

## Comparison of cascade P-PI controller tuning methods for PMDC motor based on intelligence techniques

Kareem Ghazi Abdulhussein, Naseer Majeed Yasin, Ihsan Jabbar Hasan

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

### Article Info

#### Article history:

Received Jan 21, 2020

Revised Jun 28, 2021

Accepted Jul 20, 2021

#### Keywords:

Cascade P-PI controller

Classical method

Genetic algorithm

Particle swarm optimization

PMDC

### ABSTRACT

In this paper, there are two contributions: The first contribution is to design a robust cascade P-PI controller to control the speed and position of the permanent magnet DC motor (PMDC). The second contribution is to use three methods to tuning the parameter values for this cascade controller by making a comparison between them to obtain the best results to ensure accurate tracking trajectory on the axis to reach the desired position. These methods are the classical method (CM) and it requires some assumptions, the genetic algorithm (GA), and the particle swarm optimization algorithm (PSO). The simulation results show the system becomes unstable after applying the load when using the classical method because it assumes cancellation of the load effect. Also, an overshoot of about 3.763% is observed, and a deviation from the desired position of about 12.03 degrees is observed when using the GA algorithm, while no deviation or overshoot is observed when using the PSO algorithm. Therefore, the PSO algorithm has superiority as compared to the other two methods in improving the performance of the PMDC motor by extracting the best parameters for the cascade P-PI controller to reach the desired position at a regular speed.

*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A permanent magnet DC motor (PMDC) is a simple type of DC motor where field windings have been replaced by permanent magnets. This motor works with the same basic principle as a shunt-connected motor but the difference between them is that the permanent magnet generates the required flux instead of the field windings [1], [2]. The power of these motors is less than that of the shunt-connected motors because the flux generated by the permanent magnet is less than the flux generated by the field windings. Therefore, most types of small DC motors are PMDC motors and these motors operate at high speed and low torque [1], [3]. There are many applications for the PMDC motor, such as moving windows in cars and front areas of cars. It is also used in children's toys and household appliances like food mixers and others [4], [5] and its important use in computer numerical control (CNC) machines as an actuator, electric vehicles, and robotics [6], [7].

The main problem statement in this paper is how to get the best parameter values for the proportional integral derivative (PID) cascade controller to give the most accurate results for tracking the trajectory of the reference position to reaching the desired position at a regular speed. To address this problem, a comparison was made between the three methods classical method (CM), genetic algorithm (GA), and particle swarm optimization algorithm (PSO) to extract the best parameters of the cascading controller, in addition to using the objective function, integral time absolute error (ITAE) to reduce the error between the

input and output of this system. The cascade P-PI controller used in this paper consists of three controllers: the current controller as an inner loop, the speed controller, and the position controller as outer loops [8]. P, PI, and PI controllers are used for position, speed, and current respectively [9].

The purpose of using the cascade P-PI controller is due to many reasons, the most important of which is robustness to reduce or reject external disturbances and back to the steady-state in a short time [10]-[12]. There are several methods to tuning the parameters of the cascade P-PI controllers such as the ziegler-nichols (Z-N) method, cohen-coon method, neural network, and fuzzy logic [13], [14]. Particle swarm optimization (PSO) algorithm [15], [16] and genetic algorithm [17].

To compare this work with other researcher's work in the same field, the following literature review was made: In 2014, Mustafa *et al.* [18] presented a brushless DC (BLDC) motor speed control system using GA. In 2015, Taha *et al.* [9] used three methods to control the cascade control system. In 2018, Wisam *et al.* [19] presented a system for controlling PMDC speed using GA and direct search (DS) algorithms. In 2019, Fadhel *et al.* [20] used a fractional PID controller to control PMDC speed based on PSO. In 2021, Ahmed *et al.* [21] presented a system to control the position and speed of a servo motor. This paper is organized as: The second section contains the mathematical model of the PMDC motor and explaining the general structure of the system, in the third section, the three tuning methods are explained with the objective function (ITAE). The fourth section contains the results and comparison, and the final section contains the conclusion.

## 2. MATHEMATICAL MODEL AND GENERAL STRUCTURE

### 2.1. Mathematical model of PMDC motor

Figure 1 shows the equivalent circuit of a PMDC motor consisting of an armature resistance ( $R_a$ ) and inductance ( $L_a$ ) connected in series. The back emf ( $E_a$ ) is generated when the flux lines generated by the permanent magnet are cutting and its direction opposite to the direction of the applied voltage. While the mechanical part consists of coefficient of friction ( $B_m$ ) and moment of inertia ( $J_m$ ). In addition to other parameters are the back emf constant ( $K_v$ ) and torque constant ( $K_t$ ). All components or parameters of this PMDC motor and their values are shown in Table 1 and Figure 2 shows the block diagram of PMDC motor.

