Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Optimal Control of Brushless Dc Motor using Soft Computing Technique Particle Swarm Optimisation and Grasshopper

[1]Jayesh Rajaram Dhuri, [2]Dr. E. Vijay Kumar
[1]PhD Research Scholar Department of Electrical Engineering RKDF IST, SRK University Bhopal, M.P., India jayeshdhuri87@gmail.com
[2]Professor & HOD Department of Electrical Engineering RKDF IST, SRK University Bhopal, M.P., India
rvijaykumareda03@gmail.com

ARTICLE INFO	ABSTRACT
Received: 18 Dec 2024 Revised: 10 Feb 2025 Accepted: 28 Feb 2025	The research paper introduces an optimization approach that combines Grass Hopper Optimization (GOA) and PSO techniques. Determining the optimal PID controller parameters to regulate a BLDC motor's speed is the aim. The Simulink environment in MATLAB software is employed to simulate the BLDC motor. Optimally tuned PID controllers are then employed to select an optimal PI controller for BLDC motor. Lastly, the efficacy of this proposed optimization method is validated through various simulations along with experimental results obtained from the BLDC motor control. **Keywords:** BLDC, Grasshopper algorithm, particle swarm optimization, Matlab simulation, PID tuning

I. INTRODUCTION

In this study, PSO (Particle Swarm Optimization) and GOA (Grasshopper Optimization Algorithm) are employed to tune a PID controller for BLDC motor speed regulation. Simulation results based on a BLDC motor model confirm that the proposed control strategy effectively identifies the optimal PID gain parameters. Moreover, a comparative analysis between the two optimization techniques indicates that GOA provides enhanced dynamic performance of the overall system.

A variety of control methods—including optimum, variable structure, adaptive, and nonlinear strategies—have been widely reported for BLDC motor speed regulation [1]. Nonetheless, many of these approaches are either challenging to implement in practice or involve intricate theoretical frameworks [2]. In contrast, the PID controller, with its 3 part structure addressing both transient along with steady-state responses, remains one of the most straightforward and effective solutions for many real-world control challenges [3].

It is well recognized that PID controllers significantly improve the dynamic performance of controlled systems; however, the selection of optimal gains is inherently an optimization problem [9]. In recent years, techniques for example, the FA (Firefly Algorithm) and other competitive optimization algorithms are successfully applied to tune PID parameters across various applications in electrical engineering and beyond [10]. A particularly notable application is in the control of BLDC motors. Evolved from brushed DC motors, BLDC motors are synchronous machines powered by DC voltage and utilize solid-state switching for current commutation, with rotor positions typically monitored via Hall sensors [11].

BLDC motors possess several advantages, including high efficiency, extended operational life, compactness, low noise, and superior speed–torque characteristics. These benefits have driven significant advancements in aerospace, automotive, and numerous other engineering fields. However, the performance of these motors can be compromised by varying load conditions, a challenge that traditional control methods may struggle to address. This limitation has spurred the development of advanced control strategies, particularly those on the basis of artificial intelligence (AI). Techniques for example fuzzy control [12, 13], the Improved or Modified Firefly Algorithm (IFA/MFA) [4–8], neural control [14, 15], PSO-based control [38], Genetic Algorithm (GA) control [16, 17], as well as more recent approaches

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

like FA and BAT control [11] have been proposed. These methods largely focus on optimizing the PID controller's settings and derivatives to achieve superior performance under a range of operating conditions.

In this article, a Brushless DC (BLDC) motor controller design technique that relies on the GOA is presented, and its performance is contrasted with that of PSO. Optimization of the PID controller settings for the BLDC motor control is conducted employing MATLAB simulations, in which parameters kpk_p, kik_i, along with kdk_d are systematically tuned to assess their impact on motor performance. MATLAB's robust simulation tools facilitate detailed analysis and precise fine-tuning to attain optimal control efficiency. Furthermore, performance of both GOA and PSO algorithms is evaluated against standard indices, namely the IAE (Integral of Absolute Error), ISTE (Integral of the Squared Time Error), ITAE (Integral of Time-weighted Absolute Error), as well as ISE (Integral of Squared Error). Comparative results indicate that the GOA method outperforms PSO in optimizing the PID controller settings, underscoring its effectiveness for this application.

