

Comparison between butterfly optimization algorithm and particle swarm optimization for tuning cascade PID control system of PMDC motor

Kareem G. Abdulhussein, Naseer M. Yasin, Ihsan J. Hasan

Department of Electrical Power Engineering Techniques, Middle Technical University, Iraq

Article Info

Article history:

Received Jan 21, 2021

Revised Apr 3, 2021

Accepted Apr 11, 2021

Keywords:

Butterfly optimization algorithm

Cascade PID controller

Particle swarm optimization

PMDC

ABSTRACT

In this paper, two optimization methods are used to adjust the gain values for the cascade PID controller. These algorithms are the butterfly optimization algorithm (BOA), which is a modern method based on tracking the movement of butterflies to the scent of a fragrance to reach the best position and the second method is particle swarm optimization (PSO). The PID controllers in this system are used to control the position, velocity, and current of a permanent magnet DC motor (PMDC) with an accurate tracking trajectory to reach the desired position. The simulation results using the MATLAB environment showed that the butterfly optimization algorithm is better than the particle swarming optimization (PSO) in terms of performance and overshoot or any deviation in tracking the path to reach the desired position. While an overshoot of 2.557% was observed when using the PSO algorithm, and a position deviation of 7.82 degrees was observed from the reference position.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Kareem Ghazi Abdulhussein

Department of Electrical Power Engineering Techniques

Middle Technical University

Babylon, Iraq

Email: engkarim1984@gmail.com

1. INTRODUCTION

A permanent magnet DC motor (PMDC) is a simple structure type of DC motor in which the field windings have been replaced by permanent magnets and have many applications so It operates as the same basic principle as a shunt connected motor but the difference is that the permanent magnet generates the required flux instead of field windings [1], [2].

Permanent magnet DC motors provide less output power than DC shunt motor because the flux generated from permanent magnets is less than what can be achieved with field windings so most types of small DC motors are permanent magnet DC motors and these motors usually operate at high velocity and little torque [1], [3]. This motor has many applications such as its use in car front wipers and moving windows for cars and household appliances such as food mixers and others [4], [5]. In addition to the importance of using it in CNC machines, robotics, and electric vehicles [6], [7].

The main problem statement in this paper is how to get the best parameter values for cascade PID controller which gives the best accurate results for tracking trajectory the reference position and reaching the desired position at a regular velocity. To address this problem statement, a comparison was made between the BOA and PSO methods for extracting the parameters of the cascade PID controller, in addition to using one

of the performance criteria which is integral time absolute error (ITAE) to reduce the error between input and output for this system.

The cascade PID controller which is used in this paper consists of three controllers, the current controller is designed as an inner loop, and both position and velocity are outer controllers [8]. P, PI, and PI controllers are used instead of a PID controller for position, velocity, and current respectively [9].

The purpose of using the cascade control unit is due to several reasons, the most important of which is to reduce or reject disturbance and return to the steady-state [10]-[12]. There are several methods to tuning the parameters of PID and use it in PMDC motor such as Ziegler-Nichols Method (Z-N), Cohen-Coon method, artificial neural, network fuzzy logic [13], [14], particle swarm optimization (PSO) [15], [16], genetic algorithm [17], and butterfly optimization algorithm (BOA) [18].

To compare this work with similar works, a literature review was made. In 2014 Md Mustafa et al introduced a BLDC speed control system using GA [19]. In 2015, Taha *et. al.* used three methods to control the cascade control system [9]. In 2018, Wisam *et. al.* introduced a system for controlling PMDC velocity using the GA and DS methods [20]. in 2019, Fadhil *et. al.* used a fractional PID controller to control PMDC speed based on PSO [21]. In 2021, Ahmed et al introduced a system to control the speed and position of the servo motor [22].

This paper is organized as follows: In the second section, the mathematical model of the PMDC motor is represented and explain the general structure of the system, in the third section, the two tuning methods are explained which are PSO and BOA in addition to the use of ITAE. The fourth section includes the results and comparison, while the final section includes a conclusion.