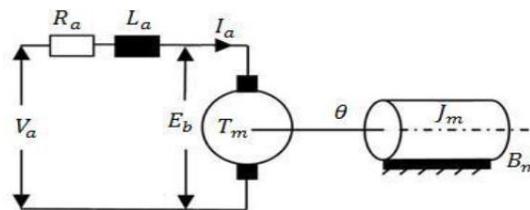


Figure 1. Equivalent circuit of PMDC motor [22]

Table 1. PMDC parameters [9]

Motor parameters	Value
Torque constant	$K_t=2.35 \text{ Nm/A}$
Armature inductance	$L_a=2.61 \times 10^{-10} \text{ H}$
Armature resistance	$R_a=2.61 \text{ } \Omega$
Inertia of the motor	$J_m=0.068 \text{ kg.m}^2$
Friction constant	$B_m=0.008 \text{ Nms/rad}$
Back emf constant	$K_v=2.35 \text{ Vs/rad}$
Nominal load	$T_l=17.6 \text{ Nm}$
Nominal voltage	$V_a=230 \text{ v}$

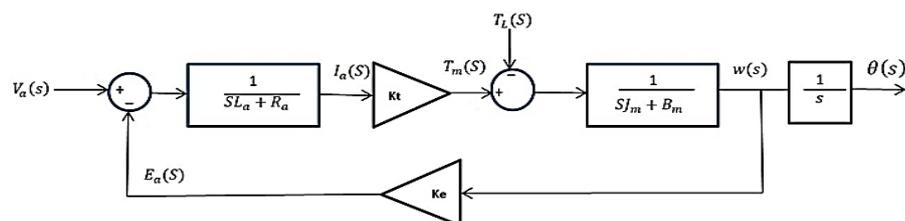


Figure 2. The block diagram of PMDC motor [22]

The electrical and the mechanical equations corresponding to block diagram from Figure 2 are [22].

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

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

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

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

By using Laplace transformation for the (1)-(4) we obtain;

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

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

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

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

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

$$\frac{\omega_m(s)}{V_a(s)} = \frac{k_e}{J L_a S^2 + (J R_a + B L_a) S + B R_a + k_e^2} \quad (9)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{k_e}{J L_a S^3 + (J R_a + B L_a) S^2 + (B R_a + k_e^2) S} \quad (10)$$

Where  $k_e$  is equal to  $K_t$  [23] and

- $R_a$  = armature resistance ( $\Omega$ )
- $L_a$  = armature inductance (H)
- $E_a$  = electromotive force or back emf (v)
- $\theta(s)$  = actual position (rad)
- $V_a$  = nominal voltage (v)
- $J_m$  = moment inertia ( $\text{kg.m}^2$ )
- $T_L$  = nominal load torque (Nm)
- $K_t$  = torque constant (Nm/A)
- $K_v$  = back emf constant (v.sec/rad)
- $B_m$  = friction coefficient (Nm.s/rad)
- $\omega_m$  = motor velocity (rad/sec)

## 2.2. General structure of the system

The general structure of the system consists of a cascade P-PI controller as shown in Figure 3. This controller consists of three (P, PI) controllers for current, speed, and position. The output of the position controller represents the reference speed, the output of the speed controller represents the reference current, and the output of the current controller represents the control voltage (VC).

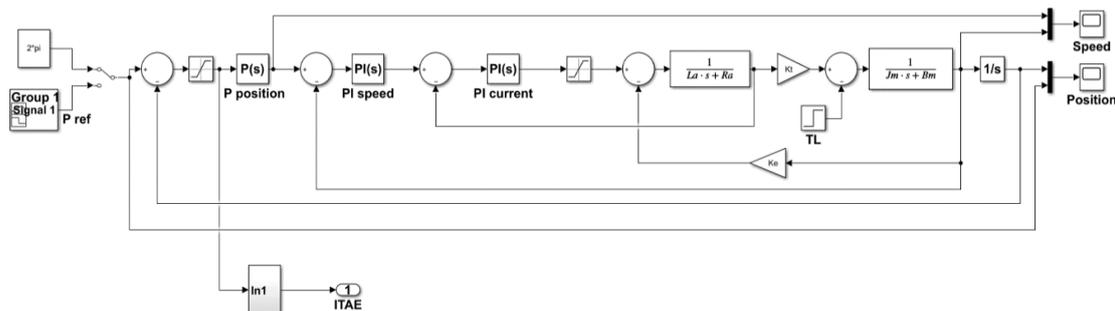


Figure 3. General structure of cascade PID control system for PMDC

An advantage of the cascade control system, in addition to the other advantages, is the ability to set limits in order to protect the PMDC motor and the power electronic converter. In this paper, limits are placed on the speed reference by an amount not exceeding the motor's rated speed. As well as the limit, placed on the reference voltage exiting the PI current not exceeding the motor's applied voltage [23].

