International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 56, No. 3, June, 2023, pp. 475-482
Journal homepage: http://iieta.org/journals/jesa

Comparative Performance Evaluation of Swarm Intelligence-Based FOPID Controllers for PMSM Speed Control

Hussain Ali Azzawi[1*], Nihad M. Ameen[2], Sabah A. Gitaffa[1]

[1] Electrical Engineering Department, University of Technology, Baghdad 10066, Iraq
[2] Communication Engineering Department, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: eee.21.12@grad.uotechnology.edu.iq

https://doi.org/10.18280/jesa.560315

Received: 10 May 2023
Accepted: 11 June 2023

Keywords:
fractional-order proportional-integral-derivative (FOPID), proportional-integral-derivative (PID), permanent magnet synchronous motors (PMSMs), mGrey wolf optimization (GWO), ant colony optimization (ACO), particle swarm optimization (PSO)

ABSTRACT

This paper presents a study focused on the design and performance evaluation of Fractional-Order Proportional-Integral-Derivative (FOPID) controllers for the speed control of Permanent Magnet Synchronous Motors (PMSMs). Effective speed control of PMSMs is of great importance in various applications such as robotics, electric vehicles, and industrial automation. However, achieving precise and efficient speed control poses several challenges due to the nonlinear and time-varying nature of PMSMs. To address these challenges, the study proposes the utilization of FOPID controllers, which offer advantages over traditional PID controllers, including improved robustness and greater flexibility in handling complex system dynamics. Additionally, the study explores the use of Swarm Intelligence (S.I.) algorithms for the design and tuning of FOPID controllers. Swarm Intelligence algorithms, such as Particle Swarm Optimization (PSO), ant colony optimization (ACO), and Grey Wolf Optimization (GWO), are known for their ability to effectively search and optimize complex parameter spaces. The main contribution of this work is the comparison and evaluation of PSO, GWO, and ACO algorithms for the design of FOPID controllers in PMSM speed control applications. The controllers are assessed through both simulations and experimental tests to analyze their performance in terms of speed-tracking accuracy, overshoot, and settling time. The key finding of the study is that the ACO-FOPID controller exhibits the best performance in terms of transient response. It achieves a rise time of 0.008978 s, a settling time of 0.01 s, and zero absolute time error (ITAE). These results indicate that the ACO-FOPID controller provides precise and fast speed control for PMSMs, making it a promising solution for practical applications. In summary, this study highlights the importance of PMSM speed control and the challenges associated with it. It introduces the FOPID controller as a potential solution and motivates the utilization of Swarm Intelligence algorithms for its design. The comparison of PSO, GWO, and ACO algorithms for FOPID controller design demonstrates the superiority of the ACO-FOPID controller in terms of transient response. This research contributes to the advancement of control systems for PMSMs and showcases the potential of Swarm Intelligence algorithms in optimizing complex control parameters.

1. INTRODUCTION

The Permanent Magnet Synchronous Motor (PMSM) is a widely used electric motor in various industrial applications due to its high efficiency, reliability, and precise speed control capabilities [1]. Achieving accurate speed control is essential for optimal motor performance and system efficiency. To address this requirement, control methods such as Proportional-Integral-Derivative (PID) controllers have been commonly employed [2]. However, the inherent limitations of PID controllers in dealing with nonlinear and time-varying systems have led to the development of more advanced control techniques, such as Fractional-Order Proportional-Integral-Derivative (FOPID) controllers, which offer higher precision and robustness [3, 4]. FOPID controllers have emerged as a promising solution for achieving accurate speed control in PMSMs. By incorporating fractional-order calculus into the controller design, FOPID controllers can effectively handle the complex dynamics of PMSMs, making them well-suited for industrial applications. Swarm Intelligence (SI) algorithms provide a powerful optimization approach for tuning the control parameters of FOPID controllers. Inspired by the collective behavior of natural systems such as birds flocking, ant colonies, or schools of fish, SI algorithms mimic these behaviors to search for optimal solutions in complex optimization problems. The use of SI algorithms in FOPID controller design for PMSM speed control offers the potential to enhance system performance and robustness.

In the field of PMSM speed control, several studies have explored the use of different optimization algorithms and controller designs. These studies have focused on techniques such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), FireFly Algorithm (F.A.), Grey Wolf Optimization (GWO), Genetic Algorithm (G.A.), and Nelder-Mead (NM). Here is a summary of the main findings from the cited studies:

(1) Belkhadir et al. [5] proposed the use of the ACO algorithm for self-tuning controller parameters, aiming to improve system efficiency and stability.

(2) Agarwal et al. [6] utilized ACO to calculate gain coefficients for a PID speed controller, showing improved dynamic performance compared to traditional PID tuning methods.

(3) Chiranjeevi et al. [7] compared the performance of FPA-based PID and FOPID controllers with FireFly Algorithm (F.A.) and PSO methods. FOPID controllers outperformed PID controllers in terms of setup time and oscillation amplitude, with FPA exhibiting better performance than F.A. and PSO in terms of ISE-based responses.

(4) Yadav and Verma [8] compared the Ziegler-Nichols (Z-N) method with the PSO technique for controlling PMSM speed. They found that the PSO-based PID speed controller improved motor performance across different operating conditions.

