Bitlis Eren Üniversitesi Fen Bilimleri Dergisi

BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 13 NO: 1 PAGE: 192-204 YEAR: 2024
DOI:10.17798/bitlisfen.1370223

Design and Implementation of an Optimized PID Controller for Two-Limb Robot Arm Control

Said MÜFTÜ[1*], Barış GÖKÇE[2]

[1] *Pamukkale University, Department of Mechatronics Engineering, Denizli,20160, Türkiye*
[2]*Necmettin Erbakan University, Department of* Mechatronics *Engineering, Konya, 42090, Türkiye*
(ORCID: 0000-0001-5621-7805) (ORCID: 0000-0001-6141-7625)

Keywords: DC motor, Optimization, PID Control, Robot Arm.

Abstract
Advances in technology have increased the use of robot arms and led to more research and development on robot arms. Controllability, which is the main focus of the studies on robot arms, generally provides speed and precision to robot arms. In this study, a two-limbed robot arm is controlled using the MATLAB support package for Arduino Hardware, and a microcontroller is used to optimize the control of this robot arm with a PID controller. In addition, in the direct current (DC) brushed motor system, the transfer function was obtained using values from the motor data sheet. Feedback is provided to this control system with a Hall-effect encoder. For the square reference tracking of the gripper end of the two-armed robot arm, the controller parameters were obtained by particle swarm (PSO), artificial bee colony (ABC), and chaos game optimization (CGO) algorithms, and these parameters were applied to the robot arm. The CGO algorithm, which is one of the methods in the literature and has become popular in recent years, was used for the first time to determine the PID parameters. It has been shown that the CGO algorithm can be used to determine the coefficients of the PID controller.

1. Introduction

Two-limbed robotic arms are a widely used manipulator in industry for many different purposes. It can perform complex work with rotational movement at two separate connection points. With this robotic system, which can work quickly, precisely, and without fatigue, many applications in the industry, such as material handling, mass production in automotive, and welding processes, can be realized. The importance of these robot arms, which save manpower, is increasing day by day, with the ability to do sensitive work at the same time.

The use of robot arms in industry started with a crane-like robot made by P. Taylor in 1937 [1]. Since 1937, there have been many developments in robotic systems. Studies on the controllability of robot arms, which is one of these developments, have increased. Many different controllers have been developed for the control of robot arms, and determining the coefficients that make up this controller has created a new problem.

PID control is used to minimize the error between the reference and the measured value determined in robotic systems [2]. In a study conducted in 2001, it was concluded that the rate of using a PID controller for a system that requires control is more than 90%. It was observed that this rate decreased by 50% in 2017, but it is still the most preferred controller [3]. PID controllers, which date back to the 1890s, are generally used to control robot arms [4].

There are proportional (k_p), derivative (k_d), and integral (k_i) constants in the PID controller. With the proportional gain coefficient (k_p), the error in the process output can be directly controlled; with the derivative coefficient (k_d), it can control the rate of change of the error, and the integral (k_i) controls the sum of the error over time. It is aimed at obtaining the closest result to the reference determined by these

*Corresponding author: *smuftu@pau.edu.tr*

Received: 02.10.2023, Accepted: 08.03.2024

three coefficients that make up the controller. If these coefficients are chosen smaller or larger than the required value, the targeted result cannot be achieved, and the control performance can be increased if the correct parameter is selected. Therefore, determining the coefficients of the PID controller is of great importance for the robot to perform the desired task.

There are many methods used to determine PID control parameters. In their most general form, classical techniques can be classified as analytical, parametric, frequency response, adaptive tuning, and metaheuristic algorithms such as the Ziegler-Nichols and Cohen Coon methods. Metaheuristic optimization algorithms can be divided into swarm intelligence, Evolutionary Immunity, neural, probability, and physics-related algorithms [4]. In this study, swarm intelligence algorithms are used to solve the problem of determining the parameters of the PID controller.

In this study, a PID controller was designed for the control of the two-limb robot arm, and it was aimed at obtaining the most appropriate controller parameters with the PSO, ABC, and CGO algorithms, which are metaheuristic optimization algorithms. PSO and ABC have been used in studies on existing robotic system control, but it is thought that CGO will contribute to existing studies with this study. It is seen as a great advantage that the CGO algorithm has certain parameters, such as other optimization algorithms based on swarm intelligence, and it does not contain any parameters other than these. The parameters found were tested on the realized robot arm, and the results were compared both as a simulation and on the real robot arm.

2. Material and Method

The two-legged robot arm is the simplest robotic arm in robotic systems. Two separate drive elements are required to perform. In this study, a 1524006SR DC motor from Faulhaber company, an L298 motor driver relay to drive motors, and an Arduino Mega as a microcontroller were used. The parts of the designed robot arm were obtained from the 3D printer. The designed two-limbed robot arm can be seen in Figure 1 a-), and the produced two-limbed robot arm can be seen in Figure 1 b-).

The 1524006R model of Faulhaber has a DC motor with a 76:1 rotation ratio and a 2-channel field effect encoder. This brand has clearly presented all the necessary parameters to its users to obtain the transfer function of the DC motor required for control.