### 3. TUNING METHODS

#### 3.1. Classical method (CM)

Figure 4 represents the first control loop or the inner control loop. Since this method assumes some assumptions to simplify the circuit, the effect of the torque load is neglected, and the effect of  $E_a$  is neglected because the value of  $J_m$  is high [9], [23]. As a result, the simplified current control loop as shown in Figure 5.

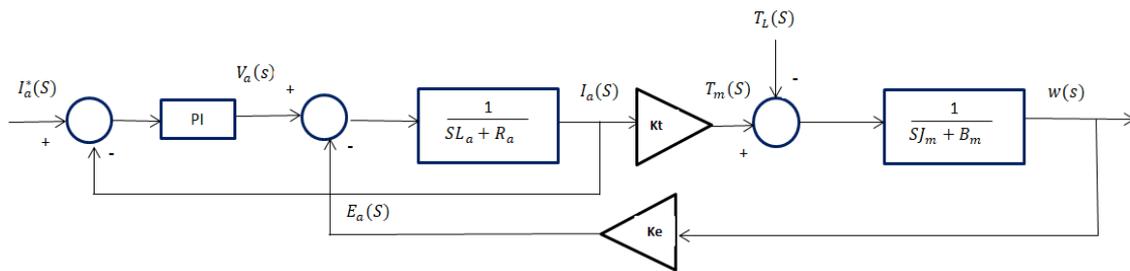


Figure 4. Inner current control loop

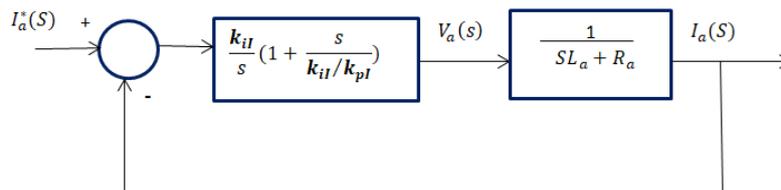


Figure 5. Simplified inner current control loop

From Figure 5 the transfer function of the current control loop,  $G_{I_{ol}}(s)$ , and it can be written like in (11) where,  $k_{il}$  is the integral gain,  $k_{pl}$  is the proportional gain of the controller, and the electrical time constant  $\tau_e$ , is calculated using (12).

$$G_{I_{ol}}(s) = \frac{k_{il}}{s} \left( 1 + \frac{s}{k_{il}/k_{pl}} \right) * \left( \frac{1/R_a}{1+s\tau_e} \right) \quad (11)$$

$$\tau_e = \frac{L_a}{R_a} \quad (12)$$

The (13) is used to cancel the motor pole, which is formulated in (11).

$$\frac{k_{il}}{k_{pl}} = \frac{1}{\tau_e} \quad (13)$$

The cancellation of the pole in the motor transfer function can be illustrated as:

$$G_{I_{ol}} = \frac{k_{il}}{s} \left( 1 \frac{s}{k_{il}/k_{pl}} \right) * \left( \frac{1/R_a}{1+s\tau_e} \right) \text{ OR } G_{I_{ol}} = \frac{k_{il}}{s} (1+s\tau_e) * \left( \frac{1/R_a}{1+s\tau_e} \right)$$

Finally, in the open-loop transfer function, the bandwidth (crossover) frequency ( $\omega_{ci}$ ) It is represented by (14).

$$\omega_{ci} = \frac{k_{iI}}{R_a} \tag{14}$$

Also, the crossover frequency of the current open-loop can be calculated from the following relationship  $\omega_{ci} = 2\pi f_{ci}$  and its value about ten times smaller than the switching frequency for DC-DC converter [9]. So, the parameter values of the PI current controller can be found from (13), (14). To find the speed loop parameters it is assumed that the closed current loop is ideal for design purposes and is represented by the unity [23] as shown in Figure 6.