(5) El-Saadawi et al. [9] introduced the Gray Wolf Optimization (GWO/FOPID) method for speed control of a D.C. motor. The Nelder-Mead (NM) algorithm-based FOPID approach yielded the best results, achieving zero overshoot with low settling and rising times.

(6) Szczepanski et al. [10] investigated the Model Reference Adaptation System (MRAS) for PMSM speed control. Their results showed that the adaptive ABC-based SFC (State Feedback Control) improved performance compared to non-adaptive conditions.

(7) Ibrahim et al. [11] compared the effectiveness of FOPID controllers with other techniques, such as PSO and ACO algorithms. The PSO-FOPID approach outperformed ACO-FOPID and ACO-PID controllers in terms of transient and frequency response, although it exhibited a larger maximum overshoot.

The motivation behind this study is to investigate and compare the performance of different SI algorithms in tuning FOPID controllers for accurate speed control of PMSMs. The goal is to determine the best controller design that can overcome the limitations of traditional PID controllers and achieve optimal speed control in PMSMs.

By evaluating the performance of FOPID controllers with SI algorithms such as Particle Swarm Optimization (PSO), ant colony optimization (ACO), and Grey Wolf Optimization (GWO), we can identify the most effective approach for achieving accurate speed control in PMSMs. The study aims to improve system efficiency, reliability, and overall motor performance, thereby contributing to advancements in industrial automation and energy efficiency.

The format of this essay is as follows: The fractional order controller is shown in Section 2. The Mathematical Modeling of the PMSM Motor is the subject of Section 3. A summary of optimization methods and various metaheuristic PSO, GWO and ACO algorithms are provided in Section 4. Results from the tests are applied in Section 5. then highlights the conclusions of the proposed system.

2. FRACTION ORDER CONTROLLER

A Fractional Order Controller (FOC) is a type of controller used in control systems that incorporates fractional calculus principles in its design. Unlike traditional controllers, which utilize integer order differential and integral operations, FOCs employ fractional order derivatives and integrals.

In mathematical terms, a fractional order controller utilizes non-integer orders, such as fractional values to define the control action. These fractional orders allow the controller to capture the dynamics and memory effects of the controlled system more accurately.

The key advantage of using FOCs lies in their ability to handle systems with non-linearities, time delays, and complex dynamics more effectively. By incorporating fractional calculus, FOCs can better capture the behavior of these systems and provide improved control performance.

FOCs offer several benefits over traditional integer order controllers. They exhibit increased flexibility, adaptability, and robustness, making them suitable for controlling complex processes and systems. FOCs can provide better tracking of setpoints, rejection of disturbances, and stability in the presence of uncertainties.

The design and implementation of FOCs require specialized mathematical tools and algorithms that deal with fractional calculus operations. These tools enable the analysis and synthesis of FOCs for specific control applications.

FOCs find applications in various fields, including robotics, chemical processes, power systems, biomedical systems, and many others. They offer an alternative approach to control system design, allowing for improved control performance in challenging and dynamic environments.

In summary, a Fractional Order Controller (FOC) is a controller that utilizes fractional calculus principles and non-integer orders to achieve improved control performance, flexibility, and adaptability in controlling complex systems [12, 13]. The Laplace transform of a PID controller can be written as in Eq. (1).

$$G_c(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

where, Kp: proportional gain; G_c: transfer function of PID controller; K_i: integral gain; K_d: derivative gain.

The FOPID Eq. (2) describes the controller's generalized transfer function [14].

$$G_c(s) = \frac{u(s)}{e(s)} = K_p + K_i\left(s^{-\lambda}\right) + K_d(s^{\mu}) \tag{2}$$

where, $G_c(s)$: FOPID transfer function; $u(s)$: Controller output. $e(s)$: An error has been produced; K_p, K_i, and K_d: are the earnings equitably distributed, integral and derivative terms, respectively; μ: Fractional portion of the derivative part; λ: fractional component of the integral term.

Figure 1 shows the block diagram of the PID controller. Figure 2 shows the block diagram of the FOPID controller.

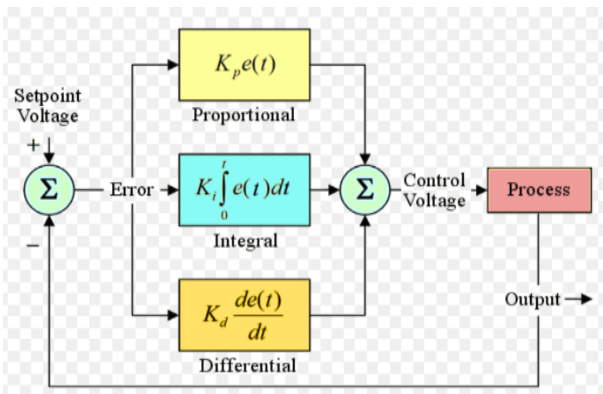

Figure 1. PID controller block diagram

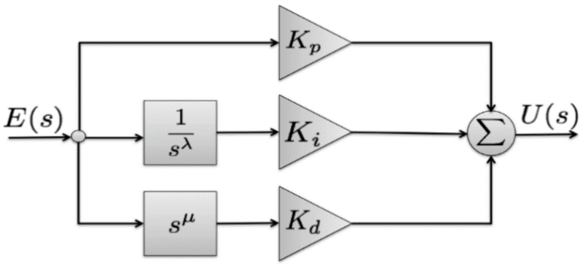

Figure 2. FOPID controller block diagram

3. MATHEMATICAL MODELING OF PMSM MOTOR

A Permanent Magnet Synchronous Motor (PMSM) is a type of electrical motor in which a permanent magnet serves as the magnetic field source for the stator. These motors offer a high torque/inertia ratio, a high-power density, a high efficiency, are durable, and are simple to maintain [14]. Following is a derivation of the modelling of PMSM that was done to establish the transfer function between output and input [15].