The rest of the article has been structured as subsequent. Section 2 introduces the BLDC motor system's mathematical model, laying the groundwork for subsequent controller design. Section 3 describes in detail the PSO along with GOA algorithms developed for optimal value determination for the kpk_p, kik_i, and kdk_d parameters. In Section 4, the MATLAB Simulink model, constructed based on the methods presented in Section 3, is discussed alongside the corresponding simulation results and findings. The final section offers a comparative analysis of the PSO and GOA approaches, demonstrating the superior performance of the GOA, and concludes with suggestions for future research.

II. BLDC MOTOR MODELLING

For the purpose of analysis and modeling, it is assumed that the stator winding parameters are identical and remain constant across all phases. This assumption simplifies the representation of the motor and facilitates a clearer comparison of its performance characteristics, as depicted in Figure 1.

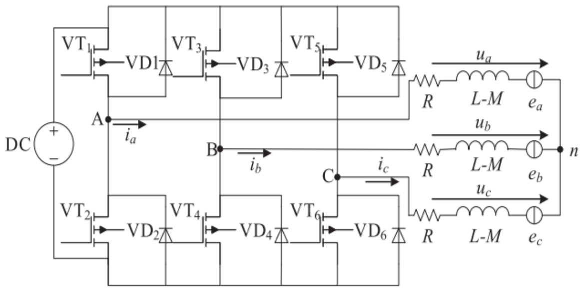

Fig. 1. BLDCM Circuit Diagram

The voltage equations for a Brushless DC Motor (BLDCM) can be formulated as follows:

$$u_a = Ri_a + (L - M)\frac{di_a}{dt} + e_a, \quad (1)$$

$$u_b = Ri_b + (L - M)\frac{di_b}{dt} + e_b, \quad (2)$$

$$u_c = Ri_c + (L - M)\frac{di_c}{dt} + e_c, \quad (3)$$

Where, i_x, e_x and u_x (x = a, b, c) indicate, respectively, the current, e.m.f, and voltage. Of three-phase windings; L and R signify the self-inductance and resistance of each phase's windings; M denotes the mutual inductance between any two windings; and n denotes the electric potential reference point. Because LM evenly represents the terms of LM in the preceding equations, the voltage equations of BLDCM may be rewritten as

$$u_a = Ri_a + L_M\frac{di_a}{dt} + e_a, \quad (4)$$

$$u_b = Ri_b + L_M\frac{di_b}{dt} + e_b, \quad (5)$$

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$u_c = Ri_c + L_M \frac{di_c}{dt} + e_c, \quad (6)$$

III. OPTIMIZATION Algorithm

Optimization is crucial for finding the best design based on specific criteria or constraints. It can be categorized into single-objective or multi-objective optimization, depending on the problem. Initially, random values are generated for the given situation. Nature-inspired swarm algorithms are widely used in stochastic optimization. Inspired by natural processes, many algorithms exist for single and multiple-solution-based optimization. Examples include ACO, PSO, and GA. The GOA is considered one of the most effective techniques in artificial intelligence.

A. Grass Hopper Algorithm

Grasshoppers are widely recognized as agricultural pests due to their significant impact on crops. They may appear either as solitary individuals or in large groups during both their nymph and adult stages. These insects consume a considerable amount of the vegetation in their path, which can lead to substantial crop damage. Notably, while nymph swarms tend to move slowly, adult swarms are capable of rapid, abrupt movement over long distances. Drawing inspiration from such natural behaviors, nature-inspired algorithms are designed to balance exploration along with exploitation. Particularly, GOA is based on the food-seeking behavior observed in grasshoppers. The following mathematical model effectively simulates this swarming behavior::

$$X_i = S_i + G_i + A_i \quad (7)$$

Where,

A_i - Wind advection

X_i – i^{th} grasshopper's position

G_i – Gravity force on i^{th} grasshopper

S_i – Social interaction

For the random behavior of grasshoppers, this equation can be manipulated as below:

$$X_i = r_1 S_i + r_2 G_i + r_3 A_i \quad (8)$$

Here, $r_{1,}, r_2$, r_3 are random numbers in the range (0,1). The S_i can be given by equation (3):

$$S_i = \sum_{j=1\ j\neq i}^{N} s(d_{ij})\overrightarrow{d_{ij}} \quad (9)$$

Here, distance between i^{th} and j^{th} grasshopper is provided by d_{ij} that given as follows:

$$d_{ij} = |X_j - X_i| \quad (10)$$

S factor in the above equation provides strength of the social forces and $\overrightarrow{d_{ij}}$ is unit vector from the i^{th} grasshopper to j^{th} grasshopper :

$$\overrightarrow{d_{ij}} = \frac{x_j - x_i}{d_{ij}} \quad (11)$$

And s can be given by equation (12):

$$s(r) = f\,.e^{\frac{-r}{l}} - e^{-r} \quad (12)$$

In this context, l denotes attractive length scale, while f denotes intensity of attraction. Functions illustrate the degree of attraction or repulsion experienced by the grasshoppers.