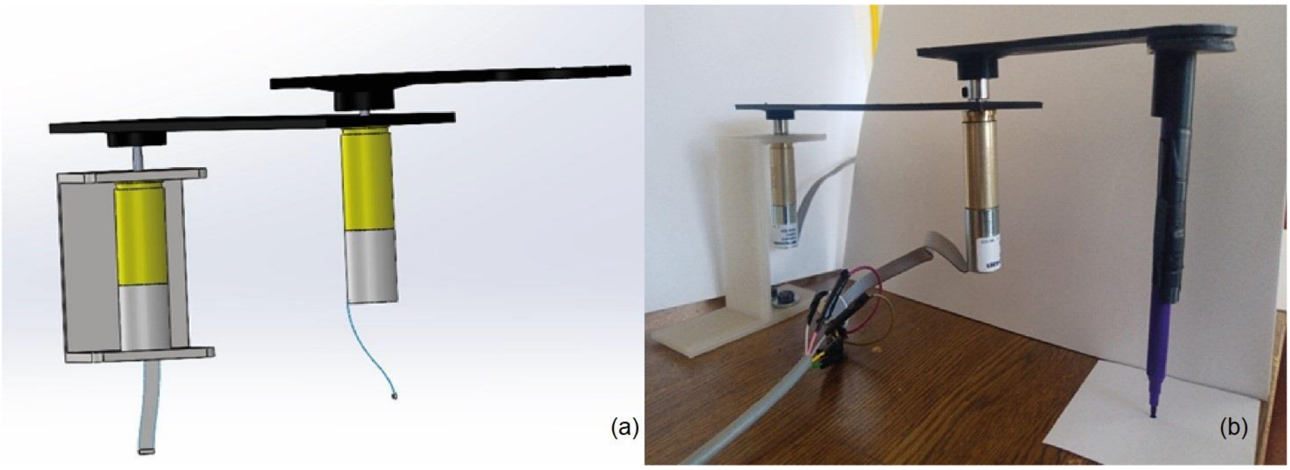

Figure 1. Two-limbed robotic arm (a) Designed two-limbed robot arm (b) two-limbed robot arm produced.

The distance between the shaft of the motor fixed to the floor of the robot arm and the motor connected to the gripper end is L1, 17.75 cm, and the other has a length of 8.25 cm. With an arm connected to the end of a single motor, only a circle with a radius of arm length can be drawn, while the gripper end of the robot arm can be guided in any way between the two-limb system and an outer circle with a radius of L1+L2 and an inner circle with a radius of L1-L2. The working area of the robot arm is shown in Figure 2.

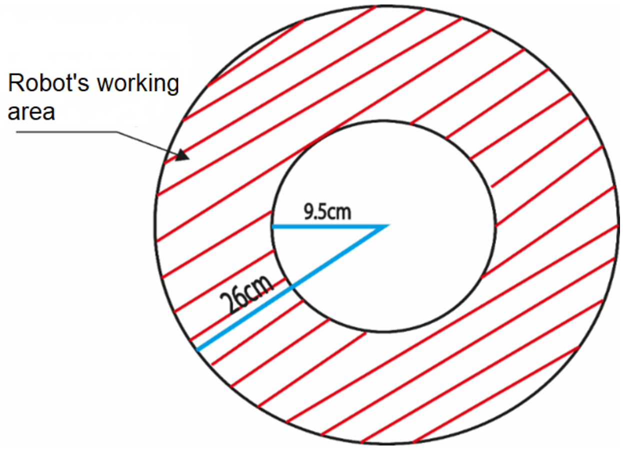

Figure 2. The working area of a two-limbed robotic arm.

2.1. Controller Design of a Two-Limb Robot Arm

In robotic systems, there are two mathematical approaches to moving the end point of the robot arm from the starting point to a determined point. These are expressed as forward and reverse kinematics. The forward kinematics approach allows us to calculate the position and direction change with the values given to the variables, while the inverse kinematics approach allows the values of the variables to be obtained for the robot arm to reach a certain position. Inverse kinematic analysis is of great importance for the control of the robot. The schematic representation of the robot arm realized in the study is given in Figure 3.

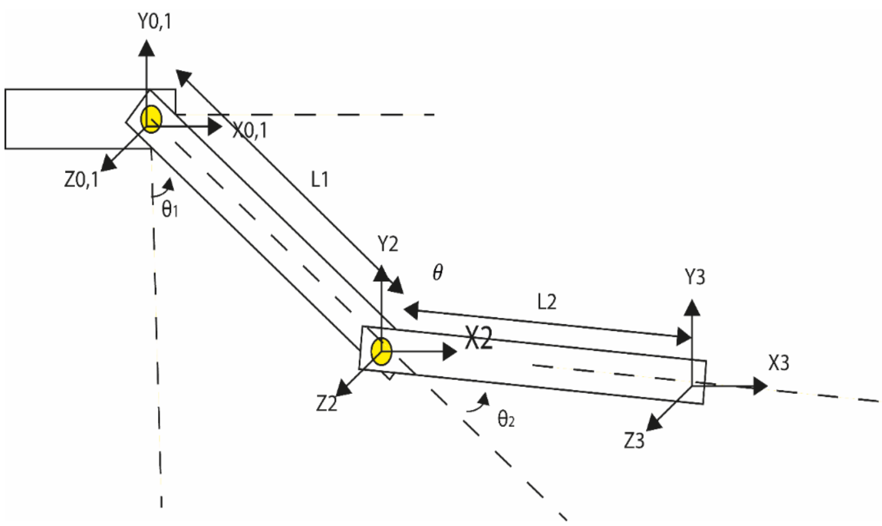

Figure 3. Schematic representation of the two-limb robot arm.

