study, these gains are optimally determined using the proposed novel NExIWAC-PSO variant. Figure 3 illustrates the transfer function representation of a closed-loop DC motor system with the PID controller integrated, showing the interaction between the controller and the motor's dynamics.

Objective Function Formulation

Metaheuristic algorithms typically rely on an objective function that they aim to minimize or maximize to guide the optimization process. In this study, the chosen objective function is the Integral Time Absolute Error (ITAE), a common performance criterion used in the analysis of control systems. The NExIWAC-PSO variants are employed during iterative runs to optimize the control of a closed-loop DC motor system. Specifically, these algorithms seek to minimize the ITAE of the system, which quantifies the overall error by considering the magnitude of the speed error over time, with the goal of achieving faster and more accurate speed regulation. The error metric is defined as the difference between the desired (setpoint) speed and the actual measured speed of the motor. The ITAE is mathematically expressed by equation (12), which integrates the product of time and the absolute value of the speed error over the duration of the response, providing a weighted measure that emphasizes errors occurring later in the response.

$$ITAE = \int_{0}^{t_{sim}} |e|.t\, dt \qquad (12)$$

Where e = the difference between the reference speed demand and the actual speed.

PID Tuning Algorithm

The improved particle swarm optimization variant, NExIWAC, is implemented using the pseudo-code below to obtain the optimal gains of the PID controller:

```
**Simulation setup and Initialization**
Model the closed-loop DC motor in Simulink using parameters from Table 3.
Define the swarm size $nPOP$ and the number of dimensions $nVAR$ as 3.
for each particle $i \in [1 \ldots\ldots nPOP]$
Generate randomly $X_i$ and $V_i$ and evaluate the $ITAE$ values of $X_i$
Set $Pbest_i = X_i$
**end for**
Set $Gbest = Pbest_i$
**for** each particle $i \in [1 \ldots\ldots nPOP]$
 **if** $Pbest_i < Gbest$ **then**
     $Gbest = Pbest_i$
 **end if**
**end for**
**while** $t$ < maximum number of iterations
**for** each particle $i \in [1 \ldots S]$
Evaluate its velocity $v_{id}\ (t+1)$ using Equation (1)
Update the position $x_{id}\ (t+1)$ for the particle using Equation (2)
If $f(x_i(t+1)) < f(Pbest_i)$ **then**
$Pbest_i = x_1(t+1)$
$f(Pbest_i) = f(x_i(t+1))$
**end if**
**if** $f(Pbest_i) < f(Gbest)$ **then**
$Gbest = Pbest_i$
$f(Gbest) = f(Pbest_i)$
**end if**
**end for**
$t = t + 1$
**end while**
return $Gbest$
```

The NExIWAC-PSO procedure begins by modeling the closed-loop DC motor in Simulink and setting the swarm size along with the search dimension, which includes the three PID gains. Each particle is randomly initialized with positions and velocities, and the ITAE value is calculated to identify the initial personal bests. The best among these is set as the initial global best. In each iteration, particle velocities and positions are updated using the NExIWAC formulas in Equations (1) and (2). After each update, the ITAE value at the new position is computed. If a particle finds a better solution than its current personal best, it updates accordingly. The global best is also updated whenever any particle finds a superior value. This process repeats until the maximum number of iterations is reached, at which point the global best particle indicates the optimal PID gains.

Table 3: NExIWAC-PSO Simulation Parameters

Parameters	**Values**
Minimum inertial weight (w_{min})	0.1
Maximum inertial weight (w_{max})	1
Acceleration coefficient (c_1)	2
Acceleration coefficient (c_2)	2
Population size	100
Maximum iteration	1000
Number of Runs	10

Table 4 presents the optimal PID gain values obtained by the proposed NExIWAC-PSO algorithm after completing the simulation-based optimization process. These gains represent the best-performing solution identified by the swarm using the ITAE criterion as the objective function within the closed-loop DC motor model. The resulting proportional, integral, and derivative gains are K_P=19.561, K_I=5.1216, and K_D=4.4665, respectively. These values correspond to the global best particle at the final iteration and reflect the ability of the NExIWAC-PSO to converge toward a parameter set that minimizes the transient error of the system. The obtained gains serve as the reference parameters for evaluating the closed-loop performance and conducting subsequent stability and robustness analyses.

Table 4: NExIWAC-PSO Tuned PID Gain Values Obtained from the Simulation

	Gain Value
Proportional (K_P)	19.561
Integral (K_I))	5.1216
Derivative (K_D)	4.4665

Implementation

All computational simulations are performed using MATLAB/Simulink 2023a software on an AMD Ryzen 5 4500U 2.38 GHz Processor on a Windows 11 installed PC with 8 GB RAM. The simulation setup is presented in Figure 4.

RESULTS AND DISCUSSIONS

Step response

The step response performance of the proposed NExIWAC algorithm, along with comparison algorithms for tuning PID controllers comprising Atomic Search Optimization (ASO) [13], Sand Cat Swarm Optimization (SCSO) [2], Grey Wolf Optimization (GWO) [25], Invasive Weed Optimization (IWO) [26], and the Stochastic Fractal Search (SFS) algorithm [27] The reference speed demand of 1 p.u. is presented in Table 5 and Figure 5. Table 5 shows that the proposed NExIWAC algorithm outperforms the others across multiple performance metrics. It achieves a steady-state error of 0%, meaning the system reaches and maintains the desired motor speed with no residual error. The settling time of 0.13 seconds and rise time of 0.03 seconds are also the shortest among all the algorithms, highlighting the control technique's ability to stabilize the system quickly.