2. MATHEMATICAL MODEL OF PMDC MOTOR AND GENERAL STRUCTURE

2.1. Mathematical Model of PMDC Motor

The equivalent circuit for a PMDC motor shown in Figure 1 consists of an armature circuit consisting of inductance (L_a) and resistance (R_a) connected on series. An electromotive force (E_a) is generated as a result of cutting the lines of flux generated by the permanent magnet, and it has a direction opposite to the direction of the source voltage. The mechanical part consists of moment inertia (J_m) and friction coefficient (B_m). The other parameters are torque constant (K_t) and back emf constant (K_v). The block diagram of PMDC motor with the position output showed in Figure 2.

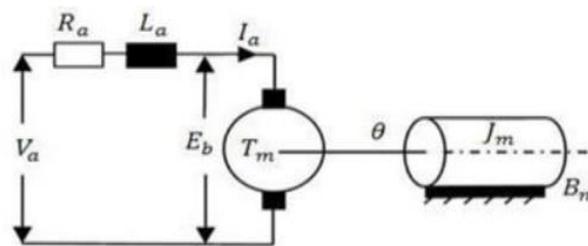


Figure 1. Equivalent circuit of permanent-magnet DC motor[23]

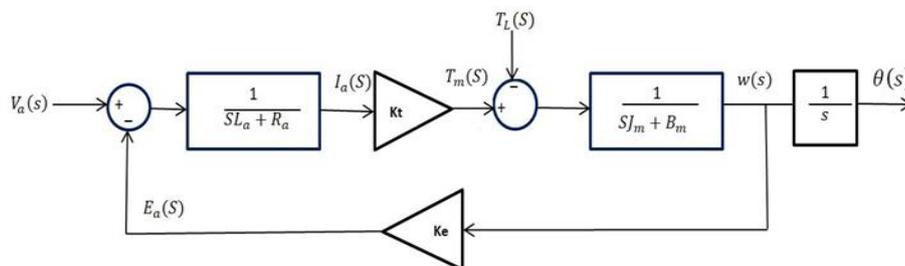


Figure 2. Block diagram of permanent-magnet DC motor [23]

From the block diagram, The Mathematical Model of PMDC motor with position output can be listed in the (1) to (4) [23].

$$v_a(t) = e_a(t) + R_a i_a(t) + L_a \frac{d}{dt} i_a(t) \quad (1)$$

$$e_a(t) = k_e \omega_m(t) \quad (2)$$

$$T_m(t) - T_L = J_m \frac{d}{dt} \omega_m(t) + B_m \omega_m(t) \quad (3)$$

$$T_m(t) = K_t i_a(t) \quad (4)$$

By using Laplace transformation for the (1) to (4) we obtain.

$$V_a(s) = E_a(s) + R_a I_a(s) + sL_a I_a(s) \quad (5)$$

$$E_a(s) = k_e \omega_m(s) \quad (6)$$

$$T_m(s) - T_L = sJ_m \omega_m(s) + B_m \omega_m(s) \quad (7)$$

$$T_m(s) = K_t I_a(s) \quad (8)$$

The overall transfer functions are defined for velocity and position control of the PMDC motor, respectively.

$$\frac{\omega_m(s)}{V_a(s)} = \frac{k_e}{JL_a s^2 + (JR_a + BL_a)s + BR_a + k_e^2} \quad (9)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{k_e}{JL_a s^3 + (JR_a + BL_a)s^2 + (BR_a + k_e^2)s} \quad (10)$$

Where k_e equal to K_t [24].

Table 1 lists the main components and their values of PMDC motor Parameters.

Table 1. The PMDC Parameters [9]

| Motor Parameters | Value |
|----------------------|---------------------------------|
| Torquen constant | $K_t = 2.35 \text{N.m/A}$ |
| Armature inductance | $L_a = 2.61 * 10^{-3} \text{H}$ |
| Armature resistance | $R_a = 2.61 \Omega$ |
| Inertia constant | $J_m = 0.068 \text{kg.m}^2$ |
| Friction coefficient | $B_m = 0.008 \text{N.m.s/rad}$ |
| Back emf constant | $K_v = 2.35 \text{V.s/rad}$ |
| Load torque | $T_L = 17.6 \text{N.m}$ |
| Applied voltage | $V_a = 230 \text{v}$ |