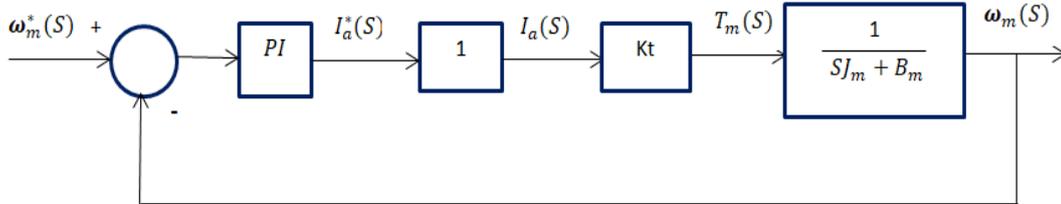


Figure 6. Outer speed loop

From Figure 6, the transfer function of this loop  $G_{s\ ol}(s)$ , can be represented in (15).

$$G_{s\ ol}(s) = \frac{k_{iS}}{s} \left( 1 + \frac{s}{k_{iS}/k_{pS}} \right) * \left( \frac{1/B_m}{1+s\tau_m} \right) \tag{15}$$

Where

$$\tau_m = \text{mechanical time constant} = \frac{J_m}{B_m} \tag{16}$$

Just like the inner current control loop the pole in the mechanical part of the motor will be canceled as show in (17). Here,  $k_{iS}$  is the integral gain and,  $k_{pS}$  is the proportional gain of the controller.

$$\frac{k_{iS}}{k_{pS}} = \frac{1}{\tau_m} \tag{17}$$

The bandwidth (crossover) frequency of the speed control loop,  $\omega_{cs}$  can be chosen to be ten times lower than  $\omega_{ci}$  and it can be represented by (18).

$$\omega_{cs} = \frac{k_{iS} \cdot k_t}{B_m} \tag{18}$$

Thus,  $k_{iS}$  and  $k_{pS}$  can be calculated using (16) and (18).

Finally, the last outer loop is the position control loop it can be illustrated in Figure 7. To find the  $k_{pP}$  parameter of the position loop it is assumed that the speed loop is perfect and is represented by unity [23] as shown in Figure 7 the open-loop transfer function for position control is given in the (19). The value of the position parameter  $k_{pP}$  can be calculated from (20) where the bandwidth frequency ( $\omega_{ps}$ ) is chosen to be ten times smaller than ( $\omega_{cs}$ ).

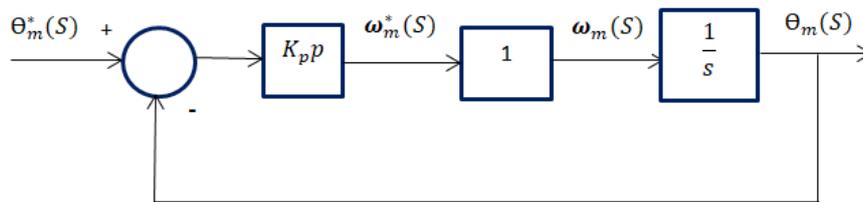


Figure 7. Outer position loop

$$G_{P_{ol}}(s) = \frac{k_{pP}}{s} \quad (19)$$

$$\omega_{ps} = k_{pP} \quad (20)$$

### 3.2. 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 [24]. The PSO algorithm bases on swarm intelligence techniques for the observation of social behavior of moving organisms such as a gathering of fish or birds. This algorithm is related to the computational method to solve the problem, for example, bird flocks aim to find eating behaviors and use 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 speed vectors. At each iteration, the best value is determined by evaluating position and speed 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. Each particle's best value is called a local best value and recorded into the P best matrix. After each iteration, the best value for each particle is updated if the best new value is found to control each current particle and the previous positions. Besides, the position is affirmed as the global best after controlling for the best matrix values at each iteration which continues until it reaches the specified number, as the last updated value for the best position represents the optimal value. 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 (21) and (22) [20], [25].

$$V_{i,j}(t+1) = W \cdot V_{i,j}(t) + r_1 c_1 [Pbest_{i,j}(t) - X_{i,j}(t)] + r_2 c_2 [Gbest_{i,j}(t) - X_{i,j}(t)] \quad (21)$$

$$X_{i,j}(t+1) = V_{i,j}(t+1) + X_{i,j}(t) \quad (22)$$

Where

$i$  = Particle index

$j$  = Dimension number

$t$  = Iteration

$V_{i,j}(t)$  = Current velocity

$X_{i,j}(t)$  = Current position

$V_{i,j}(t+1)$  = Velocity updated or new velocity

$X_{i,j}(t+1)$  = Position updated or new position

$W$  = Weighted inertia its value between 0 and 1

$r_1, r_2$  = Random coefficients there values between 0 and 1

$c_1, c_2$  = Acceleration coefficients there values between 0 and 2

In Table 2 are the parameters of the PSO used in this work and Figure 8 shows the flowchart of the particle swarm optimization algorithm.