The geometric kinematic analysis of the two-legged robot arm can be easily calculated. The projection along the x-axis in equation (1) is indicated along the y-axis in equation (2). This kinematic analysis gives the position of the robot's endpoint in the x and y axes versus the two-limb robot's variables θ_1 and θ_2.

$$x = l_1 cos\theta_1 + l_2\, cos(\theta_1 + \theta_2) \tag{1}$$

$$y = l_1 sin\theta_1 + l_2\, sin(\theta_1 + \theta_2) \tag{2}$$

Kinematically, equations (1) and (2) are used in the gripper end position problem of the robot at any moment, while inverse kinematics is the most used mathematical expression in robotic systems, as it creates the result of variable values against the entered position value. Inverse kinematic values can be found as a result of many operations performed in the robotic system. In equations (3) and (4), the inverse kinematics expression of the variables θ_1 and θ_2., which are the variables of the two-limbed robot, are stated, respectively [5].

$$\theta_1 = atan\left(\frac{y}{x}\right) + tan^{-1}\left(\frac{l_2 sin\theta_2}{l_1 + l_2 cos\theta_2}\right) \qquad (3)$$

$$\theta_2 = +\cos^{-1}\left(\frac{x^2+y^2-{l_1}^2-{l_2}^2}{2l_1 l_2}\right) \qquad (4)$$

There are many systems used as drive elements in robot arms. One of them, DC motors, are electromechanical devices that convert electrical energy into mechanical energy. In control science, the transfer function is used to model the behavior of a DC motor. It connects the electrical voltage information entering the system with the output mechanical position information. The Laplace transform is used to model the nonlinear behavior around the equilibrium point of the DC motor. The transfer function for the 1512406sr type DC motor of the Faulhaber company used in this study was found with the values specified in the catalog section. In Table 1, the required features of the Faulhaber 1524006sr DC motor for TF are given.

Table 1. Faulhaber 1524006sr DC motor specifications.

Feature Name	Value
Rotor Inertia	0.66 gcm^2
Viscous Damping	0
Electrical Constant	0.172 $\frac{A}{mNm}$
Torque Constant	5.8 $\frac{mNm}{A}$
Robor Inductance	70 μH
Terminal Resistance	1.1 Ω

We can consider the equivalent circuit of the DC motor as in equation 5 [6].

$$\frac{\theta(s)}{V(s)} = \frac{K_t}{(R_a+s+La)\,(J*S*B)s+K_t*K_b*s} \qquad (5)$$

The values in Table 1 are replaced by the Rotor inertia J, viscous damping B, electrical constant K_b, torque constant K_t, Rotor inductance L rotor resistance R in the transfer function found in Equation (16), and the transfer function for the Faulhaber DC motor is found. The transfer function for the DC motor is given in Equation (6). If the unknowns in equation 5 are filled by means of table (1), equation (6) is obtained.

$$\frac{\theta(s)}{V(s)} = \frac{5{,}8\times10^{-3}}{(5.1+70\times10^{-6}s)(0.66\times10^{-7}s+1)s+5{,}8\times10^{-3}s} \qquad (6)$$

If PID control is used correctly in robot arms, it enables robots to save energy with precision, stability, and less movement. Since the study has two degrees of freedom, two PID control blocks were used. The reference value for the PID control block is obtained from inverse kinematic analysis. Since each PID control has three variables, there are six parameter values in total. Angle obtained after the PID control block value is sent to the motors, respectively. The operation block of the angle values sent to the motors against the entered reference x and y axes is given in Figure 4.

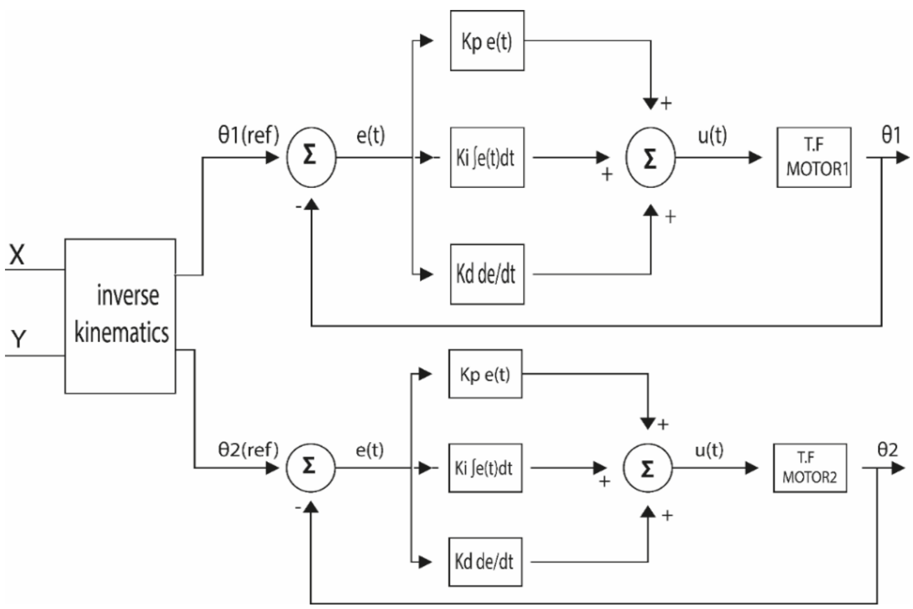

Figure 4. Blok diagram of the robot arm.