2.2. General structure of system

Figure 3 represents the general structure of the system, which consists of three controllers for current, velocity, and position. The output from the position controller represents the reference velocity, the output from the velocity controller represents the reference current, while the output from the current controller represents the control voltage (vc) and it is compared with a triangular waveform signal that is generated internally in the PWM circuit, producing a KPWM gain that simply represents the PWM controller and the dc-dc switching mode converter (i.e. H bridge). KPWM can be represented by the following equation in Laplace transform after making linearized of the dc-dc converter [24].

$$V_a(s) = \text{KPWM} * V_c(s) \quad (11)$$

One of the benefits of the cascade control system, in addition to the benefits mentioned previously, is the ability to put limits on the reference signals for the system in order to protect the PMDC motor and the power electronic converter. In this paper, limits were placed on the velocity reference by an amount that does not exceed the rated motor velocity, as well as limit, was placed on the reference voltage coming out of the PI

current by 5v, which represents the control voltage (vc) and is compared with a triangular voltage as explained [24].

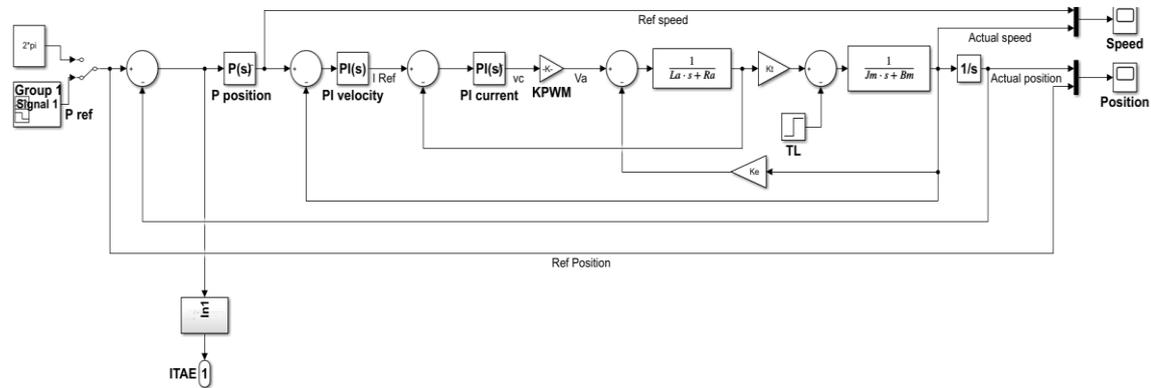


Figure 3. General structure

3. TUNING METHOD

3.1. Particle swarm optimization (PSO)

Particle swarm algorithm (PSO) is proposed by Kennedy and Eberhart and it was modified to improve its performance by adding a new parameter called inertia weight [25]. It which bases on a swarm base for using an observation of social behavior of moving organisms Such as a gathering of fish or a flock of birds.

The algorithm is related to the computational method that optimizes the problem Where flocks of birds aim to find eating behaviors and it uses repeated steps to reach the best solutions. This means that the candidate solutions are particles that move in the search space based on a specific formula above the particle position. Each particle's movement is affected by its local value, and its objective is to reach the best-known positions in the search-space by updating better positions found by other particles.

The algorithm is begun by establishing the starting position and velocity vectors. And each iteration is tried to find the best value by evaluating position and velocity vectors. Every particle has variables and dimensions, and these variables are problems that need to be solved. If the problem consists of five different variables, the particles' dimension should be chosen as five. This algorithm depends mainly on finding the position of each particle with the best local value, as well as finding the best general swarm position in each iteration. The position and speed are updated at each iteration based on (12) and (13) [21], [26].

$$V_{i,j}(t + 1) = W \cdot V_{i,j}(t) + r_1 c_1 [Pbest_{i,j}(t) - X_{i,j}(t)] + r_2 c_2 [Gbest_{i,j}(t) - X_{i,j}(t)] \quad (12)$$

$$X_{i,j}(t + 1) = V_{i,j}(t + 1) + X_{i,j}(t) \quad (13)$$

Where, i=particle index, t=iteration, $V_{(i,j)}(t+1)$ =velocity updated or new velocity, j=dimension number, W=weighted inertia its value between [0-1], $V_{(i,j)}(t)$ =Current velocity, r_1, r_2 =random coefficients there values between [0-1], $X_{(i,j)}(t)$ =current position, c_1, c_2 =acceleration coefficients there values between [0-2] and $X_{(i,j)}(t+1)$ =position updated or new position. The particles in the PSO, position values are updated until the number of iterations is reached to specified value. The particles' changing positions are illustrated in Figure 4 and Table 2 represents the parameters of the PSO that were used in this work. Figure 5 shows the flowchart of the particle swarm optimization algorithm.

