

Optimization and Performance Analysis of Fractional Order PID Controller for DC Motor Speed Control



Entidhar K. Ibrahim^{1*}, Abbas H. Issa², Sabah A. Gitaffa²

¹ Ministry of Labor, the National Centre for Occupational Health and Safety, Baghdad 10066, Iraq

² Department of Electrical Engineering, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: eee.20.48@grad.uotechnology.edu.iq

<https://doi.org/10.18280/jesa.550605>

ABSTRACT

Received: 2 November 2022

Accepted: 22 November 2022

Keywords:

DC Motor, PID, FOPID, PSO, ACO

Direct Current (DC) motors need control and reliability to enhance their performance. The most used controller is PID. However, due to the empirically chosen parameters K_p , K_i and K_d beside the limitation of traditional PID controllers in achieving perfect control response for systems with higher order, a Fractional Order Proportional-Integral-Derivative (FOPID) controller is based on efficient optimization methods. The aim of this paper is using a metaheuristic algorithms for designing a speed controller for a DC motor based on FOPID parameter. System parameters are studied and compared. The simulations of meta-heuristic algorithms are carried out in MATLAB version R2020b. The results showed that the Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) algorithms reduced the ITAE objective function in speed control system using a unity feedback FOPID controller. Performance comparisons of the traditional PSO tuned PID and FOPID controllers, beside newer methods, were undertaken to illustrate the efficacy of the proposed PSO-FOPID methodology. ACO-FOPID and ACO-PID were chosen as recent techniques since they utilized the same DC motor type. According to the findings, the suggested PSO-FOPID controller had the superior transient and frequency response in terms of rising time which equals 0.01537 sec, settling time equals 0.01556 sec and zero ITAE; however, it had the poorest maximum overshoot with (2.557%) compared with ACO-FOPID controller.

1. INTRODUCTION

Because of their ease of control, reliability and great performance, direct current DC motors are frequently employed in applications of industry. Robustness is the key criteria of the design in DC motor speed control, followed by system performance [1, 2]. As a result, several controllers are built to provide disturbance rejection and resilience. Many approaches, including traditional controllers like Proportional-Integral-Derivative (PID), meta-heuristic approaches and intelligent swarm techniques have been utilized to modify the parameters of these controllers [3]. The most often utilized controllers to control a DC motor's speed were PI and PID. However, changing PID controller parameters when distributing is the main problem [4] and due to the empirically chosen parameters K_p , K_i and K_d as well as the limitation of traditional PID controllers in achieving perfect control response for higher-order systems, where K_p has the ability to amplify the error signal but it affects the system stability, K_i has the unique ability to return the controlled variable back to the exact set point, however it responds relatively slower to error signals and it can initially allow a large deviation at the instant of the error which can lead to system instability and cyclic operation. K_d reduces the initial overshoot of the measured variable and therefor stabilizing the process earlier, an FOPID controller based on efficient techniques of optimization like PSO and ACO can be proposed. FOPID controller settings are investigated for reducing the Integral Time Absolute Error (ITAE) between

the output of the reference model and the output of the plant [5]. To keep the required performance, the controller must be fine-tuned. To achieve an effective gain values for the controller, numerous tuning procedures were used. Many of the tuning tasks are done by hand, but tuning by hand procedure is time consuming and also challenging. Computational intelligence approaches are now widely utilized for tuning Controller parameters, as it extracts the most accurate results in record time, while the manual tuning method requires many attempts, though, the results may be inaccurate. Many strategies for designing FOPID controllers have been reported as the following literature [6]. Idir et al. proposed Differential Evolution (DE) and PSO to design FOPID controller to control the DC motor's speed. The results showed that the DE-FOPID controller exceeds the PSO-FOPID controller in terms of performance, noise rejection, flexibility, and robustness [5]. Altinoz. and Yilmaz presented Gravitational Search Algorithm (GSA), PSO and DE for quarter-car suspension system, when using conventional PID, all optimization techniques got almost the same performance, however the greatest performance is achieved from the PSO algorithm with the minimum overshoot in FOPID. GSA had the worst performance. Furthermore, the DE's performance was almost the same for traditional and fractional order PID implementations [7]. Al-Dhaifallah et al. introduced a Fuzzy Fractional-Order PID (FFOPID) controller scheme for pneumatic pressure regulating systems and compared it to FOPID and