Component G in equation (13) can be given as:

$$G_i = -g\,.\overrightarrow{e_g} \quad (13)$$

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

In this context, $\overrightarrow{e_g}$ denotes the unit vector pointing towards the Earth's center whereas g represents gravitational constant. Gravitational constant g quantifies the force of gravity exerted by the Earth, while the unit vector $\overrightarrow{e_g}$ provides a directional reference, indicating the direction of the gravitational pull towards the Earth's core.

Factor A can be calculated as:

$$A_i = u \cdot \overrightarrow{e_w} \qquad (14)$$

$\overrightarrow{e_w}$ is a wind direction unity vector

u is drift constant

Nymph grasshoppers are wingless, which significantly influences their movement patterns. As a result, the direction and speed of their movement are closely linked to the direction of the wind. This relationship can be mathematically described using equation (15). This dependency on wind direction performs a vital role in understanding and modeling the nymph grasshoppers' behavior.

$$X_i = \sum_{j=1\, j\neq i}^{N} s\left(d_{ij}\right)\overrightarrow{d_{ij}} - g \cdot \overrightarrow{e_g} + u \cdot \overrightarrow{e_w} \qquad (15)$$

IIn this context, NN denotes the total number of grasshoppers within the swarm. The equation under consideration models the interactions among these individuals. However, applying a direct mathematical formulation to grasshopper swarms presents challenges, as the grasshoppers tend to quickly settle into a state of equilibrium, thereby failing to adequately reach certain target points. To mitigate this limitation, a modified version of the original Equation (15) is introduced, represented as Equation (16). This updated formulation refines the model's capacity to accurately simulate grasshopper behavior and enhances the ability of the algorithm to comprehensively explore search space.

$$X_d^i = \left(\sum_{j=1\, j\neq i}^{N} c\, \frac{U_{bd} - L_{bd}}{2} \cdot s \cdot \left(\left|X_j^d - X_i^d\right|\right)\frac{X_j - X_i}{d_d}\right) + T_d \qquad (16)$$

Where, U_{bd} - upper bound in Dth dimension

T_d - solution best

L_{bd} - Dth dimension Lower bound

C - comfort, repulsion, and attraction zone decreasing coefficient.

GOA advantages include: (i) finding promising regions in a given space, (ii) accurate solutions for unconstrained optimization, (iii) constrained optimization promising results, and (iv) obtaining global and local optima.

The GOA process (Figure 4) involves initialization, random population generation, fitness evaluation, normalizing distances between grasshoppers, updating positions, iterating for all population members, updating the best solution, and repeating until termination criteria are met.

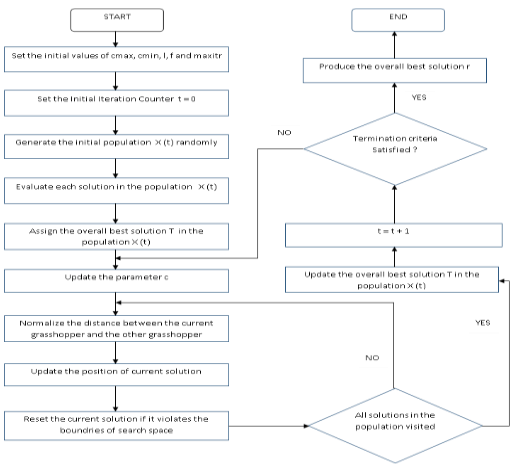

Fig. 2: Block Diagram of GRASSHOPPER ALGORITHM

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Optimizing the PID controller's tuning parameters is the aim of this study, specifically kdk_d, kpk_p, and kik_i, by employing the GOA. Optimized PID controller is then used as part of a control strategy for regulating the BLDC motor's speed. Typically, motor's speed deviates from its desired setpoint; hence, the PID controller has been implemented to mitigate these deviations as well as stabilize the motor's speed in line with the required specifications.