Table 2. PSO Parameters

| PSO parameters | Value |
|------------------|--------|
| Iteration | 10 |
| Swarm size | 20 |
| No dimension | 5 |
| Weighted inertia | 0.9 |
| C1, C2 | 2, 1.5 |
| LB, UB | 0, 300 |

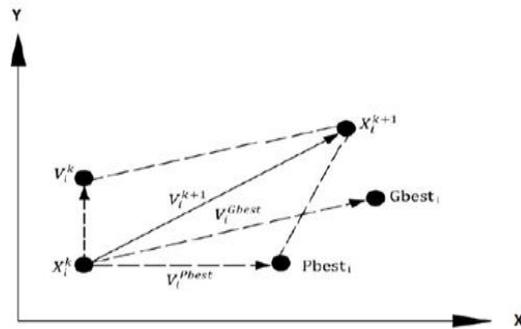


Figure 4. The concept of updating the research point by the PSO [26]

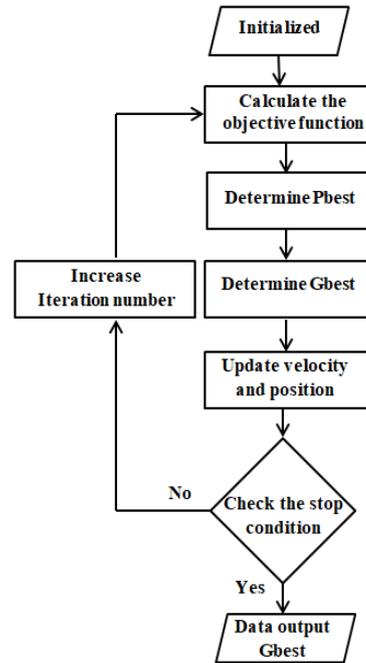


Figure 5. PSO flowchart [27]

3.2. Butterfly optimization algorithm (BOA)

In this paper, a new metaheuristic algorithm inspired by nature called the Butterfly Improvement Algorithm (BOA for short) has been proposed. This algorithm adopts the strategy of searching for food or mating between butterflies from a biological point of view. Butterflies use sense receptors to find the source of food by sensing the scent of the fragrance (nectar) [18], [28].

These receptors are spread over the body parts of the butterfly, such as the antennae and the legs, and they are nerve cells that are found on the surface of the butterfly’s body and are called chemical receptors. The butterfly generates a scent of different intensity according to its fitness, meaning that when the butterfly moves or moves from one place to another, the fragrance spreads across these distances, and other butterflies can sense it, and this is the method of communication between butterflies [18], [28].

The entire concept of sensing and odor processing is based on three important terms, a) the sensory modality, denoted by the symbol *c*, and its value between [0-1], b) stimulus intensity, symbolized by the symbol *I*, and c) the size of the stimulus or force on which the butterfly depends. It is known as power exponent and It is denoted by the symbol *a* and its value is between [0-1].

The natural phenomenon of butterflies is based on two important issues, which are the formulation of the fragrance function and the variation in the intensity of the fragrance. The fragrance function can be formulated as a function of physical density and according to (14).

$$f = cI^a \tag{14}$$

where *f* is the perceived magnitude of the fragrance, i.e., how stronger the fragrance is perceived by other butterflies. *c*, *I* and *a* are explained by (14).