Table 5: System Performance for Step Response Analysis

Control Strategy	Steady state error	Settling Time (s)	Rise Time (s)	Overshoot	ITAE
NExIWAC (Proposed)	**0**	**0.13**	**0.03**	5.9%	**0.1195**
ASO-PID [13]	0.50%	0.17	0.06	**0%**	0.2242
SAND CAT [2]	3.20%	3.19	0.21	17.8%	1.071
GWO [25]	1.00%	0.54	0.14	1.1%	0.5299
IWO [26]	1.38%	1.57	0.51	6.7%	1.561
SFS [27]	2.00%	3.11	0.60	**0 %**	1.608

Despite a small overshoot of 5.9%, the NExIWAC still performs significantly better in terms of the Integral of Time-weighted Absolute Error, with a value of 0.1195, the lowest among the comparisons. This metric indicates that the overall error throughout the control process is minimized effectively. The ASO algorithm-tuned PID performs decently, but it has a steady-state error of 0.50% and slightly longer settling times (0.17 seconds) and rise times (0.06 seconds) compared to the PID tuned by NExIWAC. However, it achieves a 0% overshoot, making it a viable option for systems requiring strict overshoot control, although its overall error performance (ITAE = 0.2242) is higher. The SAND CAT algorithm shows relatively poor performance. It has a steady-state error of 3.20%, the highest among all algorithms, indicating a significant deviation from the desired speed. Its settling time of 3.19 seconds and rise time of 0.21 seconds are also among the longest, and it suffers from an excessively large overshoot of 17.8%, making it less suitable for high-performance control. The ITAE value of 1.071 further confirms its suboptimal performance.

Similarly, the IWO algorithm-based strategy performs poorly, with a settling time of 1.57 seconds and a rise time of 0.51 seconds, indicating that it is slow to respond to changes in speed. Although its steady-state error (0.38%) and overshoot (6.7%) are moderate, its ITAE value of 1.561 shows that the cumulative error is relatively large, making it less effective. The GWO algorithm shows promise, with a steady-state error of 1.00% and relatively low overshoot (1.1%). However, its settling time (0.54 seconds) and rise time (0.14 seconds) are still longer than those of the proposed NExIWAC. Its ITAE value of 0.5299 is better than several other algorithms, but still not as low as that of the NExIWAC.

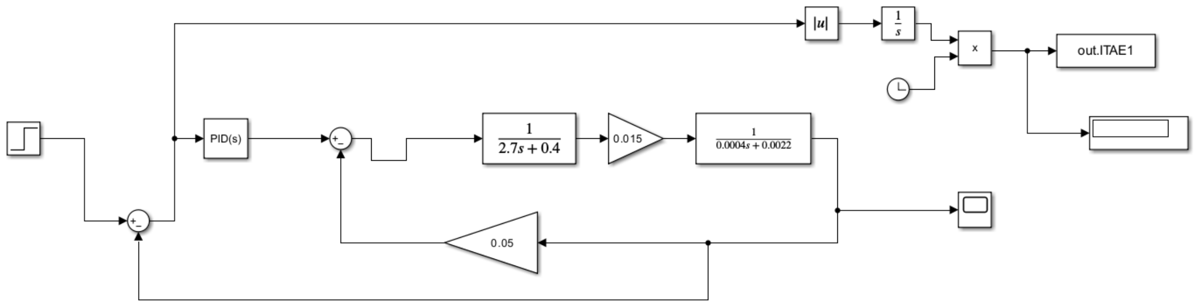

Figure 4: Simulink Simulation Setup

The SFS algorithm, as reported in the literature, is among the least effective for PID tuning, with a steady-state error of 2.00%, a long settling time of 3.11 seconds, and a rise time of 0.60 seconds. While it shows no overshoot, its ITAE of 1.608 indicates poor overall performance. The NExIWAC algorithm emerges as the top metaheuristic for tuning PID controllers in DC motor speed control, offering the fastest transient response, lowest steady-state error, and most favorable ITAE. Other algorithms like ASO-PID and GWO perform reasonably but lack in certain areas. Conversely, SAND CAT, IWO, and SFS demonstrate significant shortcomings in both transient and steady-state performance.

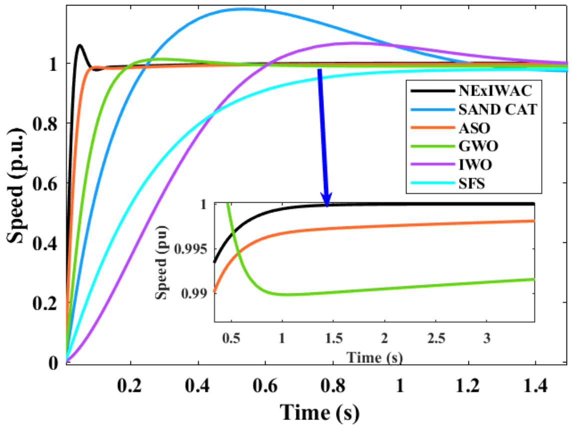

Figure 5: Step Response Performance Metrics

Robustness Analysis

Four distinct operating points were considered to evaluate the robustness of the proposed NExIWAC-tuned PID controller for DC motor speed control, as shown in Table 6. In this analysis, the armature resistance (**R$_a$**) and the motor constant (**K**) were varied by ±50% and ±40% respectively. These variations simulate real-world conditions where factors such as temperature fluctuations, long-term wear, and operational stress impact motor parameters. For instance, increases in temperature or long-term wear can raise armature resistance. At the same time, changes in the magnetic field strength and mechanical stress from high-speed operations can affect the motor constant (**K**).

Table 6: The four operating scenarios are designed to test the controller's performance under different parameter variations

Scenario	R_a Ω	*K N.m/A*
1	0.20	0.009
2	0.20	0.021
3	0.60	0.009
4	0.60	0.021

Scenario 1

In Operating Scenario 1, where the armature resistance is 0.20Ω (-50%) and the motor constant, K, is 0.009 N.m/A (-40%), the results presented in Table 7 and Figure 6 demonstrate that the NExIWAC-tuned PID controller outperforms other algorithms in speed, precision, and overall control efficiency. It achieves a 0% steady-state error, the fastest settling time of 0.16 seconds, and a rise time of 0.03 seconds, with no overshoot, indicating robustness in maintaining accurate motor speed control.