For square movement, the robot arm started from the point where theta1 and theta2 limb angles were 0 degrees and followed the points determined on a coordinate axis. The determined points form a square shape. The gripper end of the two-limbed robot arm first starts from point (3.19) and goes to point (8.19). Afterwards, it goes to (8.14), (3.14), and (3.19) points, respectively, and finishes the movement. The end point of the robot arm follows a square shape with a corner length of 5cm. The reference dimension placed on the coordinate axis of the square shape is indicated in Figure 5.

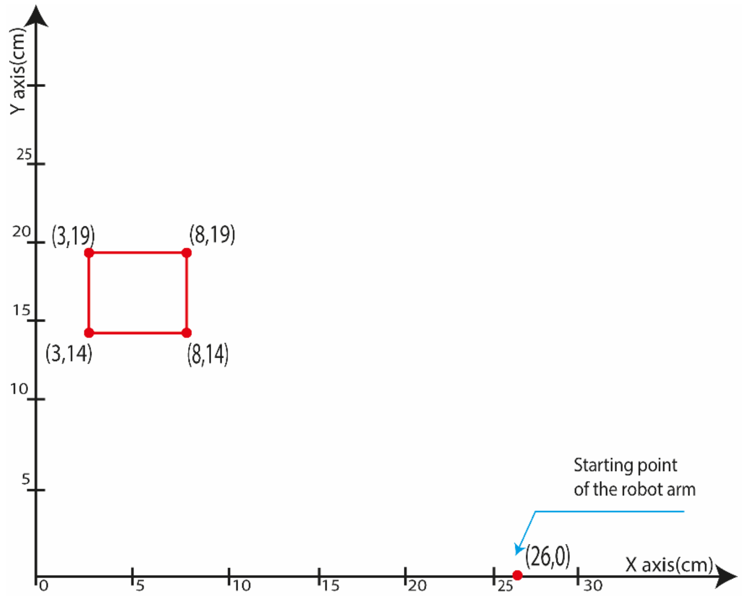

Figure 5. Square motion plan of a two-pronged robotic arm.

In order to achieve the desired movement in robot arms, it is necessary to utilize inverse kinematics and any controller. The trajectory created with the Simulink Support Package for Arduino Hardware was converted into an angle value with inverse kinematics and sent to the PID expression for control. The PID parameters obtained as a result of the optimization process were set as controller parameters in the real system. The block diagram of the realized system is shown in Figure 6.

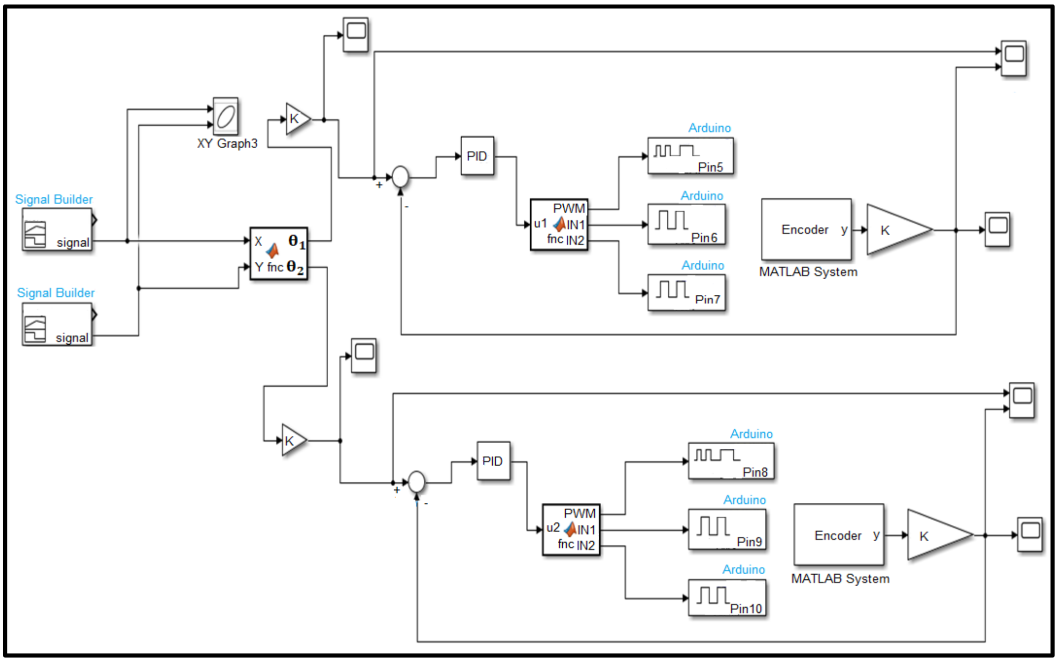

Figure 6. Block diagram of the two-limbed robot arm in the Simulink Support Package for Arduino Hardware.