When the butterfly can sense the smell emanating from another butterfly, it will move towards it, and this stage is known as the global search and can be represented by (15), and when the butterfly is unable to sense the smell emanating from any other butterfly, it will move randomly and this stage is called the local search and It can be represented by (16).

$$X_i^{t+1} = X_i^t + (r^2 * g^* - X_i^t) * f_i \tag{15}$$

$$X_i^{t+1} = X_i^t + (r^2 * X_j^t - X_k^t) * f_i \tag{16}$$

Where, $X_i^{(t+1)}$ =new solution vector, X_i^t =solution vector, r =random number (n) between $[0,1]$, g =current best solution, t =iteration number, i =butterfly, X_j^t and X_k^t are j th and k th butterflies from the solution space, and f_i =the perceived magnitude of the fragrance.

The butterflies' search for food or mating can occur on both local and global searches. The butterfly can switch between local search and global search, meaning that it has the decision to move towards the best butterfly or move randomly. The mode to switch between local and global search is called a switch Probability and denoted by the symbol ρ . Table 3 represents the parameters of the BOA that were used in this work.

The butterfly optimization algorithm can be represented in the approximate flowchart shown in Figure 6. The design of any complex system requires the selection of certain criteria that give the best performance. These functions are known as performance indices. In this paper, ITAE was used and it can be represented by (17).

$$ITAE = \int_0^{\infty} t|e(t)|dt \tag{17}$$

Table 3. Parameters of the BOA

| Parameters | Value |
|---------------------------|-------|
| Max iteration | 5 |
| No search agents | 20 |
| No demension | 5 |
| Switch probability ρ | 0.8 |
| Power exponent a | 0.1 |
| Sensory modality | 0.01 |
| LB, UB | 0.300 |

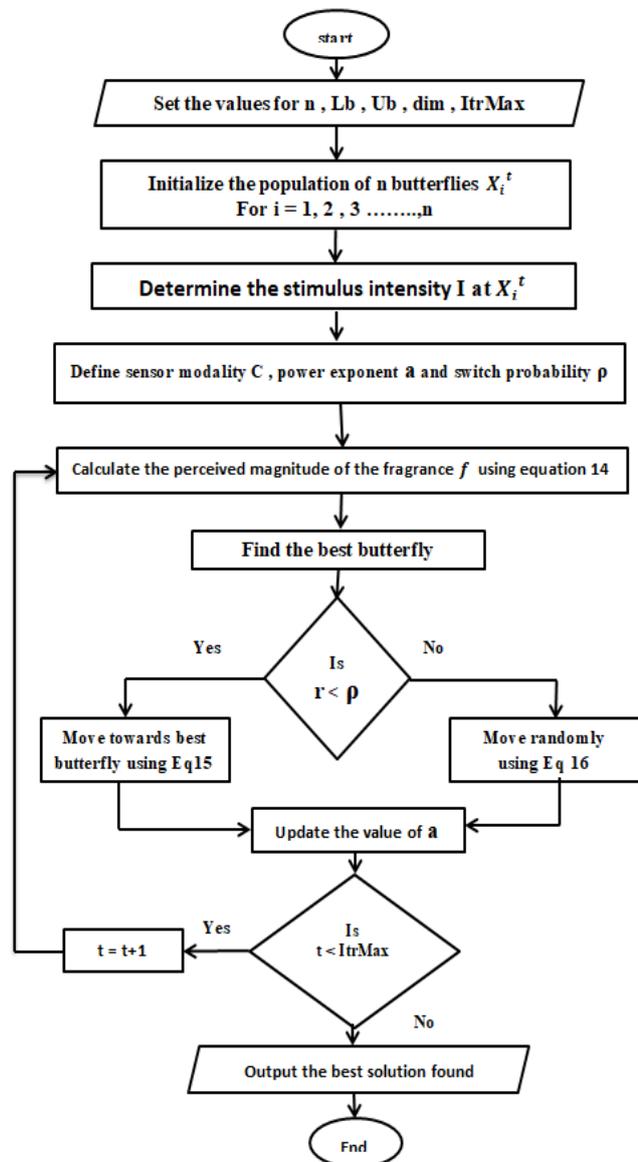


Figure 6. BOA flowchart

4. RESULTS AND COMPARISON

Simulation was performed on the system using both the BOA and PSO methods. The system was tested in cases of loading and non-loading, as well as in the case of a single input or multiple inputs to ensure that the PMDC motor rotates a full rotation, 360 degrees or a half rotation. Table 4 represents the cascade PID parameters extracted by both the BOA and PSO methods, while Table 5 represents the performance parameter values obtained from these two methods.