Table 2. PSO parameters

PSO Parameters	Value
Iteration	100
Swarm size	20
No. dimension	5
Weighted inertia	0.9
C1	2
C2	1.5
LU, UB	0, 300

### 3.3. Genetic algorithm

In 1970, Holland introduced the genetic algorithm (GA). This method is based on the principle of evolution and natural selection. As an optimizer, this algorithm begins by selecting a specific number of

chromosomes from a primary set, and each chromosome represents a solution to a specific problem and its performance is evaluated by the fitness function [18]. The present parents produce new offspring called the new generation, as this new generation is supposed to reach a better solution than previous parents, due to the process of evolution. This mechanism is repeated for several successive generations until one of them obtains the optimal solution for the system. Since the process is associated with many phases, Figure 9 shows a flowchart of the genetic algorithm [26]. Table 3 represents the parameters of the GA that were used in this work.

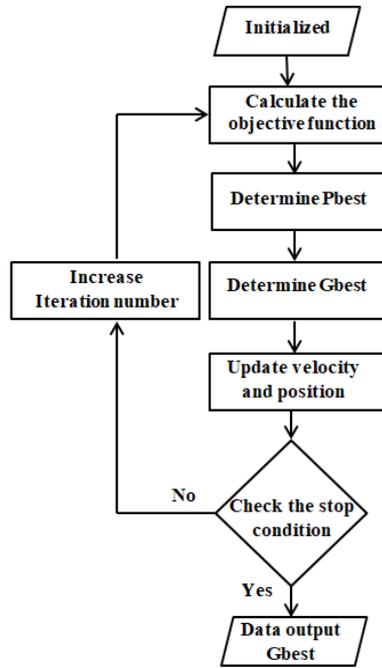


Figure 8. PSO flowchart [27]

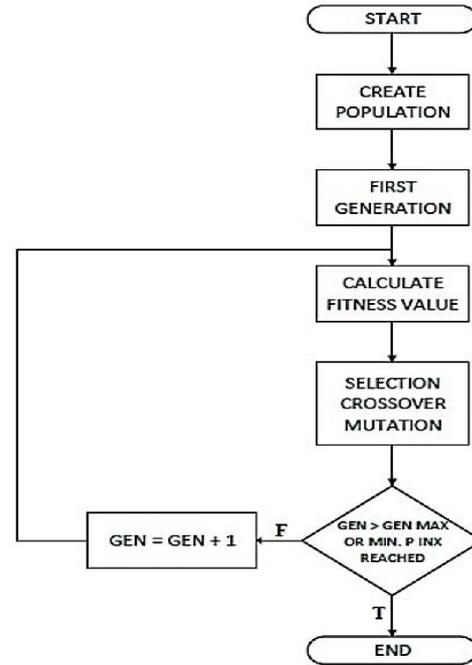


Figure 9. Genetic algorithm process flowchart [26]

Table 3. GA parameters

GA Parameters	Value
Generation	100
Population size	20
No. dimension	5
Crossover	0.6
LU, UB	0, 300

When designing any control system for a particular physical process, certain criteria must be chosen to provide the best performance. These criteria are known as objective functions. These functions reduce the error between the input and output and have several types. In this work, a performance indicator was used to reduce the error in the position signal called an integral absolute time error (ITAE) and can be represented by (23).

$$ITAE = \int_0^{\infty} t |e(t)| dt \quad (23)$$

#### 4. RESULTS AND COMPARISON

The simulation of the system was performed using MATLAB 2019, and the system was tested after entering the cascade P-PI parameters extracted from the three tuning methods. Table 4 shows the values of these parameters extracted from each method. The system has also been tested in other cases, such as the load condition, no-load, and single or multiple reference position to ensure the PMDC motor rotates 360 degrees or several revolutions. Table 5 shows the performance parameters for the system at load case, which were obtained from each tuning method.

Table 4. PID parameters values

PID Parameters	CM	GA	PSO
KP. Position	125.6637	244.3101	173.1189
KP. Speed	36.3623	98.7314	269.2364
KI. Speed	4.2779	291.317	3.6981
KP. current	32.7982	101.5191	216.5567
KI. current	32798	268.9725	23.8736

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

Performance criteria	CM	GA	PSO
Rise time	2.129 sec	2.033 sec	2.053 sec
Settling time	0.3 sec	0.2 sec	0.2 sec
overshoot	7.558%	3.763%	0

Through the simulation results attached in the appendix, and for the purpose of obtaining the best results, the following comparison was made between the three tuning methods:

- At no-load case, high overshoot is observed in speed when using the CM method as shown in Figures 10(a) and 10(b), and after applying the full load at the fifth second, the system becomes unstable because this method assumes neglecting the load effect to simplify the system as shown in Figures 11(a) and 11(b).
- When using the GA algorithm and at the no-load case, an overshoot was observed about 3.646 %, and a deviation of  $12.03^\circ$  was observed from the desired position, as well as a high overshoot was observed in the speed as shown in Figures 12(a) and 12(b). After applying full load, a very small deviation of about 0.0003 rad was observed. This is an indication of the robustness of the cascade P-PI controller in rejecting external disturbances as shown in Figures 13(a) and 13(b).