In the system implemented with Arduino support from Simulink, 8 seconds are determined for square movement. During the sampling period, angular position data from the encoders provided feedback to the PID control block. The 8-second temporal change reference for θ_1 and θ_2 separately is given in Figure 7.

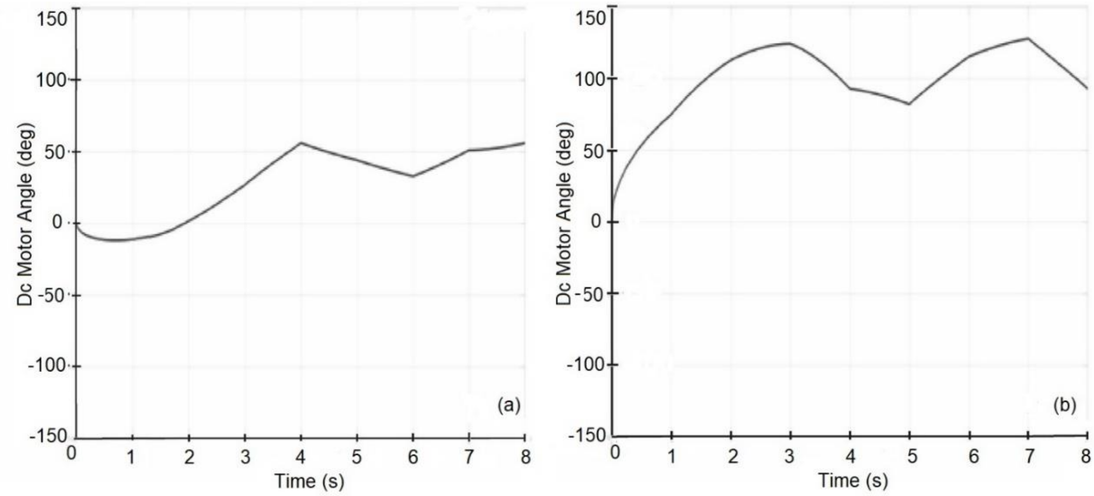

Figure 7. Angular positions that a two-limbed robot arm must make for square motion tracking.

2.2. Controller Parameter Optimization of a Two-Limbed Robot Arm

Optimization is the process of finding the best value of a given objective function with constraints. This process is encountered in many fields, from engineering systems to economics, from health to business. Optimization problems can be categorized as continuous or discrete, dynamic or static, constrained or unconstrained. These algorithms are a sub-branch of artificial intelligence, and as the popularity of artificial intelligence increases, so does its use for different problems [8].

Meta-heuristic algorithms can often be classified according to their inspiration. Another classification is the type of initial solution, i.e., multiple or single solution-based. Multiple solution-based algorithms are usually called population-based, while single solution-based algorithms are called trajectory-based [9]. Classifying metaheuristic optimization algorithms according to the source of inspiration is usually classified as human, evolutionary, swarm logic, or science-based. PSO, ABC, CGO meta-heuristic algorithms were used in this study.

2.2.1. Particle Swarm Optimization Algorithm

PSO is a meta-heuristic optimization algorithm that mimics particle swarm behavior to solve optimization problems. It was developed in 1995 by Dr. Kennedy and Dr. Eberhart. PSO is applied in many different fields thanks to its easy implementation, fast solution, and high success rate. PSO is based on the principle of particles searching for the optimal solution to a given problem [10].

The velocity and position of each particle are expressed as a vector. There is also a fitness function that measures the fitness of each particle, which varies according to the problem. Each particle also keeps track of its past best position (pbest) and the best position in the swarm (gbest). The particles move towards the optimal solution by continuously updating pbest and gbest. Finally, when the optimal solution is obtained, the solution is achieved [11]. The steps that the particle swarm optimization algorithm takes to solve a problem are shown in Figure 8 as a flowchart.

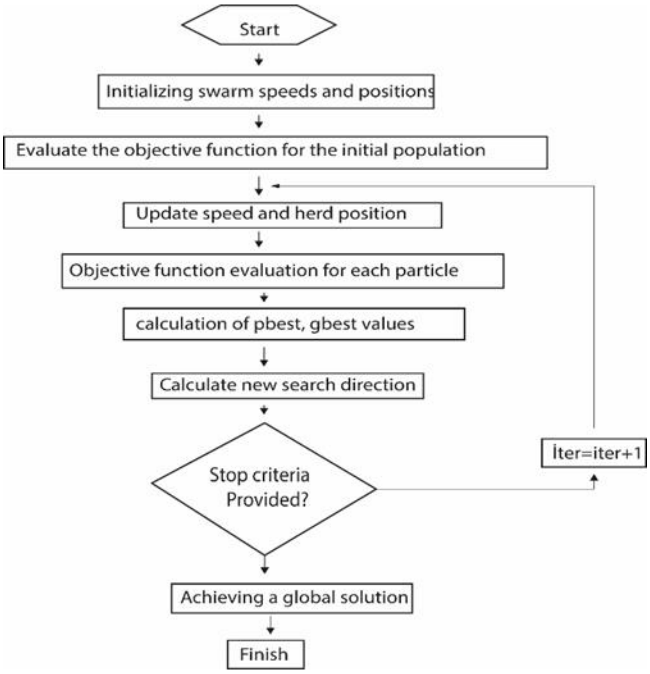

Figure 8. Flowchart of the particle swarm optimization algorithm [12].