For the purpose of comparison between BOA and PSO:

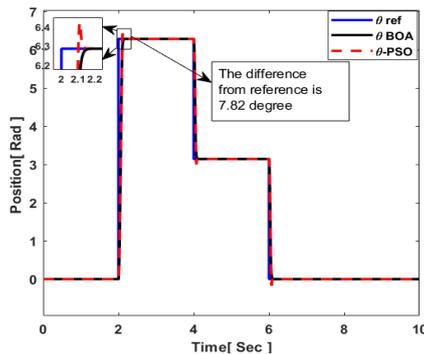
- a. An overshoot was observed when using the PSO method and its value was 2.557%. This indicates a deviation of the position of 7.82 degrees from the reference position while no deviation of the position occurred when using the BOA method see Figure 7(a).
- b. An overshoot and velocity deviation of about 18rad/sec were observed when using the PSO, while no velocity deviation occurred when using the BOA method see Figure 8(a).
- c. When applying full load (17.6N.m) on the system at the fifth second, it was noticed that the system was not affected when using both methods, or was affected imperceptibly, and this indicates the durability of the cascade controller in rejecting the disturbance see Figure 7(b) and Figure 8(b)
- d. The BOA method reached the required values at the second iteration, while the PSO method needed 10 iterations to reach the required values see Figures 9(a) and 9(b).

Table 4. PID parameters values

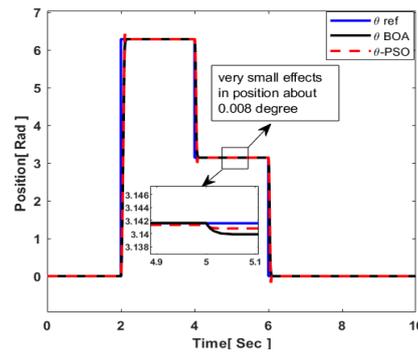
| PID parameters | PSO | BOA |
|----------------|----------|---------|
| KD position | 132.6568 | 77.3994 |
| KP velocity | 102.3941 | 55.5583 |
| KI velocity | 14.2140 | 0.524 |
| KP current | 23.68556 | 74.3044 |
| KI current | 0.1725 | 0.0236 |

Table 5. The values of the performance criteria for each tuning method

| Performance criteria | PSO | BOA |
|----------------------|--------|--------|
| Rise time (ms) | 67.316 | 68.182 |
| Settling time (s) | 0.2 | 0.17 |
| Overshoot | 2.577 | 0% |

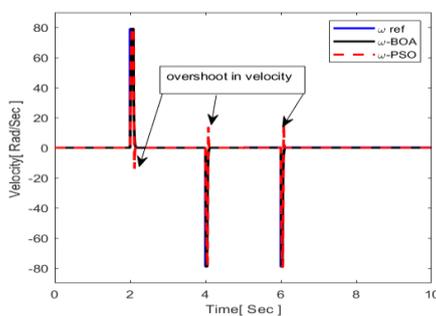


(a)

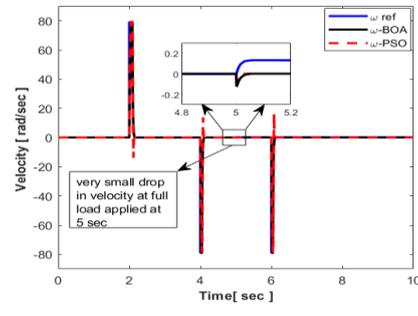


(b)

Figure 7 (a) Position control system at no load case (b) Close up load case position control at fifth second



(a)



(b)

Figure 8. (a) velocity control system at no load case (b) close up load case velocity control at fifth second