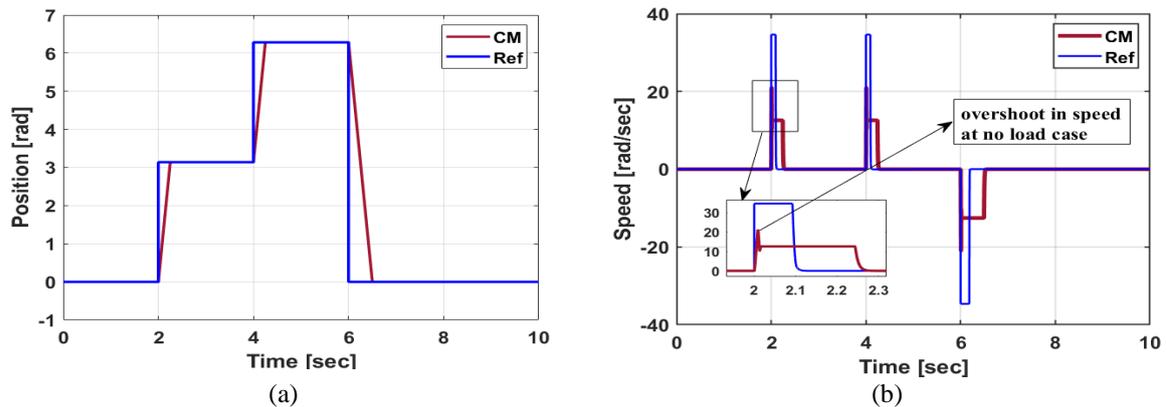


Figure 10. These figures are: (a) position control at no load case, and (b) speed control at no load case

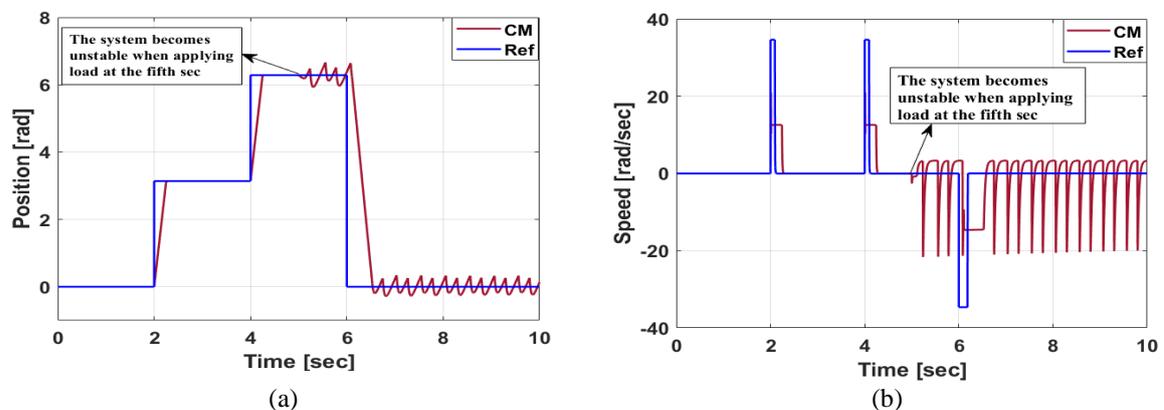


Figure 11. These figures are: (a) position control at load case, and (b) speed control at load case

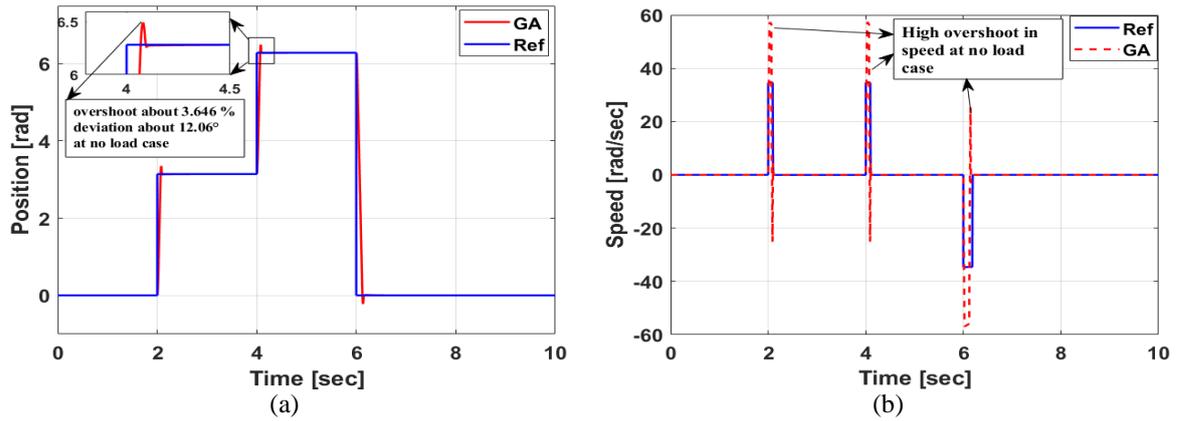


Figure 12. These figures are: (a) position control at no load case and (b) speed control at no load case