Table 7: Systems Performance for Operating Scenario 1

Control Strategy	Steady state error	Settling Time (s)	Rise Time (s)	Overshoot	ITAE
NExIWAC (Proposed)	**0**	**0.16**	**0.03**	**0%**	**0.2523**
ASO-PID [13]	0.35%	0.21	0.06	**0%**	0.361
SAND CAT [2]	3.16%	3.43	0.21	22.1%	2.221
GWO [25]	1.00%	0.71	0.14	**0%**	0.7207
IWO [26]	1.12%	1.87	0.51	4.3%	3.094
SFS [27]	2.00%	3.52	0.60	**0 %**	2.492

The Integral of Time-weighted Absolute Error for NExIWAC is the lowest at 0.2523, highlighting its superior ability to minimize cumulative error compared to ASO, GWO, IWO, SFS, and especially SAND CAT. The latter exhibited the highest steady-state error at 3.16% and the longest settling time of 3.43 seconds. These results confirm that the NExIWAC algorithm is highly effective under varying motor parameters, ensuring stable, precise, and efficient performance.

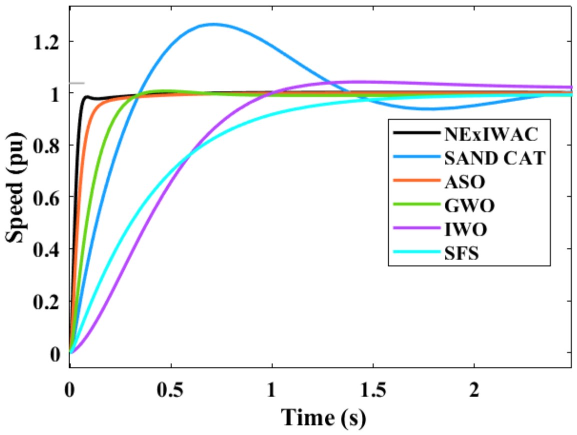

Figure 6: step response for Scenario 1

Scenario 2:

In Operating Scenario 2, where the armature resistance is 0.20 Ω (-50%) and the motor constant, K, is 0.021 V/rad/s (+40%), the NExIWAC algorithm demonstrates exceptional performance compared to the other comparison algorithms, as presented in Table 8 and Figure 7.

Table 8: Systems Performance for Operating Scenario 2

Control Strategy	Steady state error	Settling Time (s)	Rise Time (s)	Overshoot	ITAE
NExIWAC (Proposed)	**0**	**0.13**	**0.03**	13.7%	**0.09789**
ASO-PID [13]	0.55%	0.17	0.06	4.7%	0.1538
SAND CAT [2]	3.32%	3.19	0.41	13.3%	0.7255
GWO [25]	1.08%	0.54	0.19	2.1%	0.3632
IWO [26]	0.39%	1.57	0.51	4.3%	1.303
SFS [27]	2.12%	3.11	0.80	**0 %**	1.047

It achieves 0% steady-state error, meaning the system reaches the desired speed without any residual error while also exhibiting the fastest settling time of 0.13 seconds and the shortest rise time of 0.03 seconds. Despite a relatively higher overshoot of 13.7%, the NExIWAC algorithm still outperforms others in overall error minimization, with the lowest ITAE value of 0.09789, effectively reducing cumulative error throughout the control process.

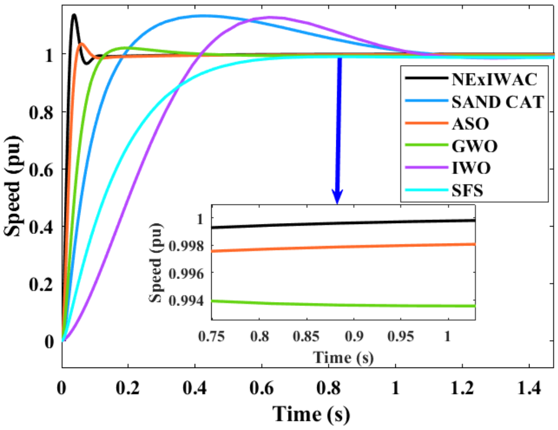

Figure 7: Step response for Scenario 2

Comparatively, the ASO-PID controller shows a moderate performance, with a steady-state error of 0.55%, a settling time of 0.17 seconds, and a rise time of 0.06 seconds. While ASO has a lower overshoot of 4.7%, its ITAE value of 0.1538 is still notably higher than that of NExIWAC. The SAND CAT algorithm performs poorly in this scenario, with the highest steady-state error of 3.32%, the longest settling time of 3.19 seconds, and a rise time of 0.41 seconds, indicating sluggish system response and poor error management (ITAE = 0.7255). GWO shows promise, with a moderate steady-state error of 1.08% and a relatively low overshoot of 2.1%. However, it exhibits longer settling times (0.54 seconds) and rise times (0.19 seconds) compared to NExIWAC. IWO and SFS also underperform compared to NExIWAC, with IWO having a settling time of 1.57 seconds and a rise time of 0.51 seconds, and SFS lagging with the longest rise time (0.80 seconds) and a high ITAE value (1.047). In summary, the NExIWAC algorithm maintains the fastest system response and minimizes errors under this operating scenario, making it the most suitable choice for robust DC motor speed control.

Operating Scenario 3:

In Operating Scenario 3, where the armature resistance is 0.60 Ω (+50%) and motor constant, K = 0.009 *N.m/A* (-40%), the NExIWAC algorithm once again exhibits outstanding performance compared to other algorithms, as presented in Table 9 and Figure 8. NExIWAC achieves 0% steady-state error, the fastest settling time of 0.13 seconds, and the shortest rise time of 0.03 seconds. It records a modest overshoot of 5.9% and the lowest ITAE value of 0.1931, demonstrating superior overall error minimization and rapid system response.