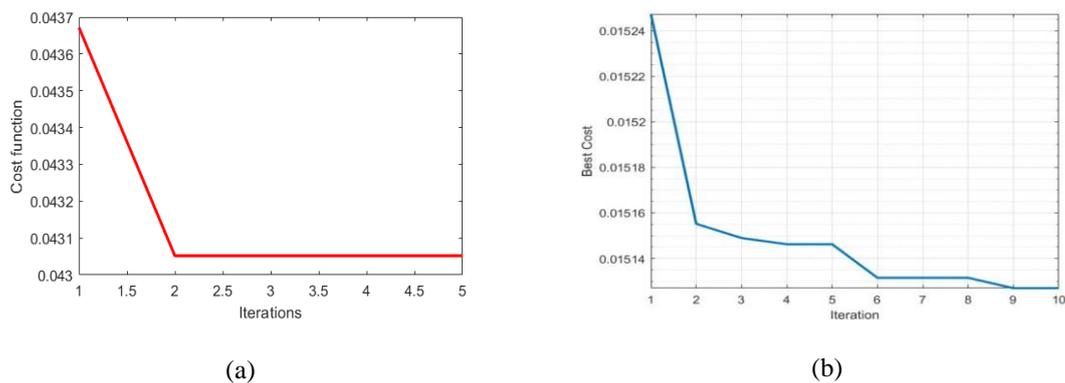


Figure 9. (a) BOA cost function vs iteration (b) PSO cost function vs iteration

The simulation results showed that there was no position deviation or overshoot in velocity or position when using the parameters extracted by the BOA in addition to accurate tracking of the path. On the other hand, when using the PSO, a position deviation of 7.82 degrees and an overshoot of 2.557% were observed, in addition to an overshoot in the velocity. The simulation results and the comparison mentioned above showed that the BOA algorithm gave better results than the PSO algorithm.

5. CONCLUSION

In this paper, two optimization algorithms are used to extract the PID gains for the cascade controller of a PMDC motor. These algorithms are PSO and BOA. The simulation results in the case of using PSO showed a clear deviation of the position from the reference position about 7.82 degrees, in addition, an overshoot of 2.557%. On the other hand, no position deviation or overshoot occurred when using the BOA algorithm. Also, there is no overshoot in the velocity when using the BOA algorithm, which led to an accurate tracking of the path. As a result, the BOA algorithm is better than the PSO algorithm.

REFERENCES

- [1] T. A. Bigelow, *Electric Circuits, Systems, and Motors*, Springer, 2020.
- [2] J. A. Momoh, *Energy Processing and Smart Grid*, John Wiley & Sons, 2018.
- [3] P. Krause, O. Wasynczuk, S. D. Sudhoff, and S. D. Pekarek, *Analysis of Electric Machinery and Drive Systems*, vol. 75. John Wiley & Sons, 2013.
- [4] B. L. Theraja, *A textbook of electrical technology*, S. Chand Publishing, 2008.
- [5] W. H. Ali, M. N. O. Sadiku, and S. Abood, *Fundamentals of Electric Machines: A Primer with MATLAB: A Primer with MATLAB*. CRC Press, 2019.
- [6] J. Bae, K. Cho, and D.-H. Lee, "Parallel position control scheme of permanent magnet DC motors with a low-resolution sensor," *2020 IEEE International Conference on Industrial Technology (ICIT)*, 2020, pp. 199-204, DOI: 10.1109/ICIT45562.2020.9067269
- [7] Z. Adel, A. A. Hamou, and S. Abdellatif, "Design of real-time PID tracking controller using arduino mega 2560 for a permanent magnet DC motor under real disturbances," *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, 2018, pp. 1-5, DOI: 10.1109/CISTEM.2018.8613560
- [8] M. F. Cankurtaran and A. E. Kocamis, "Sensorless speed control of PMDC motor with cascade PI controller," *2019 International Symposium ELMAR*, 2019, pp. 203-206, DOI: 10.1109/ELMAR.2019.8918654
- [9] Taha. N. Gücin, M. Bıberoğlu, B. Fincan, and M. O. Gülbahçe, "Tuning cascade PI (D) controllers in PMDC motor drives: A performance comparison for different types of tuning methods," *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*, 2015, pp. 1061-1066, doi: 10.1109/ELECO.2015.7394556
- [10] L. Wang, *PID control system design and automatic tuning using MATLAB/Simulink*, John Wiley & Sons, 2020.
- [11] G. L. Raja and A. Ali, "Series cascade control: An outline survey," *2017 Indian Control Conference (ICC)*, 2017, pp. 409-414, DOI: 10.1109/INDIANCC.2017.7846509
- [12] Y. Xie, J. Jin, X. Tang, B. Ye, and J. Tao, "Robust cascade path-tracking control of networked industrial robot using constrained iterative feedback tuning," *IEEE Access*, vol. 7, pp. 8470-8482, 2018, DOI: 10.1109/ACCESS.2018.2889702
- [13] S. Kumar, D. Roy, and M. Singh, "A fuzzy logic controller based brushless DC motor using PFC cuk converter," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 10, no. 4, pp. 1894-1905, 2019, DOI: 10.11591/ijpeds.v10.i4.pp1894-1905