conventional PID. FFOPID achieves the best performance [8]. Deacha Puangdownreong presented an optimum FOPID controller for the system of DC motor speed control by utilizing the Flower Pollination Algorithm (FPA), as the simulation findings showed that the FOPID could respond significantly faster than the classical PID. The results showed 0.001 percent error, 0.1455 sec settling time, 0.91 percent for MP and 0.0889 sec rising time [9]. El-Saadawi et al. [10] proposed Gray Wolf Optimization (GWO/FOPID), tuning of Single-loop PID, Nelder-Mead (NM/FOPID) and Genetic Algorithm (GA) to regulate the speed of a DC motor. It is determined that the NM algorithm-based FOPID provides the optimum response. The best results are obtained when the associated settling time and rising time are 0.022 sec, 0.02 sec, with no overshoot. Tirumalasetty Chiranjeevi et al. proposed FPA based PID and FOPID controller for controlling the speed of electric machinery, a comparison of FPA with Fire Fly Algorithm (FA) and PSO approaches were performed. FOPIDC responses exceed PIDC in terms of setup time and oscillation amplitude for output and input responses. ISE-based responses improve dynamics over ITAE and IAE while FPA outperforms FA and PSO [11]. K. Vanchinathan and N. Selvaganesan proposed the Artificial Bee Colony (ABC) algorithm to improve the performance of DC motors (operate at a variety of speeds and loads with improved performance and robust speed control), the obtained results were compared to those of Conventional Genetic Algorithm (CGA) and Modified Genetic Algorithms (MGA). The results were that the ABC optimized FOPID controller can reduce the peak time, rise time and settling time compared to the MGA and CGA optimized FOPID controllers for all load cases (0%, 40%, 60%, 80% and 100% load) and the steady state error is less than the CGA and MGA optimized FOPID controllers [12]. This paper provides metaheuristic algorithms for designing a speed controller for a DC motor based on FOPID parameter selection. To demonstrate the efficacy of this combination, all system parameters are studied and compared.

The following is the structure of this paper: Section 2 presents the fractional order PID controller. Section 3 is concerned with the mathematical modelling of a direct current motor. Section 4 offers a summary of optimization techniques and some metaheuristic algorithms which are PSO and ACO algorithms. Results of the experiments applies on section 5 using a simulated proposed model to build up a fractional order PID controller; finally, section 6 highlights the proposed system's conclusions.

2. FRACTIONAL ORDER PID CONTROLLER

A PID controller is a type of feedback control loop mechanism that is a term often used in control systems due to its simplicity of design and superior performance [13]. As a result, it is worthwhile to increase their operational quality. One way used by engineers to increase the performance of the PID controller is to employ a FOPID controller. A variation on the standard PID controller which has fractional integral and derivative orders rather than integer. FOPID features two additional parameters, (integral order) and (derivative order) and they are fractions; this boosts the system's flexibility and enables a more effective implementation than a standard PID controller [14]. Variations in the monitored system's and controller's

parameters are less sensitive to fractional controllers [15]. Figure 1 shows the FOPID controller block diagram [16], [17]. The PID controller's Laplace transform maybe represented as in Eq. (1) [16]:

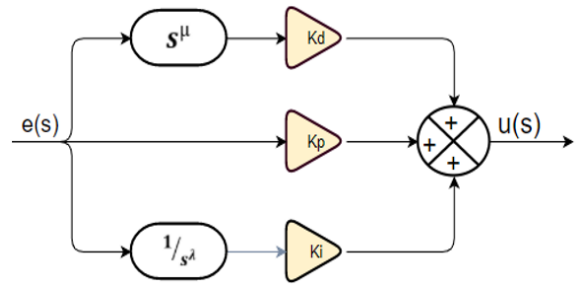


Figure 1. FOPID controller block diagram

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

where:

$G_c(s)$: transfer function of PID controller

K_p : Proportional gain

K_i : Integral gain

K_d : Derivative gain

The FOPID controller's generalized transfer function is described in Eq. (2) [18].

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

where:

$e(s)$: is error generated

$u(s)$: is the output of the controller

$K_p, K_i, \text{ and } K_d$: are the gains in terms of proportional, integral and derivative, respectively

λ : is the integral term's fractional part

μ : is the derivative term's fractional part

FOPID controller is a generalization of the classical PID controller which spans from point to point as shown in Figure 2 [19].

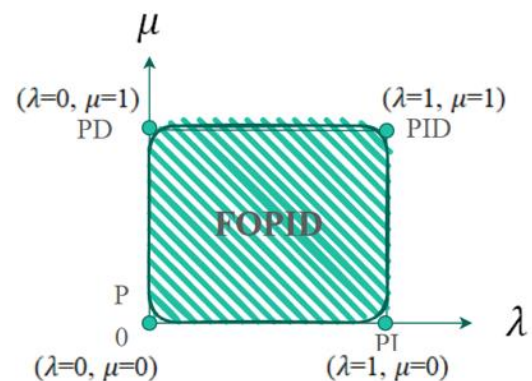


Figure 2. Active region of classical PID and FOPID controllers

3. MODELLING OF ARMATURE-CONTROLLED DC MOTOR

Figure 3 shows the schematic design of an armature-controlled DC motor equivalent circuit [20, 21].