$$V_d = \mathrm{R}_s i_d + \mathrm{p}\lambda_d - w_e \lambda_q \tag{3}$$

$$V_q = \mathrm{R}_s i_q + \mathrm{p}\lambda\mathrm{q} + w_e \lambda_d \tag{4}$$

$$T_e = \left(\frac{3}{2}\right) \mathrm{P}\left(\lambda_{af} i_q + \left(L_d - L_q\right) i_d i_q\right) \tag{5}$$

$$T_e = T_L + \mathrm{B}w_r + \mathrm{J\ p}\, w_r \tag{6}$$

$$\mathrm{we} = \mathrm{P}\, w_r \tag{7}$$

where,

$$\lambda\mathrm{q} = L_q i_q \tag{8}$$

$$\lambda\mathrm{d} = L_d\, i_d + \lambda_{af} \tag{9}$$

Currents along the *d* and *q* axes are denoted by i_d and i_q, whereas voltages along those axes are written as V_d and V_q. The pole pair count is represented by *P*; The inductances L_q, L_d are on the *q* and *d* axes; *p* is the derivative operator; T_L., T_e are the load and electric torques; The damping coefficient is denoted by *B*; The inertial moment is characterised by *J*, and the mutual flux or airgap flux is represented by λ_{af}.

Assume that every one of the Eqs. (4)-(5) is nonlinear and that the id variable is forced to zero by the vector-controlled behaviour of the PMSM. Eqs. (4)-(5) can be solved, as illustrated in the following paragraphs [16].

$$V_q = \mathrm{R}_s\, i_q + \mathrm{pL}_q \mathrm{i_q} + w_e\, \lambda_{\,af} \tag{10}$$

$$V_d = -w_e \mathrm{L}_q \mathrm{i_q} \tag{11}$$

$$\mathrm{Te} = (3/2)\mathrm{P}\, \lambda_{af} \mathrm{i_q} - k_t \mathrm{i_q} \tag{12}$$

Transfer Function of PMSM:

$$\frac{\mathrm{Wr(S)}}{\mathrm{Vq(S)}} = \frac{k_t}{\mathrm{(R + LqS)(JS + B)} + k_t \mathrm{P}\, \lambda_{af}} \tag{13}$$

The entire T.F is written in numerical value below [16].

$$\frac{wr(s)}{v_q(s)} = \frac{4.705S + 2.219}{S\hat{}3 + 7.504S2 + 3.36S + 2.702} \tag{14}$$

4. INTELLIGENT SWARMS

The phrase "swarm intelligente" was first used by Gerardo Beni and Jing Wang in a 1989 article. Swarm intelligence techniques are population-based stochastic methods applied to combinatorial optimization issues, which should not be mistaken for intelligent swarms. Due to their local interactions with their environment, these challenges require the creation of functional global patterns from the aggregate conduct of straightforward individuals. A meta-heuristic method for solving complicated issues is swarm intelligence [17].

4.1 Particle swarm optimization (PSO)

Particle swarm optimization is an optimization technique inspired by the behaviour of animal swarms like birds or bees. It is a population-based optimizing approach that may be used to iteratively search through the space of possible answers to discover the best solution to a problem. Particle Swarm Optimization (PSO) is a stochastic and swarm intelligence-based optimization technique. Nevertheless, the classic PSO suffers from early convergence when discovering the global optimal [18]. A "swarm" of potential solutions is initiated in a PSO algorithm, and each answer is portrayed by a "particle" in a swarm. The particles travel around the solution space, and their positions are updated depending on the location of the top-performing particles in the hive (the "global best") and their individual personal best. A set of equations governs particle movement, considering the particle's present position, global or personal best. Eqs. (15) and (16) [19, 20] do this:

$$\begin{aligned} V_{jd}(k+1) = w * V_{jd}(k) + C1 \\ * rand\left(P_{jd}(K) - X_{jd}(k)\right) + C2 \\ * rand\left(P_{jd}(K) - X_{jd}(k)\right) \end{aligned} \tag{15}$$

$$X_{jd}(k+1) = X_{jd}(k) + V_{jd}(k) \tag{16}$$

where, C_1 and C_2 are constant acceleration coefficients that have a positive value, and where the value of C_1 is greater than C_2.

The rand is based on either a random integer or a mathematical formula. function that produces a random number in the range of 0 to 1. The moment of inertia is denoted by the sign *w*, where *w* is less than one.

4.2 Grey wolf optimization (GWO)

Is a meta-heuristic algorithm inspired by the hunting behaviour of grey wolves in nature. In GWO, a pack of grey wolves cooperate to find suitable solutions to optimization problems by imitating the wolf pack's social hierarchy and hunting behaviour [21]. The algorithm works as follows:

Initialize the positions of the wolves randomly within the search space. Update the parts of the wolves based on their fitness values, which are evaluated using the objective function of the problem. Determine the leader, beta, and delta wolves, the three wolves with the highest fitness values.