The ability of each particle to solve the problem and find the best solution means individual performance for the PSO algorithm. The PSO algorithm tries to find the best solution by updating the position and velocity of each particle. In this way, the individual performance of each particle plays an important role in the process of solving the problem. The PSO algorithm encourages cooperation and information sharing between particles to optimize individual performance and improve each particle's ability to solve the problem. In this way, the algorithm is often used effectively in complex optimization problems.

2.2.2. Artificial Bee Colony Optimization Algorithm

ABC is a metaheuristic optimization algorithm developed by Karaboga in 2005. It is inspired by the behavior of bees as they forage for food sources and bring it back to their nests. The behavior of bees is modeled mathematically. The parts of ABC can be listed as relocation of bees, discovery of food sources, evaluation of food sources, and information sharing among bees [13].

The ABC algorithm is simulated in a search space where bees form a colony to represent potential solutions. The bees evaluate randomly selected solution candidates and use this information to generate new solution candidates by sharing the best solutions with other bees. This process is iteratively repeated to optimize a given objective function [14]. More details about the ABC algorithm are given in Figure 9 as a flowchart.

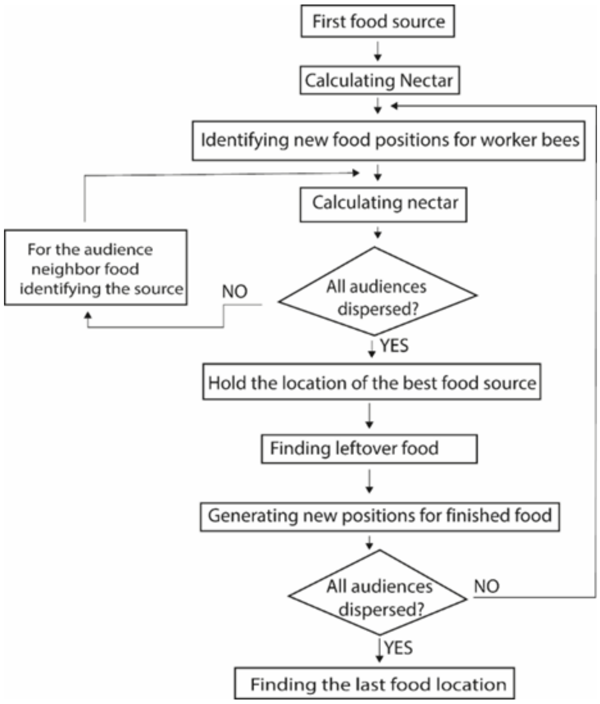

Figure 9. Flowchart of an artificial bee colony optimization algorithm flowchart [14].

Finding the optimal solution by mimicking the behavior of the bees to search for and gather the food source constitutes the individual performance of the ABC algorithm. As the bees communicate with each other, they work together to find the best solution. With the improvement in the performance of each bee, the problem-solving ability improves. As a result, thanks to the organization and cooperation exhibited by the bees, the ABC algorithm gives successful results in real-world optimization problems.

2.2.3. Chaos Game Optimization

The chaos game optimization algorithm is a population-based optimization algorithm developed by Talatahari and Azizi. This algorithm incorporates certain chaos theory principles where fractals are generated using chaos game methodology. This game, which aims to generate fractal patterns based on randomly generated starting points, is designed based on the hypothesis of generating the Sierpinski triangle geometric structure [15].

CGO is known for positive aspects such as high computation time efficiency and easy implementation to effectively solve constrained optimization problems. The biggest difference from other optimization algorithms is that it is parameter-free. In other words, it does not need any additional parameters other than parameters such as population size, and maximum iteration. Thanks to this feature, it overcomes challenging problems. The algorithm starts optimization by initially generating random search candidates, and this initialization process is performed depending on the population size, the number of decision variables, and the bounds of the solution space [15]. More details about the CGO algorithm are given in Figure 10 as a flowchart.