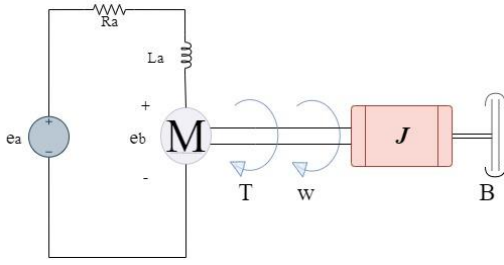


Figure 3. Simple DC motor structure

The differential equations with Laplace transform can be exploited to build a mathematical model of an armature-controlled DC motor in terms of the transfer function, as shown in Eq. (3) [11, 22]:

$$\frac{w(s)}{e_a(s)} = \frac{K_i}{La.Jm s^2 + (Ra.Jm + La.Bm)s + (Ra.Bm + Kb.Ki)} \quad (3)$$

where:

- e_a : is the armature input voltage (V)
- R_a : is the armature resistance (Ω)
- L_a : is the armature inductance (H)
- i_a : is the armature current (A)
- e_b : is back generated EMF (V)
- T_m : is motor generated torque (Nm)
- J_m : is moment of inertia of motor body (kg.m²)
- B_m : is damping coefficient (due friction) (Nm*sec/rad)
- K_i : is torque constant
- K_b : is back emf constant

For DC motor proposed, the physical parameters are in Table 1 [23]:

Table 1. DC motor parameters

Parameter	Values	Units
Friction B_m	0.00003	Nm. s
Inductance of Coil L_a	0.5	H
Motor inertia J_m	0.01	kg.m
Motor constants K_b, K_i	0.023	-
Coil resistance R_a	1	Ω

The DC motor plant in Eq. (3) may be written as in Eq. (4)

$$\frac{w(s)}{e_a(s)} = \frac{0.023}{0.005 s^2 + 0.01s + 0.000559} \quad (4)$$

4. OPTIMIZATION TECHNIQUES

Further study is necessary to regulate control software capable of overcoming the challenges of system nonlinearities in order to boost the dependability, controllability, and use of the better speed of response feasible from DC motor systems. PSO and ACO are used to optimize the FOPID parameters.

4.1 Particle Swarm Optimization (PSO)

Eberhart and Shi [24] devised the PSO in 1995. It is a method of optimization of evolutionary depends on the

mobility and the swarm intelligence which travel about in search spaces seeking the optimal result. A particle is represented by a point in N-dimensional space, and its movement is changed based on both its own and the other particles' flying experiences. Every particle attempts to update its position by utilizing its velocity, present position, the separation between the best position of it 'pbest' and the present position and the separation between the global best position 'gbest' and its present position. Eqns. (5) and (6) do this [25-27]:

$$V_{t+1} = w \times V_t + C1 \times rand(0,1) \times (Pbest - X_t) + C2 \times rand(0,1) \times (gbest - X_t) \quad (5)$$

$$X_{t+1} = X_t + V_{t+1} \quad (6)$$

where:

- V_t : Velocity of particle
- X_t : Position of particle
- $gbest$: Best position of all particles
- $pbest$: Best position of a single particle
- $C1$ and $C2$: Acceleration Coefficients
- w : Inertial Weight factor

After calculating the new X_t , the particle analyzes its new position. If fitness (X_t) is better than fitness ($pbest$), then $X_t = pbest$ and fitness (X_t) = fitness ($pbest$) and when iteration is completed, better fitness ($pbest$) = fitness ($gbest$) and $pbest = gbest$ [28].

The procedure of PSO method is shown in Figure 4.