Update the positions of the rest of the wolves using the following equation:

$$x_i\text{=alpha} * (x_{alpha}\text{-A*}D_{alpha})\text{+beta*}(x_{beta}\text{-A*}D_{beta})\text{+delta*}(x_{delta}\text{-A*}D_{delta}) \quad (17)$$

where, x_i is the position of the itch wolf, x_{alpha}, x_{beta}, and x_{delta} are the positions of the leader, beta, and delta wolves, respectively, A is a random vector, D_{alpha}, D_{beta}, and D_{delta} are the distances between the current wolf and the leader, beta, and delta wolves, respectively, and alpha, beta, and delta are the weights of the leader, beta, and delta wolves, respectively. The main advantage of GWO is distinguished by its rapid rate of convergence and its capacity to locate global optimum solutions to difficult, high-dimensional optimization issues. However, GWO can be sensitive to the choice of parameters, such as the number of wolves in the pack and the weights of the leader, beta, and delta wolves. Other examples of factors include the number of wolves in the pack. Therefore, tweaking the parameters with extreme precision is frequently necessary for good performance [22, 23].

4.3 Ant colony optimization (ACO)

Ant colony optimization is a meta-heuristic algorithm inspired by ants' behaviour, notably their ability to locate the shortest path between their nest and a food source. In particular, the method was motivated by the power of ants to discover the route that takes them the shortest distance from their colony to a source of food. In ACO, a set of artificial ants cooperate to find suitable solutions to optimization problems by following a pheromone, Trai. The algorithm works as follows:

Initialize a set of artificial ants at the starting position. Each chooses the next vertex to visit based on a probabilistic rule that considers the amount of pheromone on the edges and the heuristic information about the desirability of each vertex. After all the ants have completed their tour, the amount of pheromone on the edges is updated based on the quality of the solution found. The pheromone trail is updated by evaporating a certain percentage and depositing new pheromones on the edges of the best solution found so far [24]. The main advantage of ACO is its ability to find suitable solutions in complex, high-dimensional optimization problems with multiple local optima. ACO has been successfully applied to many issues, including the travelling salesperson problem, the quadratic assignment problem, and the job shop scheduling problem. However, ACO can be sensitive to the choice of parameters, such as the pheromone evaporation rate and the balance between pheromone and heuristic information in the decision rule. Therefore, carefully tuning the parameters is often required to perform well [25, 26].

5. SIMULATION RESULTS

The meta-heuristic approach known as transient response analysis can be simulated using the 64-bit version of MATLAB, version R2022b (9.13.0.2049777). The toolbox provided by FOMCON was used to achieve the results for each method to complete the FOPID. The Eq. (13). depicts the whole transfer function of the PMSM motor. Applying the parameter values presented in Table 1 to the proposed PMSM motor system in the form of the Eq. (14). will result in the production of the final PMSM motor transfer function. A simulation of the strategy that was suggested may be seen in Figure 3. It will produce the final PMSM engine transfer function. The output of the PMSM motor without an emulator controlling it can be seen in Figure 4.

Table 1. PMSM motor parameters

Parameter	Values	Units
Kt	6.807	N-m/A
λaf	1.513	V/rad/sec
Js	0.0337	Kg-m^2
R	0.12	Ω
Lq	0.764	mH
B	0.086	
P	2	pole pairs

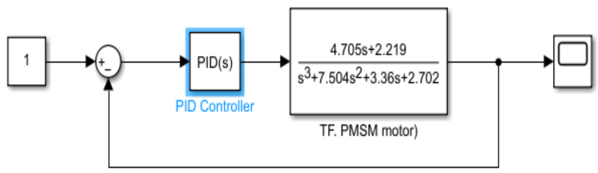

Figure 3. Diagrammatic representation of the PMSM motor block system, complete with PID controller

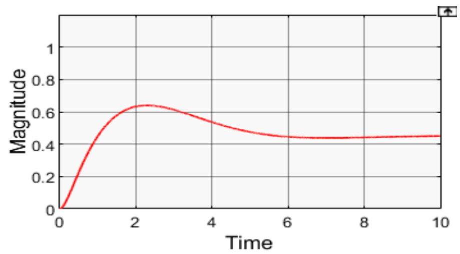

Figure 4. PMSM motor output motor without controller

The highest point grew significantly, the system's stability and elevation periods were prolonged, and the observed value was close to the expected value. The particle swarm parameters are established for the PSO technique, as stated in Table 2. Figure 5 displays the speed step response of a PMSM motor using a conventional PID - PSO.

Table 2. PSO values parameters

Titles of Parameters	Values
Inertia	0.7
c1	2
c2	2
NO. of particle	20
NO. of iteration	100

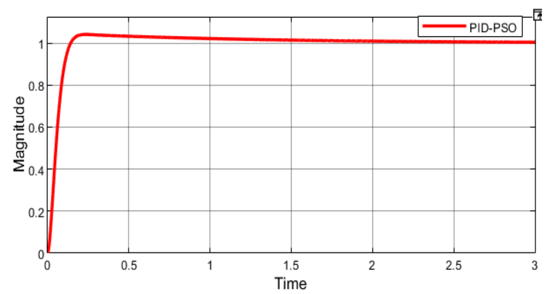

Figure 5. The step response of the PMSM motor when controlled by a PSO-PID controller

The ACO Algorithm's characteristics were set up to match the information in Table 3. Figure 6 demonstrates the step response of PMSM Motor velocity when controlled by a standard PID-ACO.