Table 9: Systems Performance for Operating Scenario 3

Control Strategy	Steady state error	Settling Time (s)	Rise Time (s)	Overshoot	ITAE
NExIWAC (Proposed)	**0**	**0.13**	**0.03**	**0%**	**0.1931**
ASO-PID [13]	0.50%	0.17	0.06	**0%**	0.2999
SAND CAT [2]	3.20%	3.19	0.21	17.8%	1.821
GWO [25]	1.00%	0.54	0.14	0.1%	0.7059
IWO [26]	0.38%	1.57	0.51	**0%**	2.478
SFS [27]	2.00%	3.11	0.60	**0 %**	2.173

The ASO-PID controller exhibits moderate performance, characterized by a steady-state error of 0.50%, a settling time of 0.17 seconds, and a rise time of 0.06 seconds. It shows no overshoot and has an ITAE value of 0.2999, which is higher than that of NExIWAC, indicating slower error convergence. SAND CAT exhibits poor performance, with the highest steady-state error of 3.20%, a long settling time of 3.19 seconds, and a rise time of 0.21 seconds. It shows a significant overshoot of 17.8% and an ITAE value of 1.821, indicating sluggish response and subpar error management.

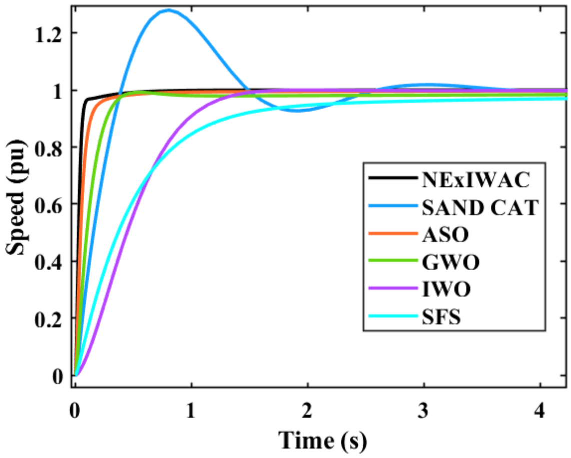

Figure 8: Graphical Representation of Operating Scenario 3

GWO performs reasonably with a steady-state error of 1.00%, a settling time of 0.54 seconds, and a rise time of 0.14 seconds. It has a low overshoot of 1.1%, but its ITAE value of 0.7059 is still considerably higher than that of NExIWAC. IWO achieves a steady-state error of 0.38%, though its settling time is relatively long at 1.57 seconds, with a rise time of 0.51 seconds. It records an overshoot of 6.7% and the highest ITAE value of 2.478,

https://doi.org/10.25077/jnte.v14n3.1401.2025

reflecting sluggish error reduction. SFS exhibits weak performance, characterized by a steady-state error of 2.00%, a settling time of 3.11 seconds, and a rise time of 0.60 seconds. It has no overshoot but reports an ITAE value of 2.173, indicating the slowest and least effective error minimization.

Operating Scenario 4:

In Operating Scenario 4, where the armature resistance is 0.60 Ω (+50%) and the motor constant, K, is 0.021 N.m/A (+40%), the NExIWAC algorithm demonstrates superior performance compared to other algorithms, as shown in Table 10 and Figure 9. It achieves 0% steady-state error, indicating that the system reaches and maintains the desired speed without any residual offset. Furthermore, NExIWAC achieves the fastest settling time of 0.13 seconds and the shortest rise time of 0.03 seconds. Although it exhibits a slightly higher overshoot of 12.9%, its Integrated Time Absolute Error (ITAE) value is the lowest at 0.1245, representing the most effective cumulative error reduction throughout the control process.

Table 10: Systems Performance for Operating Scenario 4

Control Strategy	Steady state error	Settling Time (s)	Rise Time (s)	Overshoot	ITAE
NExIWAC (Proposed)	**0**	**0.13**	**0.03**	12.9%	**0.1245**
ASO-PID [13]	0.56%	0.17	0.06	3.3%	0.2081
SAND CAT [2]	3.32%	3.19	0.43	12.5%	0.7093
GWO [25]	1.08%	0.54	0.19	1.5%	0.4635
IWO [26]	0.39%	1.57	0.51	9.9%	1.454
SFS [27]	2.12%	3.11	0.80	**0** %	1.428

The SAND CAT algorithm performs poorly in this scenario, with a steady-state error of 3.32%, a settling time of 3.19 seconds, and a rise time of 0.43 seconds. It also has a 12.5% overshoot and an ITAE of 0.7093, indicating slow response and limited error reduction. GWO shows acceptable results, with a steady-state error of 1.08%, a settling time of 0.54 seconds, and a rise time of 0.19 seconds. Its overshoot is 1.5%, and its ITAE of 0.4635, though better, remains higher than NExIWAC's, suggesting less speed and accuracy. IWO records a steady-state error of 0.39%, a settling time of 1.57 seconds, and a rise time of 0.51 seconds. Its overshoot is 9.9%, with an ITAE of 1.454, reflecting sluggishness and limited error minimization.

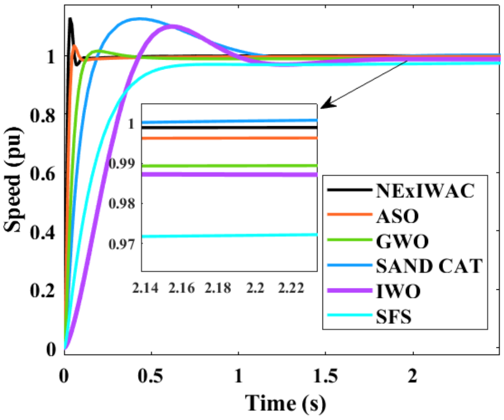

Figure 9: Graphical Representation of Operating Scenario 4

SFS presents one of the weakest performances in this scenario, with a steady-state error of 2.12%, a settling time of 3.11 seconds, and the longest rise time of 0.80 seconds. It records 0% overshoot but suffers from the highest ITAE value of 1.428, indicating slow response and poor overall error control.