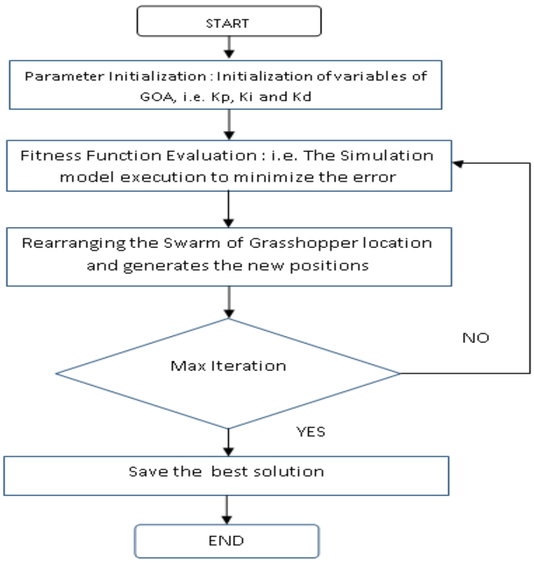

Fig. 3: PID optimization using GOA

Figure 3 shows a flowchart of the optimization process for tuning parameters kd, kp, and ki of the PID controller. The goal is to find values yielding a stable delta p (difference between desired and actual motor speeds) with minimal overshoot. GOA iteratively adjusts PID gains, converging on optimal values. These ensure the BLDC motor's speed closely follows the setpoint with a stable response, minimal oscillations, and overshoots, enhancing overall speed control system performance and efficiency.

B. Particle Swarm Optimisation Algorithm

A population-based optimization method is PSO that inspired by collective behaviors observed in bird flocking as well as fish schooling. Initially developed by Kennedy and Eberhart in 1995 [1], the PSO has become a widely adopted technique for addressing complex optimization challenges. With this method, the search space is traversed by a swarm of particles, all of which stand for a possible solution. These particles update their positions as they move through space, taking into account both their personal experiences and the swarm's shared knowledge.

Each particle maintains an adjustable velocity and retains a record of its personal best position along with the best position discovered throughout the entire swarm. In a multidimensional problem space, every particle has a unique position and velocity that describe its location and movement. As the particles navigate the space, they continuously monitor and update their personal best positions. The overall update mechanism is governed by Equations (17) and (18) in the original document. These equations incorporate an inertia component that influences the particle's momentum, a cognitive component that reflects particle's own experience, and a social component that accounts for swarm's collective wisdom. Additionally, random factors are introduced to maintain diversity in the search process.

This balanced interplay between exploration and exploitation, as mathematically described by Equations (17) and (18), allows PSO to effectively converge toward optimal solutions, ensuring that the particles are guided both by their personal discoveries and by the overall progress of the swarm [1].

The parameters c1 and c2 play a crucial role in guiding the particles' movement. These coefficients influence how strongly particles are drawn towards their personal best positions and the swarm's global best position,

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

correspondingly. When these values are set low, particles tend to wander more freely through the search space, thoroughly exploring different areas before focusing on promising regions. Conversely, higher values of c1 and c2 cause particles to make more abrupt shifts, potentially overshooting or quickly converging on target areas. The algorithm's ability to find the best solutions depends on this delicate balance between exploration and exploitation. It is typically recommended to set c1 and c2 to 2.0, according to sources [17, 18].

$$v_{id}^{n+1} = w_i v_{id}^{n} + c_1 r_1^{n}(p_{id}^{n} - x_{id}^{n}) + c_2 r_2^{n}(p_{gd}^{n} - x_{id}^{n}) \quad (17)$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1} \quad (18)$$

$$w_i = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{k_{\max}}\right) \times k \quad (19)$$

IV. SIMULATION AND RESULT

This section focuses on the MATLAB/Simulink implementation of the BLDC motor model, incorporating two distinct controllers: PSO-PID controller and GOA-PID controller. Primary aim of these simulations is to assess and compare the BLDC motor's speed performance control system when subjected to different controllers. By conducting these simulations, researchers can gain insights into how each controller influences the system's behavior, efficiency, and overall effectiveness in maintaining the desired BLDC motor speed. This evaluation is crucial for determining the most suitable controller for optimal motor performance under various operating conditions.