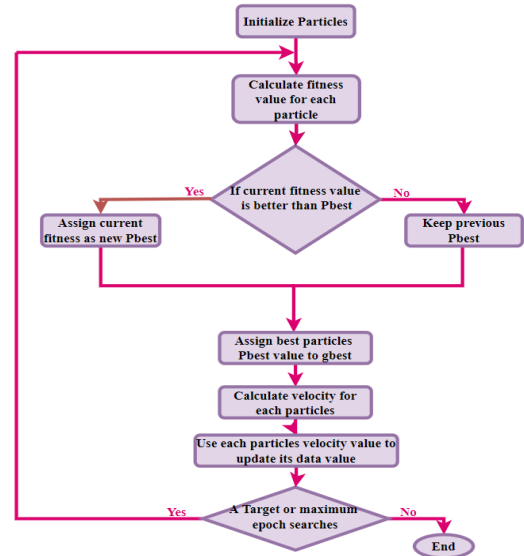


Figure 4. The flow chart of PSO method procedure

4.2 Ant Colony Optimization (ACO)

ACO is a metaheuristic that mimics the collective social insects' cognitive behavior, particularly ant foraging activity. When real ants discover food and other resources, they return back to their nest while pheromones are left on the ground. Pheromones induce other ants, and induced ants may reach a specific location. They will often return to their nest location while pheromones are being deposited on the ground. Because pheromones are volatile, the pheromone of the route

which is longer, will vanish faster than the pheromone of the shorter way. In this manner, the ants may select a quick route from their nest to the nourishment place and successfully carry supplies [29, 30].

The probability of an ant traveling from node i to node j is represented by Eq. (7) [31]:

$$p_{ij} = \frac{\tau_{ij}^\alpha d_{ij}^\beta}{\sum_{i,j=1}^n \tau_{ij}^\alpha d_{ij}^\beta} \quad (7)$$

where, α and $\beta > 0$ are the pheromone level's weight and distance, respectively. τ_{ij} refers to levels of pheromone, d_{ij} is the spacing between nodes i and j . The pheromone evaporates with time as seen in Eq. (8):

$$\tau_{ij} = (1 - \gamma)\tau_{ij} \quad (8)$$

where, γ represents the evaporation coefficient of the pheromone.

When all ants have finished their solution, the pheromone level then updated by entirely new level deposited by ants after the evaporation process has been completed, using the following formula in Eq. (9):

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (9)$$

where, k is the ant number, m is the population and $\Delta\tau_{ij}^k$ can be determined using the Eq. (10):

$$\Delta\tau_{ij}^k = \frac{1}{c^k} \quad (10)$$

C^k is the ant's reward for taking the right path. The procedure of ACO method is shown in Figure 5.

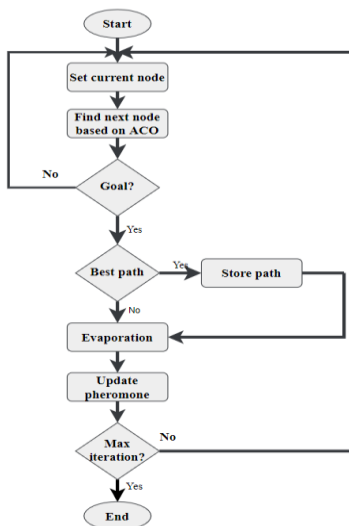


Figure 5. The flow chart of ACO method procedure

5. EXPERIMENTAL RESULTS

The simulations of meta-heuristic algorithm transient response analysis are carried out in MATLAB version

R2020b (9.9.0.1467703) 64-bit (win64). The results for each technique are achieved using the toolbox of FOMCON to implement the FOPID. The entire transfer function of DC motor is given in Eq. (3). The final DC motor transfer function may be produced by employing the indicated parameter values showed in Table (1) to the proposed DC motor system as in Eq. (4). Figure 6 depicts a simulation of the suggested system. Figure 7 shows the output of simulated DC motor with standard PID controller.

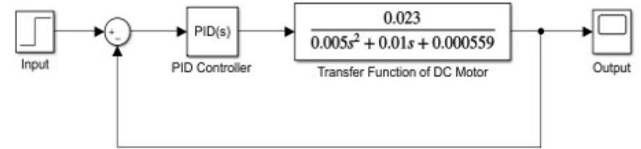


Figure 6. The schematic of the block system of DC motor

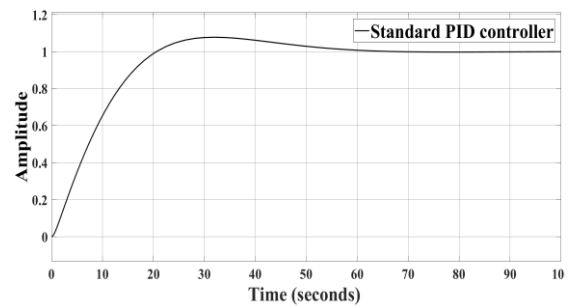


Figure 7. Output of simulated DC Motor with standard PID controller

Despite the system's long settling and rising times, the highest peak reduced dramatically and the measured value approximated the anticipated value.

For the PSO method, the particle swarm parameters are set as shown in the Table 2. Figure 8 shows the DC motor's speed Step response with standard PID controller via PSO.