Stability Analysis

The stability analysis of the NExIWAC-tuned PID system, based on both the Bode plot in Figure 10 and the pole-zero map in Figure 11, provides strong evidence of its robust stability. As shown in Figure 10, the magnitude plot indicates that the system maintains a gain of approximately 80 dB at low frequencies, which gradually decreases as the frequency increases, a typical characteristic of a stable system. The phase plot starts at -90 degrees at low frequencies, then increases to around 0 degrees before gradually decreasing to -180 degrees. Importantly, the system exhibits a phase margin (Pm) of 62.1 degrees at 56.3 rad/s, indicating a substantial tolerance for phase shifts before instability occurs. Additionally, the gain margin (Gm), which represents the amount of gain the system can tolerate before becoming unstable, combined with the delay margin of 0.0192 seconds, further supports the system's robustness. A high phase margin indicates that the system can withstand significant phase changes without losing stability, which is highly desirable for preventing oscillations and ensuring steady performance even in the presence of disturbances.

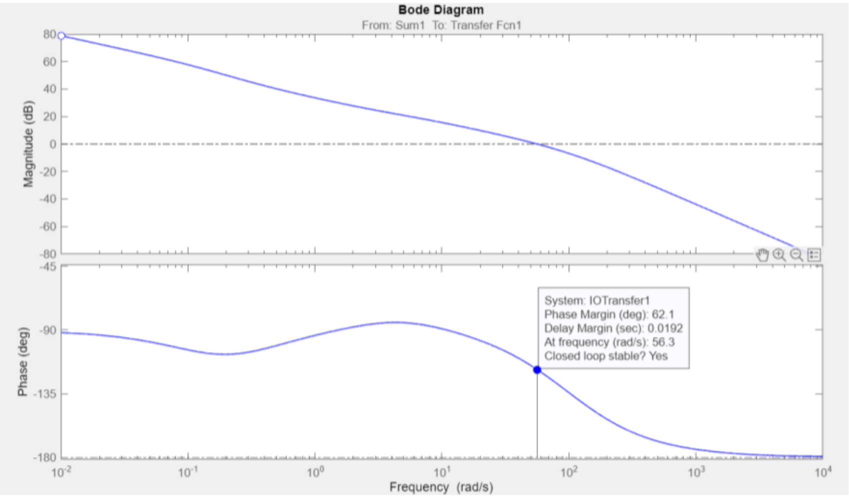

Figure 10: Bode Diagram

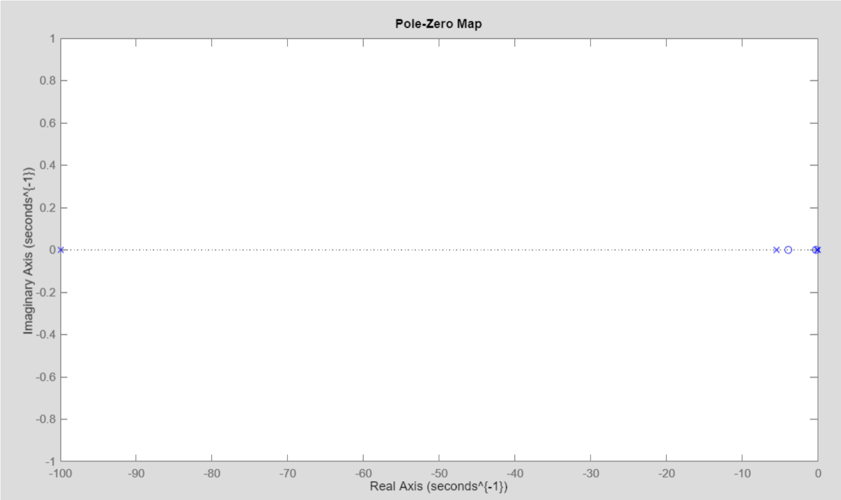

Figure 11: Pole-Zero Diagram

Moreover, the pole-zero map, as shown in Figure 11, aligns with this stability assessment. All poles are located on the left half of the complex plane, indicating that the system is stable. The fact that the poles are positioned on the real axis further confirms that the system does not exhibit oscillatory behavior, which is critical for systems requiring smooth and predictable responses. The presence of a zero to the right of all poles suggests that the system has a fast response and strong disturbance rejection capabilities, making it highly efficient in maintaining control despite external perturbations. In summary, the stability analysis, with favorable phase and gain margins and pole-zero distribution, demonstrates that the NExIWAC-tuned system is robust and stable, capable of handling various operating conditions without compromising performance.

Discussions

Appendix 1 presents sixteen widely used benchmark optimization functions to evaluate the proposed NExIWAC's performance. This evaluation compares NExIWAC-PSO against various metaheuristic algorithms and PSO variants documented in the literature. Appendix 2 presents a comparative analysis of the proposed Naturally Exponential Inertia Weight and Acceleration Coefficient PSO (NExIWAC-PSO) against various PSO variants: ADIWACO-PSO [14], Tanh Inertia Weight (TIW) PSO [22], Linearly Decreasing Inertia Weight (LDIW) PSO [23], Random Inertia Weight (RIW)-PSO [24], and the standard PSO [15] across sixteen benchmark functions, highlighting its superior performance.

For the Sphere function ($f_1(x)$), NExIWAC-PSO demonstrates near-optimal convergence with the smallest mean (6.5905e-217) and standard deviation (2.777e-41), significantly outperforming all other algorithms. On the Rosenbrock function ($f_2(x)$), a complex, non-linear problem, the NExIWAC-PSO achieves the lowest mean value (2.1743e-29), demonstrating its ability to escape local minima more effectively than the comparison PSOs.

For the Step function ($f_3(x)$), the algorithm achieves perfect convergence with both mean and standard deviation (SD) equal to zero, surpassing Tanh-PSO and standard PSO, which exhibit slight deviations. In tackling the multimodal Rastrigin function ($f_4(x)$), NExIWAC-PSO yields the smallest mean (20.0276) and SD (1.7907), indicating its superior balance between exploration and exploitation. For the Ackley function ($f_5(x)$), the algorithm maintains robust performance with a competitive mean (1.678), reinforcing its reliability on multimodal landscapes.