Table 3. ACO values parameters

Parameters' names	Values
Evaporate rate	0.5
Scaling rate	2
NO. of iteration (N)	100
Size of sample	40
Pop. size	20
size of Step	1

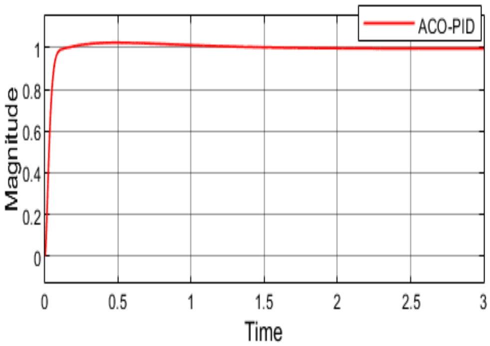

Figure 6. Speed response of the PMSM motor using the control unit ACO - PID

Table 4. GWO values parameters

Parameters' names	Values
Evaporate rate	0.5
Scaling rate	2
NO. of iteration (N)	100
Size of sample	40
size of Step	1
Pop. size	20

The response of the PMSM motor's speed stepping is shown when it is controlled by an ACO-PID controller in Figure 6.

The GWO Algorithm was configured using the parameters shown in the Table 4. Figure 7 illustrates the step response of PMSM Motor speed when controlled by a typical PID controller using GWO.

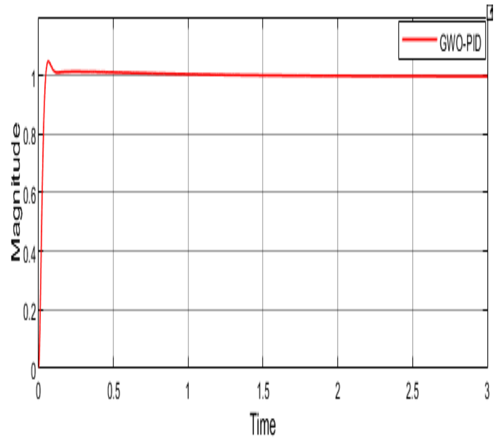

Figure 7. GWO-PID controller step response of PMSM motor speed

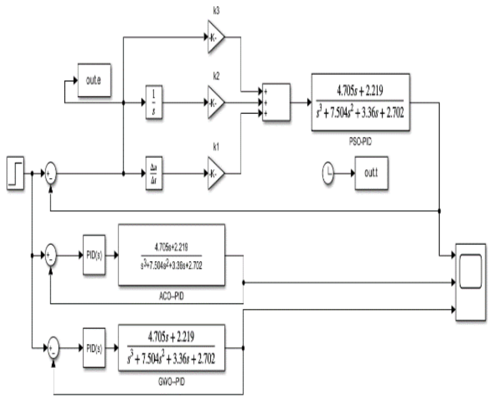

Figure 8. Block diagram for comparison between the PSO-PID, GWO-PID and ACO-PID Simulink in MATLAB

In Figure 8, block diagram shows comparison of PSO-PID, GWO-PID and ACO-PID Simulink in MATLAB.

After contrasting the PSO method with the ACO technique, we arrive at the results in Table 4.

Table 5. A comparison of the outcomes obtained by the various suggested algorithms (traditional PID controller)

Method	Input	M.P. (%)	Rise Time (sec)	Settling Time (sec)	Steady State Error
PSO-PID	K_p= 194.3689 K_i= 139.8394 K_d = 10.0119	3.646	0.08855	0.0132	0.000067
ACO-PID	K_p = 40.7362 K_i= 45.2896 K_d = 6.3493	1.484	0.0934	0.0521	0.000191
GWO-PID	K_p= 75.5372 K_i= 61.8052 K_d = 9.5482	4.747	0.0309	0.0455	0.000261

In terms of performance, Table 5 shows that the PSO and GWO algorithms are more valuable than the ACO algorithm, especially with regard to maximum overrun. A simulation of a PID controller with the molecular organization of the proposed system is shown in Figure 9 below. This Figure also compares each PSO algorithm's maximum bypass, fixation time and rise time. GWO and ACO.

Figure 10 shows PMSM motor system diagram with FOPID Control Simulink in MATLAB. Figure 11 demonstrates that the PSO- FOPID controller provides improved dynamic properties (overshoot and settling time) of a system response. Figure 12. demonstrates that the GWO- FOPID controller provides improved dynamic properties (overshoot and settling time) of a system response.

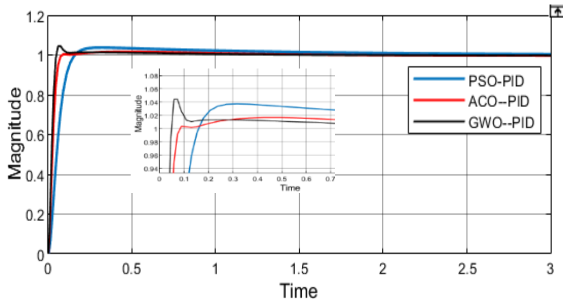

Figure 9. Compares the performance of the ant colony optimization method, the grey wolf optimization system, and the particle swarm optimization technique.