Table 2. The PSO parameters values

Parameters' names	Values
Inertia	0.4
Number of iterations	100
Number of Particles	20
Cognitive Component c1	2
Social Component c2	2

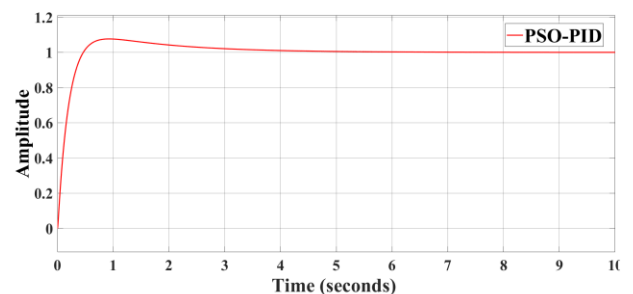


Figure 8. Step response of DC motor speed with PSO-PID controller

The settings of the ACO Algorithm were as given in Table (3). Figure 9 shows the DC motor speed step response with standard PID controller via ACO.

Table 3. The ACO parameters values

Parameters	Values
No. of iteration (N)	100
Pop. Size	10
Step size	1
Sample size	40
Evaporate rate	0.6
Scaling rate	2

The chosen parameters of PSO and ACO algorithms were set manually and they were modified till obtaining the best response for the system (Minimum overshoot, minimum settling and rising time, and minimum steady-state error). The large number of iterations was used to extend the research scope.

Table 4. A comparison between proposed algorithms results (traditional PID controller)

Method	Input parameters	MP (%)	Settling time (sec)	Rise time (sec)	Steady State Error
PID	$K_p = 0.0346$	7.73	53.2	14.5	0.0008
	$K_i = 0.0037$				
	$K_d = -0.0048$				
PSO-PID	$K_p = 217.3917$	6.989	3.07	0.3128	0.00016
	$K_i = 132.2257$				
	$K_d = 39.2870$				
ACO-PID	$K_p = 10$	1.531	0.1056	0.0339	0.0004
	$K_i = 3$				
	$K_d = 10$				

Table (4) shows that the ACO algorithm substantially exceed the PSO algorithm in terms of time domain performance, particularly for the maximum overshoot, rising time and settling time. Figures (10 and 11) depict a simulation of fractional order PID controller for the suggested system and the DC motor speed step response with FOPID controller using PSO algorithm respectively.

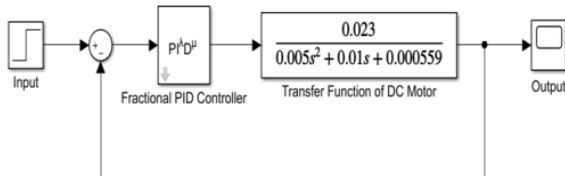


Figure 10. The block diagram system of DC motor with FOPID controller

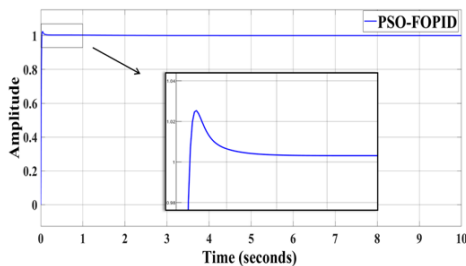


Figure 11. Step response of DC motor speed with PSO-FOPID controller

The dynamic features (settling time and overshoot) of the system response that were generated from the PSO-FOPID controller are much superior to features produced from the PID controller, as shown in Figure 12.

By comparing the Particle Swarm Optimization algorithm and Ant Colony Optimization algorithm, we get Table (4).

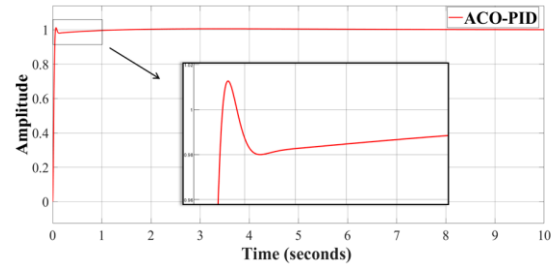


Figure 9. Step response of DC motor speed with ACO-PID controller

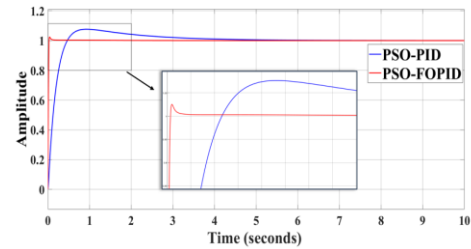


Figure 12. Step response of DC motor speed with PSO-PID and PSO-FOPID controllers