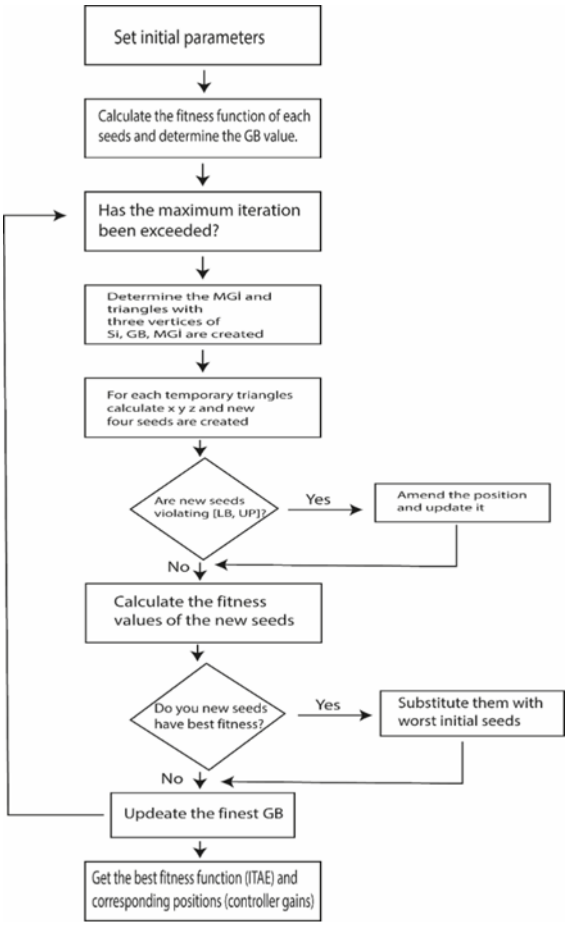

Figure 10. Flowchart of Chaos game optimization algorithm flowchart [16].

In the CGO algorithm, the success of each solution constitutes the individual performance of the algorithm. The CGO algorithm tries to find the best solution by randomly moving each solution candidate on a fractal structure. While solving the optimization problem, the CGO algorithm compares the similarities of the solution candidates and keeps the best solution.

3. Results and Discussion

There are six controller parameters in total for the realized two-legged robot arm. Particle swarm, artificial bee colony, and chaos game optimization algorithms were used to find the most suitable values of these parameters. The simulation results obtained from each metaheuristic optimization algorithm are compared. As a result of the simulation process, the controller parameters found for trajectory tracking in the real robot arm are specified.

In all three simulations, the maximum number of function evaluations was carried out at 40 and the number of populations at 50 in order to perform the most appropriate tracking of the square trajectory for the robot arm. The social and cognitive constants of the PSO algorithm were determined to be 2, and the inertia weight was determined to be [0.9, 0.4] [7]. The limit value for the ABC is determined to be 100. These coefficients were used at the same values for each step throughout the study. If we put the PID parameters in order $k_{p_1}, k_{i_1}, k_{d_1}, k_{p_2}, k_{i_2}, k_{d_2}$ limit [0 0 0 0 0 0], and upper limit [20,000 15,000 18,000 20,000 15,000 18,000]. The simulation results obtained as a result of the optimization process for frame motion tracking are shown in Figures 11 for the chaos game, 12 for the particle swarm, and 13 for the artificial bee colony.

It is clearly seen in Figure 8, Figure 9, and Figure 10 that CGO and ABC algorithms give better results than PSO algorithms. The ABC and CGO algorithms gave similar results. Figure 14 shows that the CGO algorithm gives better results than the ABC algorithm. In Figure 14, (A) is a close view of CGO, and (B) is a close view of the ABC algorithm at the point (8,14) forming the square. It is noted that CGO is closer to the reference value of red dots than ABC. In the simulation environment, all three algorithms gave successful results, but CGO gave better results with differences. For this problem, each algorithm is successful but the best ranking is CGO, ABC, and PSO. The reason for this is the parameters in the PSO and ABC algorithms, and it is thought that the CGO algorithm is more successful in generating new values.

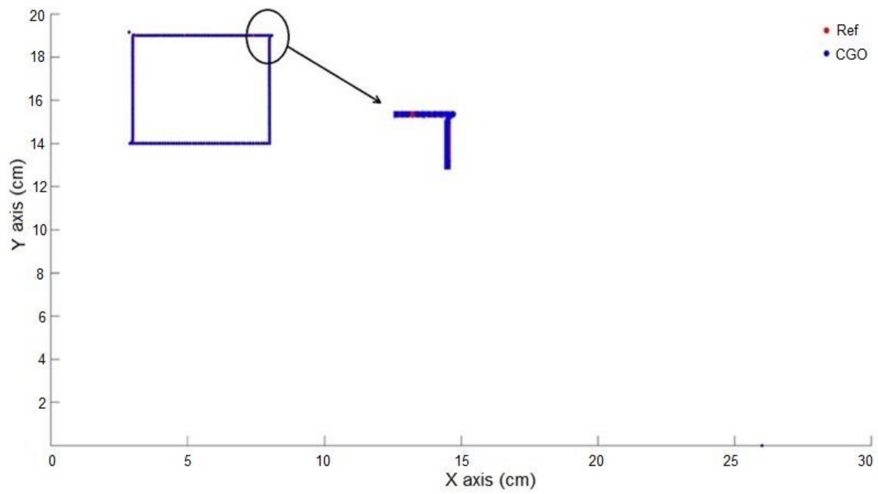

Figure 11. CGO reference tracking for square motion.