- [14] A. H. Ahmed, B. Abd El Samie, and A. M. Ali, "Comparison between fuzzy logic and PI control for the speed of BLDC motor," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 9, no. 3, pp. 1116-1123, 2018, DOI: 10.11591/ijpeds.v9.i3.pp1116-1123
- [15] A. A. Obed, A. L. Saleh, and A. K. Kadhim, "Speed performance evaluation of BLDC motor based on dynamic wavelet neural network and PSO algorithm," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 10, no. 4, pp. 1742-1750, 2019, DOI: 10.11591/ijpeds.v10.i4.pp1742-1750
- [16] Y. Ahmed and A. Hoballah, "Adaptive filter-FLC integration for torque ripples minimization in PMSM using PSO," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 10, no. 1, p. 48-57, 2019, DOI: 10.11591/ijpeds.v10.i1.pp48-57
- [17] M. S. Amiri, M. F. Ibrahim, and R. Ramli, "Optimal parameter estimation for a DC motor using genetic algorithm," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 11, no. 2, pp. 1047-1054, 2020, DOI: 10.11591/ijpeds.v11.i2.pp1047-1054
- [18] L. El Hajjami, E. M. Mellouli, and M. Berrada, "Optimal PID control of an autonomous vehicle using Butterfly Optimization Algorithm BOA," *Proceedings of the 4th international conference on big data and internet of things*, 2019, pp. 1-5, DOI: 10.1145/3372938.3372980
- [19] M. M. Kamal, L. Mathew, and S. Chatterji, "Speed control of brushless DC motor using intelligent controllers," *2014 Students Conference on Engineering and Systems*, 2014, pp. 1-5, DOI: 10.1109/SCES.2014.6880121
- [20] Wisam. N. A.-D. Abed, A. H. Saleh, and A. S. Hameed, "Speed control of PMDCM based GA and DS techniques," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 9, no. 4, pp. 1467-1475, 2018, DOI: 10.11591/ijpeds.v9n4.pp1467-1475
- [21] Fadhil. A. Hasan and L. J. Rashad, "Fractional-order PID controller for permanent magnet DC motor based on PSO algorithm," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 10, no. 4, pp. 1724-1733, 2019, DOI: 10.11591/ijpeds.v10.i4.pp1724-1733
- [22] A. R. Ajel, H. M. A. Abbas, and M. J. Mnati, "Position and speed optimization of servo motor control through FPGA," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 11, no. 1, pp. 319-327, 2021, DOI: 10.11591/ijece.v11i1.pp319-327
- [23] M. Namazov and O. Basturk, "DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods," *TJFS: Turkish Journal of Fuzzy Systems*, vol. 1, no. 1, pp. 36-54, 2010.
- [24] N. Mohan, *Electric Drives*, 2003.
- [25] M. A. Arasomwan and A. O. Adewumi, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization," *Sci. World J.*, vol. 2013, pp. 1-12, 2013, DOI: 10.1155/2013/860289
- [26] H. R. Yazgan, F. Yener, S. Soysal, and A. E. Gür, "Comparison performances of PSO and GA to tuning PID controller for the DC motor," *Sak. Univ. J. Sci.*, vol. 23, no. 2, pp. 162-174, 2019, DOI: 10.16984/saufenbilder.376464
- [27] I. J. Hasan, M. R. Ab Ghani, and C. K. Gan, "Optimum distributed generation allocation using PSO in order to reduce losses and voltage improvement," *3rd IET International Conference on Clean Energy and Technology (CEAT)*, 2014, DOI: 10.1049/cp.2014.1476
- [28] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715-734, 2019, DOI: 10.1007/s00500-018-3102-4