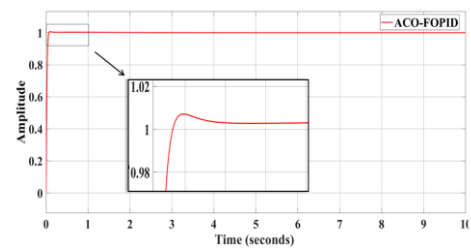


Figure 13. Step response of DC motor speed with ACO-FOPID controller

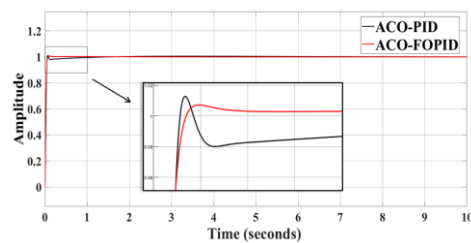


Figure 14. Step responses of DC motor with ACO-PID and ACO-FOPID controllers

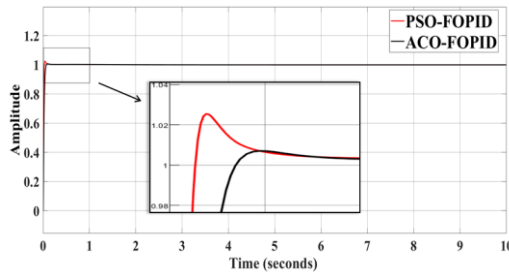


Figure 15. Step responses of DC motor with PSO-FOPID and ACO-FOPID controllers

Table 5. Comparison between proposed algorithms results (fractional PID controller)

Method	K_p	K_i	K_d	λ	μ	MP (%)	Settling time (sec)	Rise time (sec)	Steady State Error
PSO-FOPID	36.227	42.5402	29.7227	0.582781	0.925838	2.557	0.01556	0.01537	0
ACO-FOPID	11.2112	18.5922	13.9333	0.2919	0.9505	0505	0.0764	0.0337	0.0002

Table 6. The maximum overshoot values for systems responses with different noise signal values

Method	MP (%)	Noise power Value					
		0.2	0.8	1	2	3	4
PSO-FOPID	MP (%)	2.577	2.577	2.577	3.035	3.491	4.052
ACO-FOPID	MP (%)	0.505	0.505	0.505	0.505	0.505	0.505

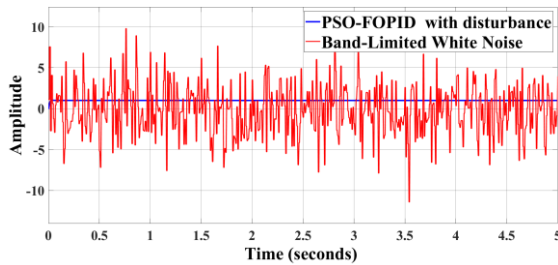


Figure 16. System response of DC motor with PSO-FOPID controllers with disturbance signal

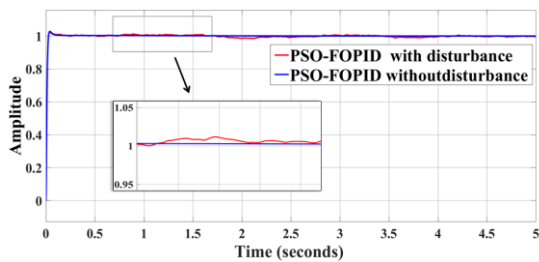


Figure 17. System response of a DC motor controlled by PSO-FOPID controllers with and without a disturbance signal

For checking the robustness of the system, a disturbance signal with (0.1) noise power was used, it was noted that the maximum overshoot of the disturbance signal was much higher than the maximum overshoot of the same disturbance signal after adding it to the system, which means that the controller has damped the noise signal by a percentage equals to (30%). From this, we conclude that the system has become more powerful and stable. Figure 16 illustrates this. Figure 17 depicts the effect of a disturbance supplied to the system using PSO-FOPID controller. In addition, when a disturbance signal was added to both controllers (ACO-FOPID and PSO-

The DC Motor speed step response using a FOPID controller and the ACO algorithm is shown in Figure 13.

As shown in Figure 14, the ACO-FOPID controller has higher reliability and offers superior tracking of a path than ACO-PID. Figure 15 shows a comparison of step response of DC motor speed with PSO-FOPID and ACO-FOPID controllers.