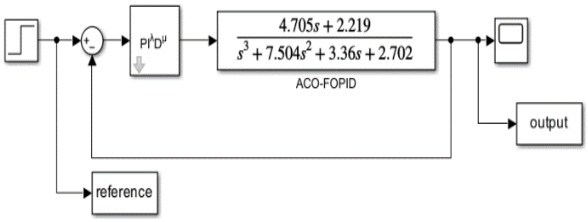

Figure 10. The block diagram of the PMSM system incorporates the FOPID controller Simulink in MATLAB.

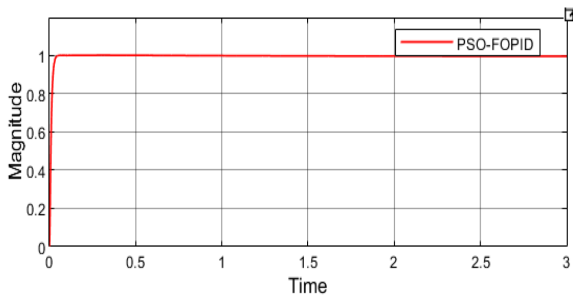

Figure 11. The step response of the PMSM motor speed when a PSO- FOPID controller controls it.

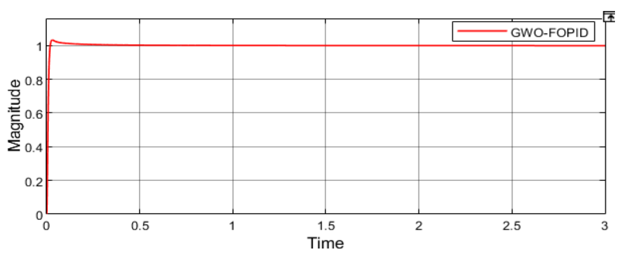

Figure 12. The step response of the PMSM motor speed when controlled by a GWO-FOPID controller

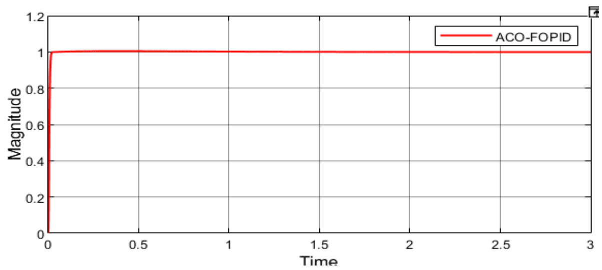

Figure 13. The step response of the PMSM motor speed when controlled by an ACO-FOPID controller.

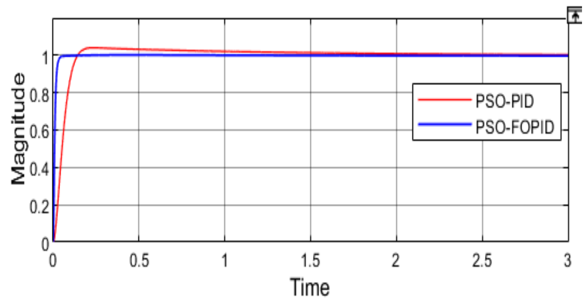

Figure 14. PSO-PID and PSO-FOPID step response of permanent magnet synchronous motor (PMSM) motor speed

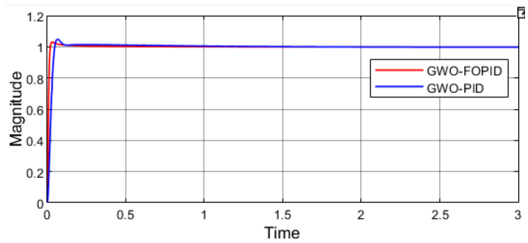

Figure 15. The step response of the PMSM motor speed when controlled by a GWO-PID or GWO-FOPID controller.

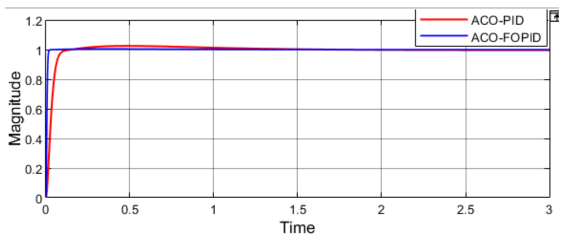

Figure 16. Demonstrates the step response of the PMSM motor speed when controlled by an ACO-FOPID controller.

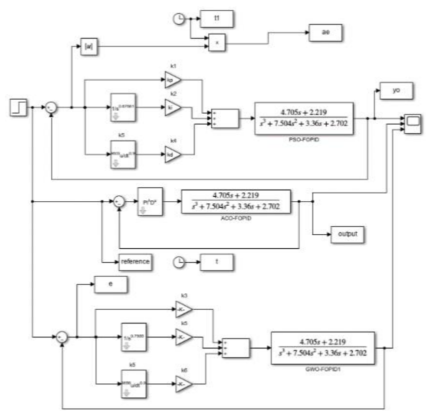

Figure 17. Block diagram for comparison between the FOPSO, FOGWO and FOACO Simulink in MATLAB

Figure 13 demonstrates that the ACO- FOPID controller provides improved dynamic properties (overshoot and settling time) of a system response compared to those produced by the PID controller.

Figure 14 demonstrates how the PSO-FOPID controller outperforms the PSO-PID controller regarding reliability and route tracking.

Figure 15 demonstrates how the GWO-FOPID controller outperforms the GWO-PID controller regarding reliability and route tracking.