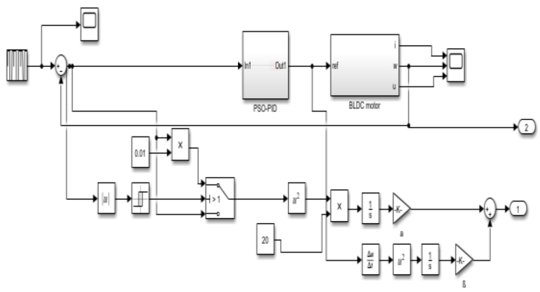

Fig. 4. BLDC motor simulation model using PSO-PID controller

Figure 4 illustrates the Simulink model developed for BLDC motor's speed control system employing the PSO-PID controller. This model encapsulates BLDC motor dynamics, PSO algorithm for PID controller parameters optimization, and closed-loop control architecture. the PID controller's tuning parameters (Kp, Ki, Kd) are adjusted iteratively through PSO algorithm for minimizing the error between desired along with actual motor speeds, thereby optimizing speed control performance.

However, Figure 5 illustrates BLDC motor simulink moodel speed control system employing the GOA-PID controller. Similar to PSO-PID model, this simulation incorporates the BLDC motor dynamics and a closed-loop control structure. However, instead of the PSO algorithm, the Grass Hopper Optimization (GOA) has been utilized to tune and optimize the parameters of PID controller.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

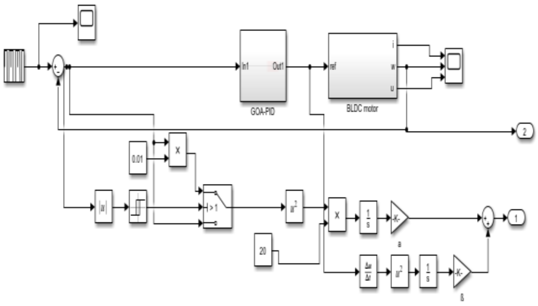

Fig. 5. Simulink Model of BLDC motor using GOA-PID controller

The GOA algorithm emulates natural behavior of grasshoppers in nature, exploring the search space and converging towards the optimal Kp, Ki, and Kd values that diminish the speed error as well as enhance the system's dynamic response.

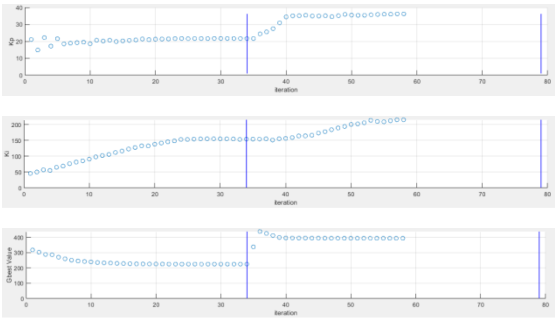

Figure 6: Kp, Ki and Gbest values Nature in each iteration

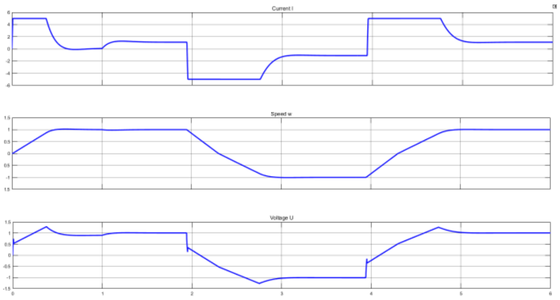

Fig. 7. Speed, current, and voltage variation of BLDC motor

Simulation outcomes, including the motor speed response, steady-state error, settling time, along with overshoot can be assessed to determine the most suitable controller architecture for the given application.

V. CONCLUSION

This study presents a comparative evaluation of PSO and the GOA for tuning PID controllers in BLDC motor speed control systems. Simulation experiments using a BLDC motor model demonstrate that both methods effectively optimize proportional, integral, along with derivative gains of the PID controller. However, the GOA—drawing inspiration from the swarming behavior of grasshoppers—outperforms PSO by significantly enhancing the dynamic responsiveness of the closed-loop system. Notably, the GOA approach yields considerable improvements in transient performance, including reduced settling time, diminished overshoot, as well as increased steady-state accuracy in tracking motor speed setpoints. These outcomes highlight the potential of GOA as a more efficient strategy for PID controller tuning in motor control applications.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