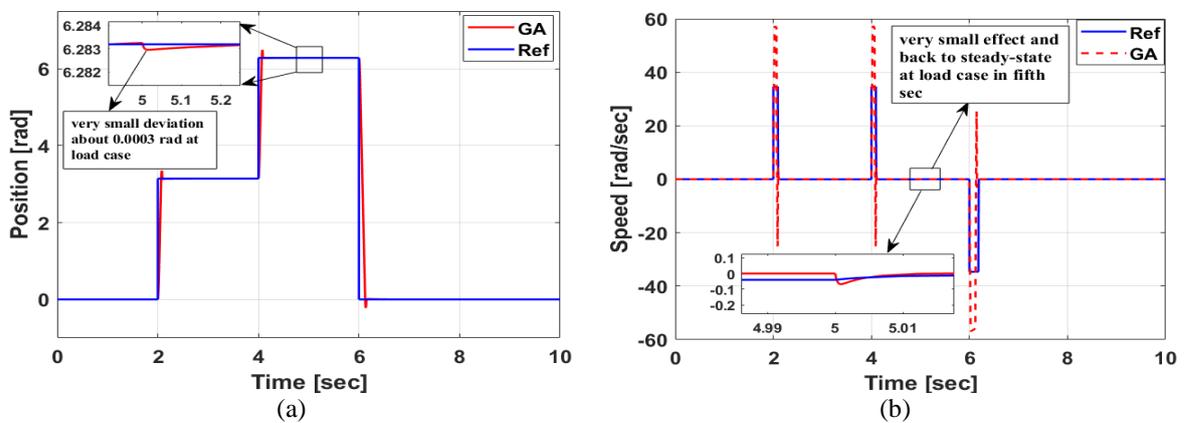


Figure 13. These figures are: (a) position control at load case, and (b) speed control at load case

c) When using the PSO algorithm, there was no deviation from the position or overshoot in speed or position in both cases load and no-load in addition to the accurate tracking trajectory to reach the position with high accuracy as shown in Figures 14(a), (b) and Figures 15(a), (b). The negative speed means that the PMDC motor is rotating in the opposite direction. Therefore, the simulation results and the comparison proved that the particle swarm optimization algorithm gave better results than the other two methods.

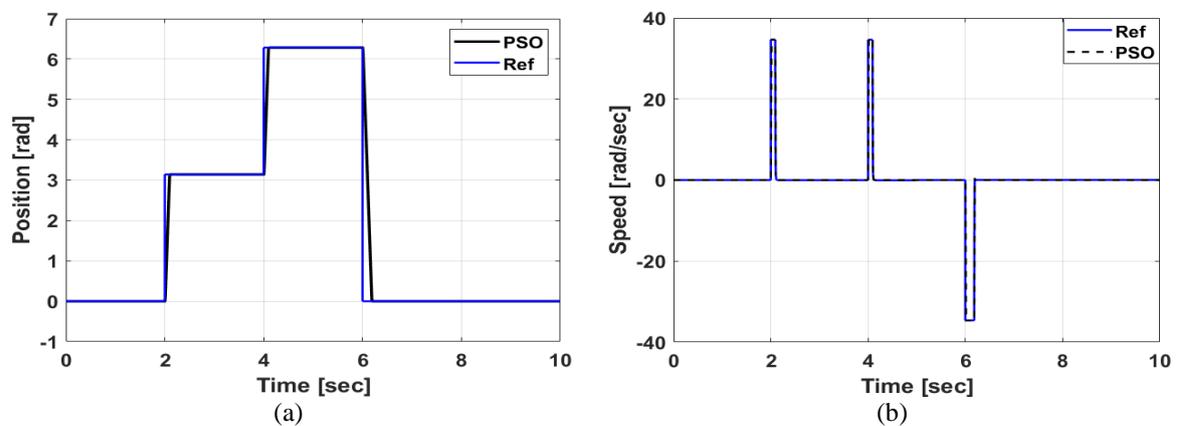


Figure 14. These figures are: (a) position control at no load case, and (b) speed control at no load case