Figure 16 demonstrates that the ACO- FOPID controller provides improved dynamic properties (overshoot and settling time) of a system response compared to those produced by the PID controller.

Figure 17 shows block diagram showing comparison of FOPSO, FOGWO, and FOACO Simulink in MATLAB.

The comparison of the PSO-FOPID, ACO-FOPID, and GWO-FOPID is depicted in Figure 18. demonstrates that the ACO-FOPID.

Figure 18 Block diagram for compares the FOPSO, FOACO, and FOGWO. The controller produces better dynamic properties (settling time and excess) of the system response than the PSO and GWO controllers.

The ACO-FOPID controller outperforms the PID controller. When all of the trials that resulted from the simulation are considered, it has been shown that the ACO-FOPID controller, which is superior to both the PSO-FOPID and the GWO-FOPID controllers, is the most effective way to attain performance as well as durability. This leads us to conclude that the stability and resilience of the system have grown, as illustrated in Figure 18 and Table 6.

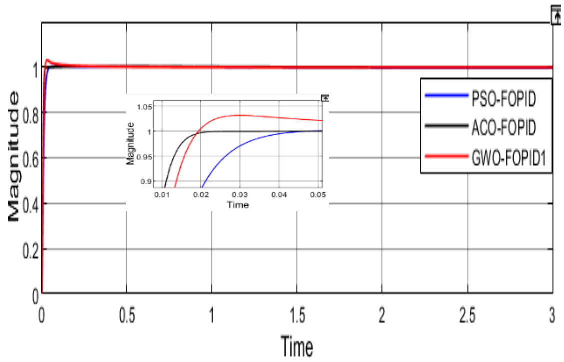

Figure 18. Compares the FOPSO, FOACO, and FOGWO

Table 6. A comparison of the outcomes obtained by the various suggested algorithms (traditional FOPID controller)

Method	Input	M.P. (%)	Settling Time (sec)	RiseTime (sec)	Steady State Error
PSO-FOPID	K_p= 135.7952 K_i= 48.8753 Lambda=0.798 K_d = 20.8952 Mu=0.985	0.452	0.0326	0.0182	0.00179
ACO-FOPID	K_p= 231.5537 K_i= 228.8589 Lambda=0.987 K_d = 34.5458 Mu=0.998	0.024	0.01	0.008978	0.000058
GWO-FOPID	K_p= 263.34 K_i= 174.73 Lambda=0.676 K_d = 32.205 Mu=0.96	0.505	0.0139	0.0085	0.0014

6. CONCLUSIONS

The search suggested adjusting the FOPID controller for managing the speed of the PMSM motor using the GWO, PSO, and ACO algorithms with the main goal of minimizing the ITAE objective function, which assesses the controller's capability to monitor the intended speed while reducing error over time. The study looked at a number of measures, including transient response, frequency response, settling time, rising time, and ITAE, to compare the efficiency of the proposed ACO-FOPID controller with that of other controllers, including ACO-PID, FOPID, GWO-PID, FOPID, PSO-PID, and PSO-FOPID. According to the comparison data, the ACO-FOPID controller outperformed the other controllers, indicating that it has the potential to regulate the speed of PMSM motors. The study hypothesizes that future investigation could enhance the performance and resilience of the controller.

REFERENCES

[1] Luo, Y., Chen, Y., Pi, Y. (2008). Authentic simulation studies of periodic adaptive learning compensation of cogging effect in PMSM position servo system. In 2008 Chinese Control and Decision Conference, pp. 4760-4765. https://doi.org/10.1109/CCDC.2008.4598233

[2] Nasri, M., Nezamabadi-Pour, H., Maghfoori, M. (2007). A PSO-based optimum design of PID controller for a linear brushless DC motor. World Academy of Science, Engineering and Technology, 26(40): 211-215.

[3] Shi, J.Z. (2020). A fractional order general type-2 fuzzy PID controller design algorithm. IEEE Access, 8: 52151-52172. https://doi.org/10.1109/ACCESS.2020.2980686

[4] Hamamci, S.E. (2007). An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers. IEEE Transactions on Automatic Control, 52(10): 1964-1969. https://doi.org/10.1109/TAC.2007.906243

[5] Belkhadir, A., Belkhayat, D., Zidani, Y., Pusca, R., Romary, R. (2022). Torque ripple minimization control of permanent magnet synchronous motor using adaptive ant colony optimization. In 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 629-635. https://doi.org/10.1109/CoDIT55151.2022.9804127

[6] Agarwal, S., Verma, A., Yadav, D. (2018). Performance analysis of PMSM drive using ant colony optimization. In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, pp. 5830-5836. https://doi.org/10.1109/IECON.2018.8592887

[7] Chiranjeevi, T., Babu, N.R., Yadav, A., Das, V.K., Prasad, S.C., Sonkar, A., Verma, S.K. (2021). Control of electric machines using flower pollination algorithm based fractional order PID controller. Global Transitions