In the Griewank function ($f_6(x)$), the NExIWAC-PSO once again achieves perfect results with both mean and SD equal to zero, reflecting exceptional precision. On the Rotated Hyper Ellipsoid ($f_7(x)$), it again outperforms the other algorithms with the smallest mean value of 2.2264e-05 and a standard deviation of 1.34e-04. In contrast, other algorithms, particularly standard PSO, show significantly higher deviations. For the Sum of Squares function ($f_8(x)$), the proposed algorithm achieves a mean value of 3.09e-06 and SD of 3.11e-06, beaten by only Tanh-PSO, while comparison PSOs struggle with higher values. Similarly, in the Coville function ($f_9(x)$), NExIWAC-PSO outshines others with a mean of 8.8552e-04 and a standard deviation of 5.81e-05, but is only beaten by the LDIW-PSO, which achieves mean and standard deviation values of 2.45136e-07 and 4.8758e-07, respectively. On functions ($f_{10}(x)$), ($f_{12}(x)$), ($f_{13}(x)$), and ($f_{16}(x)$), the proposed PSO consistently achieves perfect convergence with mean and SD values of zero. For Levy ($f_{14}(x)$), the NExIWAC-PSO once again achieves the smallest mean (0.000833174) with minimal standard deviation (0.000806), and on the Zakharov function ($f_{15}(x)$), it records the lowest mean (2336.591) and SD (1957.295336) as well, solidifying its robustness in solving high-dimensional, non-linear problems.

Overall, NExIWAC-PSO demonstrates exceptional accuracy, consistency, and reliability across unimodal and multimodal benchmark functions, achieving minimal mean values and low standard deviations while outperforming existing PSO variants. Its ability to efficiently converge to global optima across diverse optimization landscapes underscores its effectiveness and suitability for complex optimization problems, such as the optimal tuning of PID controllers for DC motor speed control.

CONCLUSION

This study presented the NExIWAC algorithm, a new PSO variant designed for optimal PID tuning in DC motor speed control. Comparative simulations against several metaheuristic

https://doi.org/10.25077/jnte.v14n3.1401.2025

algorithms demonstrated that NExIWAC consistently achieved superior transient and steady-state performance, including faster rise and settling times, minimal steady-state error, reduced overshoot, and the lowest ITAE values. The algorithm produced a highly responsive step response with a rise time of 0.03 s and maintained the smallest cumulative error, highlighting its effectiveness in accurately regulating motor speed.

Robustness and stability analyses further confirmed the reliability of the NExIWAC-tuned controller under variations in motor parameters. Across multiple test conditions, the controller maintained stability, exhibited large phase and gain margins, and kept its poles in the left-half plane, ensuring non-oscillatory and stable behavior. These findings establish NExIWAC as an efficient and robust optimization method for PID tuning in real-world applications. Future work may explore its application to more complex control systems or hardware-based validation.

REFERENCES

[1] A. D. O. D. S. Dantas, A. F. O. D. A. Dantas, J. T. L. S. Campos, D. L. De Almeida Neto, and C. E. T. Dórea, "PID Control for Electric Vehicles Subject to Control and Speed Signal Constraints," *J. Control Sci. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/6259049.

[2] W. Aribowo *et al.*, "Sand cat swarm optimization for controlling PID in DC motor," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 22, no. 2, pp. 462–470, 2024, doi: 10.12928/TELKOMNIKA.v22i2.25630.

[3] Y. O. M. Sekyere, F. B. Effah, and P. Y. Okyere, "Optimally tuned cascaded FOPI-FOPIDN with improved PSO for load frequency control in interconnected power systems with RES," *J. Electr. Syst. Inf. Technol.*, vol. 11, no. 1, p. 25, 2024, doi: 10.1186/s43067-024-00149-x.

[4] Y. O. M. Sekyere, F. B. Effah, and P. Y. Okyere, "A Novel ANFIS Controller for LFC in RES Integrated Three-Area Power System," *J. Electron. Electr. Eng.*, Aug. 2024, doi: 10.37256/jeee.3220244886.

[5] Y. O. M. Sekyere, F. B. Effah, and P. Y. Okyere, "Optimal Tuning of PID Controllers for LFC in Renewable Energy Source Integrated Power Systems Using an Improved PSO," *J. Electron. Electr. Eng.*, vol. 3, no. 1, pp. 65–83, 2024, doi: 10.37256/jeee.3120243869.

[6] R. Sen, C. Pati, S. Dutta, and R. Sen, "Comparison Between Three Tuning Methods of PID Control for High Precision Positioning Stage," *MAPAN*, vol. 30, no. 1, pp. 65–70, 2015, doi: 10.1007/s12647-014-0123-z.

[7] D. Ma, I. Boussaada, J. Chen, C. Bonnet, S. I. Niculescu, and J. Chen, "PID control design for first-order delay systems via MID pole placement: Performance vs. robustness," *Automatica*, vol. 137, p. 110102, 2022, doi: 10.1016/j.automatica.2021.110102.

[8] M. H. Ibrahim, S. P. Ang, Z. Hamid, S. S. Mohammed, and K. H. Law, "A Comparative Hybrid Optimisation Analysis of Load Frequency Control in a Single Area Power System Using Metaheuristic Algorithms and Linear Quadratic Regulator," *2022 Int. Conf. Green Energy, Comput. Sustain. Technol. GECOST 2022*, pp. 232–237, 2022, doi: 10.1109/GECOST55694.2022.10010488.

[9] H. Farshi and K. Valipour, "Hybrid PSO-GA Algorithm for Automatic Generation Control of Multi-Area Power System," *IOSR J. Electr. Electron. Eng. Ver. I*, vol. 11, no. 1, pp. 2278–1676, 2016, doi: 10.9790/1676-11111829.