Taking into account all of the experiments results from the simulation, noise rejection, good performance and robustness can be achieved with the PSO-FOPID controller and it outperforms the ACO-FOPID controller in terms of performance as shown in Figure 15 and Table 5.

FOPID) they showed the same overshoot value in the two cases (with and without disturbance signal). Also, when the noise power value increased for PSO-FOPID system response, the maximum overshoot increased relatively. But for ACO algorithm, if noise power increased, the system will maintain the same overshoot value. Table (6) shows different noise signal values chosen to test the systems responses along with the obtained results.

According to the obtained results, FOPID controller enhances the system in terms of time and transient response, the required specifications (minimum overshoot, minimum settling time, minimum rise time, and minimum steady-state error) which have been also achieved, giving better and more accurate results. In addition, the PID contains the order 1 only for both the integral term and derivative term, but in FOPID, the controller may contain a range of orders between 0 and 1 which gives more flexibility in determining the parameters of the designed controller, that gives the system more expansion and flexibility.

6. CONCLUSION

Tuning the settings of the FOPID controller is a difficult task. If the success of tuning is inadequate, not only the performances of control deteriorate, but the control system will also become ineffective. In this research, it has proposed for controlling DC motor speed using FOPID controller based on PSO and ACO algorithms. The PSO and ACO algorithms were used to reduce the ITAE objective function using a unity feedback FOPID controller. Comparisons of performance with the typical PSO adjusted PID and FOPID (PSO-PID and PSO-FOPID) controllers, as well as newer methods, were undertaken to illustrate the efficacy of the proposed PSO-FOPID methodology. ACO-FOPID and ACO-PID were chosen as recent techniques in the literature since they utilised the same DC motor type. According to the

findings of the comparison study, the suggested PSO-FOPID controller had the superior transient response and the frequency response is perfect in terms of settling time, rising time and ITAE, in addition to that the disturbance signal was damped when it was added, indicating the controller's strength and robustness. In future work, the non-linear FOPID controller can be used to further improve the system's response to external disturbances. In addition, the fuzzy logic can be used along with the FOPID controller for the purpose of increasing the robustness of the system.