Proceedings, 2(2): 227-232. https://doi.org/10.1016/j.gltp.2021.08.057
[8] Yadav, D., Verma, A. (2016). Performance analysis of permanent magnet synchronous motor drive using particle swarm optimization technique. In 2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), pp. 280-285. https://doi.org/10.22496/jetr.v1i2.90
[9] El-Saadawi, M.M., Gouda, E.A., Elhosseini, M.A., Essa, M.S. (2020). Identification and speed control of dc motor using fractional order pid: Microcontroller. European Journal of Electrical Engineering and Computer Science, 4(1): 170. http://dx.doi.org/10.24018/ejece.2020.4.1.170
[10] Szczepanski, R., Tarczewski, T., Grzesiak, L.M. (2019). Adaptive state feedback speed controller for PMSM based on Artificial Bee Colony algorithm. Applied Soft Computing, 83: 105644. https://doi.org/10.1016/j.asoc.2019.105644
[11] Ibrahim, E.K., Issa, A.H., Gitaffa, S.A. (2022). Optimization and performance analysis of fractional order PID controller for DC motor speed control. Journal Européen des Systémes Automatisés, 55(6): 741-748. https://doi.org/10.18280/jesa.550605
[12] Sebastian, A., Karbasizadeh, N., Saikumar, N., HosseinNia, S.H. (2021). Augmented fractional-order reset control: Application in precision mechatronics. In 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 231-238. https://doi.org/10.1109/AIM46487.2021.9517368
[13] Ibraheem, I.K., Ibraheem, G.A. (2016). Motion control of an autonomous mobile robot using modified particle swarm optimization based fractional order PID controller. Eng. Technol. J, 34(13): 2406-2419. https://doi.org/10.30684/etj.34.13A.4
[14] Liu, T.T., Tan, Y., Wu, G., Wang, S.M. (2009). Simulation of PMSM vector control system based on Matlab/Simulink. In 2009 International Conference on Measuring Technology and Mechatronics Automation, pp. 343-346. https://doi.org/10.1109/ICMTMA.2009.117
[15] Kulkarni, S.S., Thosar, A.G. (2013). Mathematical modeling and simulation of permanent magnet synchronous machine. International Journal of Electronics and Electrical Engineering, 1(2): 66-71. https://doi.org/10.12720/ijeee.1.2.66-71
[16] Karteek, Y.V.P., Kumar, N.P. (2016). Transfer function model based analysis of permanent magnet synchronous motor with controllers. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 4(11): 8-14. https://doi.org/10.17148/IJIREEICE.2016.41102
[17] Hinchey, M.G., Sterritt, R., Rouff, C. (2007). Swarms and swarm intelligence. Computer, 40(4): 111-113. https://doi.org/10.1109/MC.2007.144
[18] Anantathanavit, M., Munlin, M.A. (2013). Radius particle swarm optimization. In 2013 International Computer Science and Engineering Conference (ICSEC), pp. 126-130. https://doi.org/10.1109/ICSEC.2013.6694765
[19] Mohamed, M.J., Hamza, M.K. (2019). Design PID neural network controller for trajectory tracking of differential drive mobile robot based on PSO. Engineering and Technology Journal, 37(12A): 574-583. https://doi.org/10.30684/etj.37.12A.12
[20] Chaharsooghi, S.K., Kermani, A.H.M. (2008). An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP). Applied Mathematics and Computation, 200(1): 167-177. https://doi.org/10.1016/j.amc.2007.09.070
[21] Li, Y., Lin, X., Liu, J. (2021). An improved gray wolf optimization algorithm to solve engineering problems. Sustainability, 13(6): 3208. https://doi.org/10.3390/su1306320
[22] Alothman, Y.N.I., Abdul-Lateef, W.E., Gitaffa, S.A.H. (2023). Using sensorless direct torque with fuzzy proportional-integral controller to control three phase induction motor. Bulletin of Electrical Engineering and Informatics, 12(2): 738-748. https://doi.org/10.11591/eei.v12i2.3991
[23] Al-Khazraji, H. (2022). Optimal design of a proportional-derivative state feedback controller based on meta-heuristic optimization for a quarter car suspension system. Mathematical Modelling of Engineering Problems, 9(2): 437-442.
[24] Bououden, S., Chadli, M., Karimi, H.R. (2015). An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. Information Sciences, 299: 143-158. https://doi.org/10.1016/j.ins.2014.11.050
[25] Al-Khazraji, H., Khlilb, S., Alabacy, Z. (2022). Solving Mixed-Model Assembly Lines Using a Hybrid of Ant Colony Optimization and Greedy Algorithm. Engineering and Technology Journal, 40(01): 172-180. http://doi.org/10.30684/etj.v40i1.2153
[26] Abdulakareem, M.I., Raheem, F.. (2020). Development of path planning algorithm using probabilistic roadmap based on ant colony optimization. Engineering and Technology Journal, 38(3): 343-351. https://doi.org/10.30684/etj.v38i3A.389

NOMENCLATURE

Rs	stator resistance [Ω]
∅	rotor magnetic flux [Weber]
Lq, Ld	quadrature and direct axis inductance [H]
P	number of pole pairs
Iq, Id	quadrature and direct axis currents [A]
p	derivative concerning time
Eb	back emf [V]
Te	electromagnetic torque [Nm]
T	load torque [Nm]
B	friction coefficient
J	moment of inertia [$\mathrm{Kg-m^2}$]
Kt	torque constant
We	angular rotation[rad/sec]
λd, λq	flux linkages [weber]
λaf	mutual flux between magnet and stator.
Xt	Position of particle
Vt	Velocity of particle
pbest	Best position of a single particle
gbest	Best position of all particles
w	Inertial Weight factor
C2, C1	Acceleration Coefficients