[10] S. Alghamdi *et al.*, "Optimal PID Controllers for AVR Systems Using Hybrid Simulated Annealing and Gorilla Troops Optimization," *Fractal Fract.*, vol. 6, no. 11, 2022, doi: 10.3390/fractalfract6110682.

[11] R. K. Sahu, S. Panda, and S. Padhan, "A hybrid firefly algorithm and pattern search technique for automatic generation control of multi area power systems," *Int. J. Electr. Power Energy Syst.*, vol. 64, pp. 9–23, 2015, doi: 10.1016/j.ijepes.2014.07.013.

[12] W. Aribowo, Supari, and B. Suprianto, "Optimization of PID parameters for controlling DC motor based on the aquila optimizer algorithm," *Int. J. Power Electron. Drive Syst.*, vol. 13, no. 1, pp. 216–222, 2022, doi: 10.11591/ijpeds.v13.i1.pp216-222.

[13] B. Hekimoglu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, no. March, pp. 38100–38114, 2019, doi: 10.1109/ACCESS.2019.2905961.

[14] Y. O. M. Sekyere, F. B. Effah, and P. Y. Okyere, "An Enhanced Particle Swarm Optimization Algorithm via Adaptive Dynamic Inertia Weight and Acceleration Coefficients," *J. Electron. Electr. Eng.*, pp. 50–64, 2024, doi: 10.37256/jeee.3120243868.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[16] S. Sharma, R. Kapoor, and S. Dhiman, "A Novel Hybrid Metaheuristic Based on Augmented Grey Wolf Optimizer and Cuckoo Search for Global Optimization," *ICSCCC 2021 - Int. Conf. Secur. Cyber Comput. Commun.*, pp. 376–381, 2021, doi: 10.1109/ICSCCC51823.2021.9478142.

[17] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 629–640, Mar. 2010, doi: 10.1016/j.asoc.2009.08.031.

[18] N. Bilandi, H. K. Verma, and R. Dhir, "hPSO-SA: hybrid particle swarm optimization-simulated annealing algorithm for relay node selection in wireless body area networks," *Appl. Intell.*, vol. 51, no. 3, pp. 1410–1438, 2021, doi: 10.1007/s10489-020-01834-w.

[19] Y. O. M. Sekyere, P. O. Ajiboye, F. B. Effah, and B. T. Opoku, "Optimizing PID control for automatic voltage regulators using ADIWACO PSO," *Sci. African*, vol. 27, no. March, p. e02562, 2025, doi: 10.1016/j.sciaf.2025.e02562.

[20] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," *Proc. 1999 Congr. Evol. Comput. CEC 1999*, vol. 3, pp. 1945–1950, 1999, doi: 10.1109/CEC.1999.785511.

[21] C. Guimin, H. Xinbo, J. Jianyuan, and M. Zhengfeng, "Natural exponential inertia weight strategy in particle swarm optimization," *Proc. World Congr. Intell. Control Autom.*, vol. 1, no. June, pp. 3672–3675, 2006, doi: 10.1109/WCICA.2006.1713055.

[22] Y. O. M. Sekyere, F. B. Effah, and P. Y. Okyere, "Hyperbolic Tangent - Based Adaptive Inertia Weight Particle Swarm Optimization," *J. Nas. Tek. Elektro*, vol. 2, 2023, doi: 10.25077/jnte.v12n2.1095.2023.

[23] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multi-stage linearly-decreasing inertia weight," *Proc. 2009 Int. Jt. Conf. Comput. Sci. Optim. CSO 2009*, vol. 1, pp. 505–508, 2009, doi: 10.1109/CSO.2009.420.

[24] G. Yue-lin, "with Random Inertia Weight and Evolution Strategy," *Comput. Intell.*, pp. 199–203, 2007.

[25] S. Padhy, R. K. Khadanga, and S. Panda, "A modified

GWO technique based fractional order PID controller with derivative filter coefficient for AGC of power systems with plug in electric vehicles," *Int. Conf. Technol. Smart City Energy Secur. Power Smart Solut. Smart Cities, ICSESP 2018 - Proc.*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/ICSESP.2018.8376668.

[26] M. Khalilpour, N. Razmjooy, H. Hosseini, and P. Moallem, "Optimal Control of DC Motor Using Invasive Weed Optimization (IWO) Algorithm," *Majlesi Conf. Electr. Eng.* , no. August 2016, pp. 4–10, 2011, [Online]. Available: https://www.researchgate.net/profile/Navid-Razmjooy/publication/221702286_Optimal_Control_of_DC_motor_using_Invasive_Weed_Optimization_IWO_Algorithm/links/57b70f4408aec9984ff29b6d/Optimal-Control-of-DC-motor-using-Invasive-Weed-Optimization-IWO-Algorithm.

[27] I. Khanam and G. Parmar, "Application of SFS algorithm in control of DC motor and comparative analysis," *2017 4th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Comput. Electron. UPCON 2017*, vol. 2018-Janua, no. October, pp. 256–261, 2017, doi: 10.1109/UPCON.2017.8251057.