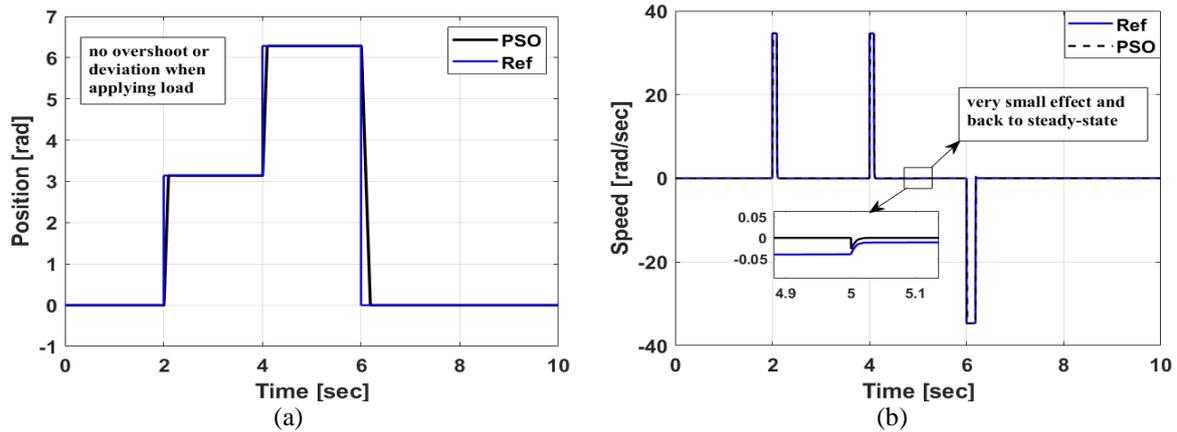


Figure 15. These figures are: (a) position control at load case, and (b) speed control at load case

## 5. CONCLUSION

This paper presents a comparison between two intelligence methods which are PSO and GA algorithms in addition to the traditional method CM. These methods are used to extract the best parameters of the cascade P-PI controller to control the speed and position of the PMDC motor to reach the desired position. The simulation results prove the effectiveness of the PSO algorithm in terms of no overshoot as compared with CM and GA methods which have an overshoot about 7.558% and 3.763%, respectively. Also, there is no deviation in position when using the PSO algorithm while a deviation about 12.03 degrees is observed when using GA algorithm. So, the PSO algorithm has superiority in improving the overall performance of the PMDC motor by extracting the best parameters for the cascade P-PI controller to reach the desired position as compared to the other two methods.

## 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," in *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," in *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," in *2019 International Symposium ELMAR*, 2019, pp. 203–206, doi: 10.1109/ELMAR.2019.8918654.
- [9] T. N. Gücin, M. Biberoglu, B. Fincan, and M. O. Gulbahce, "Tuning cascade PI (D) controllers in PMDC motor drives: A performance comparison for different types of tuning methods," in *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," in *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 Systems (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 Systems (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 Systems (IJPEDS)*, vol. 10, no. 4, pp. 1742–1750, 2019, doi: 10.11591/ijpeds.v10.i4.1742-1750.
- [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 Systems (IJPEDS)*, vol. 10, no. 1, pp. 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 Systems (IJPEDS)*, vol. 11, no. 2, pp. 1047-1054, 2020, doi: 10.11591/ijpeds.v11.i2.pp1047-1054.
- [18] M. M. Kamal, L. Mathew, and S. Chatterji, "Speed control of brushless DC motor using intelligent controllers," in *2014 Students Conference on Engineering and Systems*, 2014, pp. 1–5, doi: 10.1109/SCES.2014.6880121.
- [19] W. 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 Systems (IJPEDS)*, vol. 9, no. 4, pp. 1467-1475, 2018, doi: 10.11591/ijpeds.v9.i4.pp1467-1475.
- [20] F. 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 Systems (IJPEDS)*, vol. 10, no. 4, pp. 1724-1733, 2019, doi: 10.11591/ijpeds.v10.i4.1724-1733.
- [21] A. R. Ajel, H. M. A. Abbas, and M. J. Mnati, "Position and speed optimization of servo motor control through FPGA.," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 319-327, 2021, doi: 10.11591/ijece.v11i1.pp319-327.
- [22] M. Namazov and O. Basturk, "DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods," *TJFS: Turkish Journal of Fuzzy Systems*, pp. 36-54, 2010.
- [23] N. Mohan, "Electric drives," CRC press, 2003.
- [24] 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, 2013, doi: 10.1155/2013/860289.
- [25] 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.
- [26] E. Flores-Morán, W. Yáñez-Pazmiño, and J. Barzola-Monteses, "Genetic algorithm and fuzzy self-tuning PID for DC motor position controllers," in *2018 19th International Carpathian Control Conference (ICCC)*, 2018, pp. 162–168, doi: 10.1109/CarpathianCC.2018.8399621.
- [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.