REFERENCES

- [1] Hasan, F.A., Rashad, L.J. (2019). Fractional-order PID controller for permanent magnet DC motor based on PSO algorithm. *International Journal of Power Electronics and Drive Systems*, 10(4): 1724. <https://doi.org/10.11591/ijpeds.v10.i4.1724-1733>
- [2] Chaiya, U., Kaitwanidvilai, S. (2009). Fixed structure robust DC motor speed control. *Proceeding of the International Multiconference of Engineering and Computer Scientists*, 2: 18-20.
- [3] 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.
- [4] Abedulabbas, G.W., Yaseen, F.R. (2022). Design a PI controller based on PSO and GWO for a Brushless DC Motor. *Journal Européen des Systèmes Automatisés*, 55(3): 331-333. <https://doi.org/10.18280/jesa.550305>
- [5] Idir, A., Kidouche, M., Bensafia, Y., Khettab, K., Tadjer, S.A. (2018). Speed control of DC motor using PID and FOPID controllers based on differential evolution and PSO. *Evolutionary computation*, 11(4): 241-249. <https://doi.org/10.22266/ijies2018.0831.24>
- [6] Ameen, N.M., Humod, A.T. (2020). Robust nonlinear PD controller for ship steering autopilot system based on particle swarm optimization technique. *IAES International Journal of Artificial Intelligence*, 9(4): 662. <https://doi.org/10.11591/ijai.v9.i4.pp662-669>
- [7] Altinoz, O.T., Yilmaz, A.E. (2018). Optimal PID design for control of active car suspension system. *International Journal of Information Technology and Computer Science (IJITCS)*, 10(1): 16-23. <https://doi.org/10.5815/ijitcs.2018.01.02>
- [8] Al-Dhaifallah, M., Kanagaraj, N., Nisar, K.S. (2018). Fuzzy fractional-order PID controller for fractional model of pneumatic pressure system. *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2018/5478781>
- [9] Puangdownreong, D. (2019). Fractional order PID controller design for DC motor speed control system via flower pollination algorithm. *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, 17(1): 14-23. <https://doi.org/10.37936/ecti-ec.2019171.215368>
- [10] 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): 1-8. <https://doi.org/10.24018/ejece.2020.4.1.170>
- [11] 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>
- [12] Vanchinathan, K., Selvaganesan, N. (2021). Adaptive fractional order PID controller tuning for brushless DC motor using artificial bee colony algorithm. *Results in Control and Optimization*, 4: 100032. <https://doi.org/10.1016/j.rico.2021.100032>
- [13] Xue, D.Y., Zhao, C.N., Chen, Y.Q. (2006). Fractional order PID control of a DC motor with elastic shaft: A case study. In *Proc. 2006 American control conference*, pp. 3182-3187. <https://doi.org/10.1109/ACC.2006.1657207>
- [14] Ibraheem, I.K., Ibraheem, G.A. (2016). Motion control of an autonomous mobile robot using modified particle swarm optimization based fractional order PID controller. *Engineering and Technology Journal*, 34(13): 2406-2419
- [15] Shah, P., Agashe, S. (2016). Review of fractional PID controller. *Mechatronics*, 38: 29-41. <https://doi.org/10.1016/j.mechatronics.2016.06.005>
- [16] Rastogi, A., Tiwari, P. (2013). Optimal tuning of fractional order PID controller for DC motor speed control using particle swarm optimization. *International journal of soft computing and engineering (IJSCE)*, 3(2): 150-157.
- [17] Abood, L.H., Oleiwi, B.K. (2021). Design of fractional order PID controller for AVR system using whale optimization algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(3): 1410-1418. <https://doi.org/10.11591/ijeecs.v23.i3.pp1410-1418>
- [18] Al-Araji, A.S., Rasheed, L.T. (2017). A cognitive nonlinear fractional order PID neural controller design for wheeled mobile robot based on bacterial foraging optimization algorithm. *Engineering and Technology Journal*, 35(3): 289-300. <https://doi.org/10.30684/etj.35.3A.15>
- [19] Bouderes, N., Kerdoun, D., Djellad, A., Chiheb, S., Dekhane, A. (2022). Optimization of fractional order PI controller by PSO algorithm applied to a grid-connected photovoltaic system. *Journal Européen des Systèmes Automatisés*, 55(4): 427-438. <https://doi.org/10.18280/jesa.550401>
- [20] Hekimoğlu, B. (2019). Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access*, 7: 38100-38114. <https://doi.org/10.1109/ACCESS.2019.2905961>
- [21] Wheed, A., Issa, A.H. (2021). FPGA implementation of modified reconfigurable adaptive gain scheduling controller. *Engineering and Technology Journal*, 39(7): 1105-1116, <https://doi.org/10.30684/etj.v39i7.1772>
- [22] Mohammed, R.I., Abbas, S.M., Issa, A.H. (2018). FPGA cyclone II based the mobile robot control system. In *2018 Third Scientific Conference of Electrical Engineering (SCEE)*, pp. 186-192. <https://doi.org/10.1109/SCEE.2018.8684214>
- [23] Sultan, G.A., Sheet, A.F., Ibrahim, S.M., Farej, Z.K. (2021). Speed control of DC motor using fractional order PID controller based on particle swarm optimization. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(3): 1345-1353.

- <https://doi.org/10.11591/ijeecs.v22.i3.pp1345-1353>
- [24] Eberhart, R.C., Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation, CEC, 1: 81-86. <https://doi.org/10.1109/CEC.2001.934374>
- [25] 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(12): 574-583. <https://doi.org/10.30684/etj.37.12A.12>
- [26] Abood, L.H., Karam, E.H., Issa, A.H. (2018). FPGA implementation of single neuron PID controller for depth of anesthesia based on PSO. In 2018 Third Scientific Conference of Electrical Engineering (SCEE), pp. 247-252. <https://doi.org/10.1109/SCEE.2018.8684186>
- [27] Dawood, Z.H., AL-Rubayi, R. (2021). Analysis of distribution system reconfiguration under different load demand in AL-KUT City by using PSO Algorithm. *Engineering and Technology Journal*, 39(5): 738-753. <http://doi:10.30684/etj.v39i5A.1842>
- [28] Veysel G., Kevin M.P. (2011). *Swarm Stability and Optimization*. Springer Science +Business Media B.V.: 299. <https://doi.org/10.1007/978-3-642-18041-5>
- [29] Prakash, A., Tiwari, M.K., Shankar, R. (2008). Optimal job sequence determination and operation machine allocation in flexible manufacturing systems: An approach using adaptive hierarchical ant colony algorithm. *Journal of Intelligent Manufacturing*, 19(2): 161-173. <https://doi.org/10.1007/s10845-008-0071-y>
- [30] Arnaout, J.P. (2013). Ant colony optimization algorithm for the euclidean location-allocation problem with unknown number of facilities. *Journal of Intelligent Manufacturing*, 24(1): 45-54. <https://doi.org/10.1007/s10845-011-0536-2>
- [31] Abdulakareem, M.I., Raheem, F.A. (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

PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
FOPID	Proportional-Integral-Derivative