APPENDICES

Appendix 1. Some Benchmark Functions

Name	Definition	Dim	Range	f_{min}
Sphere, $f_1(x)$	$f_1(x)=\sum_{i=1}^{d} x_i^2$	30	[-100, 100]	0
Rosenbrock, $f_2(x)$	$f_2(x)=\sum_{i=1}^{d-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30, 30]	0
Step, $f_3(x)$	$f_3(x)=\sum_{i=1}^{d}(x_i+0.5)^2$	30	[-100, 100]	0
Rastrigin, $f_4(x)$	$f_4(x)=\sum_{i=1}^{d}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12, 5.12]	0
Ackley, $f_5(x)$	$f_4(x)=-20\exp\left(\sqrt[-0.2]{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right)-exp\left(\frac{1}{d}\sum_{i=1}^{d}(cos2\pi x_i)\right)+20+e$	30	[-32, 32]	0
Griewank, $f_6(x)$	$f_5(x)=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600, 600]	0
Rotated hyper ellipsoid, $f_7(x)$	$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$	30	[-65.536,65.536]	0
Sum of Squares $f_8(x)$	$f(x)=\sum_{i=1}^{n}i.x_i^2$	30	[-100, 100]	0
Coville, $f_9(x)$	$f(x)\ 100(x_2-x_1^2)^2+(1-x_1)^2+90(x_4-x_3^2)^2+(1-x_3)^2+10.1[(x_2-1)^2\ (x_4-1)^2]+19.8(x_2-1)(x_4-1)$	4	[-100, 100]	0
Dekerarts, $f_{10}(x)$	$(x)=10^5x_1^2+x_2^2-(x_1^2+x_2^2)^2+10^{-5}(x_1^2+x_2^2)^4$	2	[-100, 100]	0
Salmonon, $f_{11}(x)$	$f(x)=1-cos\left(\sqrt[2\pi]{\sum_{i=1}^{n}x_1^x}\right)+0.1.\sqrt{\sum_{i=1}^{n}x_i^2}$	30	[-100, 100]	0
Schaffer, $f_{12}(x)$	$f(x)=0.5+\frac{sin^2(x_1^2-x_2^2)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$ $f(x)=0.5+\frac{cos^2(\sin(\vert x_1^2-x_2^2\vert))-0.5}{(1+0.0001(x_1^2+x_2^2))^2}$	30	[-100, 100]	0
Matyas, $f_{13}(x)$	$f(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	2	[-100, 100]	0
Levy, $f_{14}(x)$	$f(x)=sin^2(\pi\omega_1)+\sum_{i=1}^{n-1}(\omega_1-1)^2\left(1+10sin^2(\pi\omega_1+1)\right)+(\omega_1-1)^2\left(1+sin^2(2\pi\omega_n)\right)$. Where $\omega_1=1+\frac{x_i-1}{4}$	30	[-100, 100]	0
Zakharov, $f_{15}(x)$	$f(x)=\sum_{i=1}^{n}x_1^2+\left(\sum_{i=1}^{n}0.5.i.x_i\right)^2+\left(\sum_{i=1}^{n}0.5.i.x_i\right)^4$	30	[-100, 100]	0
Beale $f_{16}(x)$	$f(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^2)^2+(2.625-x_1+x_1x_2^3)$	2	[-100, 100]	0

Appendix 2. Comparison of Results for Various Benchmark Functions [24]

Function	**Index**	Proposed NExIWAC	ADIWACO PSO [14]	Tanh-PSO [22]	LDIW-PSO [23]	RIW-PSO [24]	SPSO [15]
$f_1(x)$	Mean	6.5905e-217	1.8744e-182	0.00016	1.1321e-71	0.010212	0.00016
	SD	2.7277e-41	0	5.9791e-74	6.67E-18	0.005073	0.000119
$f_2(x)$	Mean	2.1743e-29	1059.101	134.3922	28.1473	97.11795	134.3922
	SD	87.2534	2050.492	128.5852	11.29836	128.7488	128.5852
$f_3(x)$	Mean	0	0	0.133333	0	0	0.133333
	SD	0	0	0.345746	0	0	0.345746
$f_4(x)$	Mean	20.0276	113.4578	29.31833	15.05704	12.52966	6.2225
	SD	1.7907	5.1185e-13	6.879953	4.439862	2.915823	6.879953
$f_5(x)$	Mean	1.678	0.343794	0.007432	3.96E-09	0.021188	0.007432
	SD	1.4385e-14	0.447427	0.014159	3.96E-10	0.004548	0.014159
$f_6(x)$	Mean	0	3.3428	0.022795	4.4727221	0.018359	0.022795
	SD	0	2.6659e-15	0.02737	2.048563	0.009772	0.02737
$f_7(x)$	Mean	2.2264e-05	0.0133323	1.68638E-06	0.0074484	0.0485892	555.262675
	SD	1.34E-04	0.0122052	3.37275E-06	0.00034613	0.08535358	375.4684072
$f_8(x)$	Mean	3.09e-06	839.8906	8.62785e-10	0.005521	0.10933	703.688625
	SD	3.11e-06	277.4305731	1.72474e-09	0.001121	0.177171671	743.5158272
$f_9(x)$	Mean	8.8552e-04	0.582273733	0.00739473	2.45136e-07	0.008161075	241851.6587
	SD	5.81e-05	0.600416304	0.01226602	4.8758e-07	0.006606125	399841.4084
$f_{10}(x)$	Mean	0	1.2287e-06	0	0	0.001849	0.07010025
	SD	0	1.41917e-06	0	0	0.003698	0.040982627
$f_{11}(x)$	Mean	0.689457	7.0999	1.209345	2.0999	1.774975	25.445225
	SD	0.101739	0.4582	0.343362774	1.166190379	0.263081146	2.267575005
$f_{12}(x)$	Mean	0	8.49000e-08	0	0	0	0.011937147
	SD	0	1.47028e-07	0	0	0	0.016750457
$f_{13}(x)$	Mean	0	6.4112e-11	9.166e-145	0	2.50507e-87	0.3308045
	SD	0	1.10845e-10	1.8332e-144	0	4.98452e-87	0.288137962
$f_{14}(x)$	Mean	0.000833174	6731.075	643.17445	47.037125	1504.600525	18834.94093
	SD	0.000806	6109.314128	860.7678384	43.44563447	1917.960562	3003.901595
$f_{15}(x)$	Mean	2336.591	135172.2	129914.3	7011.831575	74566.535	272748.22
	SD	1957.295336	35016.53085	111216.3696	5402.708432	4958997.0136	140539.57
$f_{16}(x)$	Mean	0	1.998e-07	0	0	0	3.18
	SD	**0**	2.33805e-07	0	0	0	2.21

https://doi.org/10.25077/jnte.v14n3.1401.2025