REFERENCES

[1] Nasri M, Nezamabadi-pour H, Maghfoori M. A PSO-based optimum design of PID controller for a linear brushless DC motor. World Acad Sci, Eng Technol 2007, 2, 211–5.
[2] C.L.Lin, H.Y.Jan, Evolutionarily multiobjective PID control for linear brushless DC motor, in: Proc. of IEEE Int. Conf .Industrial Elect. Society 3, Nov. 2002, pp. 2033-2038.
[3] Ang K, Chong G, Li Y. PID control system analysis, design, and technology. IEEE Trans Control System Technology 2005, 13-July, 559–76.
[4] Boris, J.; Paolo, L.; Guido, M. Control of double-loop permanent magnet synchronous motor drives by optimized fractional and distributed-order PID controllers. Eur. J. Control 2021, 58, 232–244.
[5] Mat Hussin, A.T.; Intan Zaurah, M.D.; Pakharuddin, M.S.; Hanim, M.Y.; Mohd Ibthisham, A.; Nik Mohd, R.S.; Muhamad, S.H. Vibration control of semi-active suspension system using PID controller with advanced firefly algorithm and particle swarm optimization. J. Ambient. Intell. Humaniz. Comput. 2021, 12, 1119–1137.
[6] He, S.; Chen, Z.; Gao, X. Parameter Solving of DC Servo Motor PID Controller Based on Improved Firefly Algorithm. In Proceedings of the 2018 3rd International Conference on Robotics and Automation Engineering (ICRAE), Guangzhou, China, 17–19 November 2018; pp. 136–140.
[7] Rosy, P.; Santosh, K.M.; Jatin, K.P.; Bibhuti, B.P. Optimal fractional order PID controller design using Ant Lion Optimizer. Ain Shams Eng. J. 2020, 11, 281–291.
[8] Devarapalli, R.; Naga Lakshmi, N.J.; Prasad, U. Application of a Novel Political Optimization in Optimal Parameter Design of PI Controller for the BLDC Motor Speed Control. In Proceedings of the 2020 International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), Patna, India, 10–11 July 2020; pp. 1–6.
[9] BARAN, H. Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm. IEEE Access 2019, 7, 38100–38114.
[10] Jagatheesan, K.; Anand, B.; Samanta, S.; Dey, N.; Ashour, A.S.; Balas, V.E. Design of a Proportional-Integral-Derivative Controller for an Automatic Generation Control of Multi-Area Power Thermal Systems Using Firefly Algorithm. IEEE/CAA J. Autom. Sin. 2019, 6, 503–515.
[11] Chaib, L.; Choucha, A.; Arif, S. Optimal Design and Tuning of Novel Fractional Order PID Power System Stabilizer Using a New Metaheuristic Bat Algorithm. Ain Shams Eng. J. 2017, 8, 113–125.
[12] Xiaoyan, P.; Mingfei, J.; Lei, H.; Xiang, Y.; Yibin, L. Fuzzy sliding mode control based on longitudinal force estimation for electro-mechanical braking systems using BLDC motor. CES Trans. Electr. Mach. Syst. 2018, 2, 142–151.
[13] Walekar, V.R.; Murkute, S.V. Speed Control of BLDC Motor Using PI Fuzzy Approach: A Comparative Study. In Proceedings of the 2018 International Conference on Information, Communication, Engineering and Technology (ICICET), Pune, India, 29–31 August 2018; pp. 1–4.
[14] Ahmed, R.; Paul, Y. Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives. IEEE Trans. Ind. Appl. 2016, 52, 414–424.
[15] Xiong, S.; Junguo, G.; Jian, C.; Biao, J. Research on Speed Control System of Brushless DC Motor Based on Neural Network. In Proceedings of the 2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), Nanchang, China, 14–15 June 2015; pp. 761–764.
[16] Putra, E.H.; Zulfatman, Z.; Effendy, M. Robust Adaptive Sliding Mode Control Design with Genetic Algorithm for Brushless DC Motor. In Proceedings of the International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Malang, Indonesia, 16–18 October 2018; Volume 5, pp. 330–335, ISBN 978-1-5386-8402-3.
[17] Li, J.; Li, W. On-Line PID Parameters Optimization Control for Wind Power Generation System Based on Genetic Algorithm. IEEE Access 2020, 8, 137094–137100.
[18] Merugumalla, M.K.; Navuri, P.K. PSO and Firefly Algorithms Based Control of BLDC Motor Drive. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 994–999.
[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. IV, pp. 1942-1948, 1995.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[20] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in Proc. IEEE Int. Conf. Evol. Comput., pp. 69-73, 1998.

[21] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in Proc. IEEE Int. Conf. Evol. Comput., pp. 1945-1950, 1999