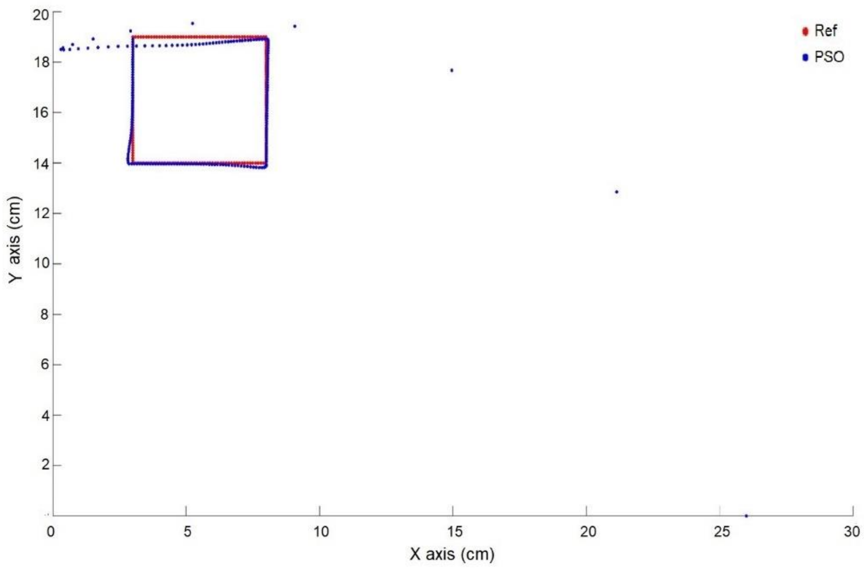

Figure 12. PSO reference tracking for square motion

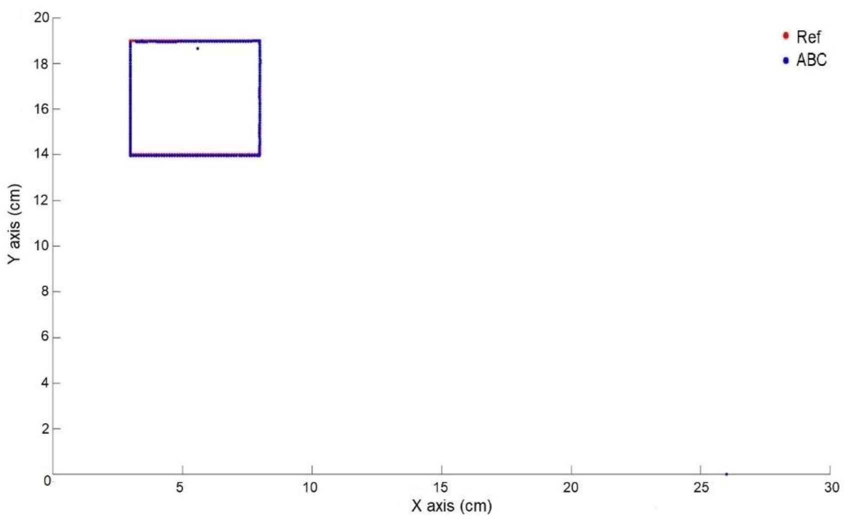

Figure 13. ABC reference tracking for square motion

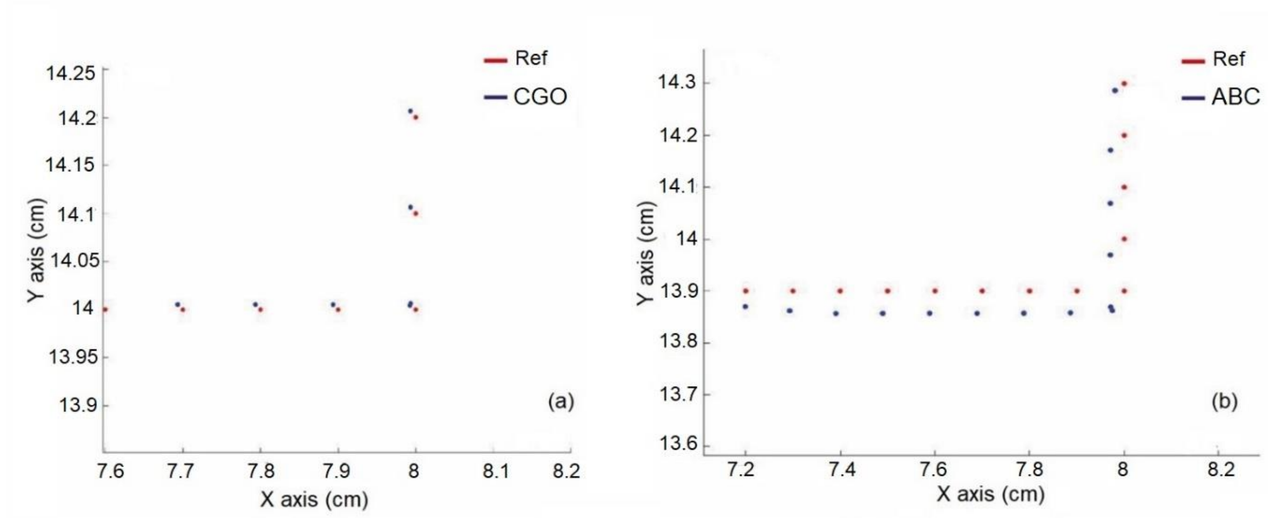

Figure 14. Comparison of ABC and CGO reference

Mean Square Error (MSE) and Root Mean Squared Error (RMSE) were used to compare the error parts of the optimization process of the angles θ_1 and θ_2 that make up the two-limbed robot arm. The MSE and RMSE values obtained by CGO, PSO, and ABC are given in Table 2.

Table 2. MSE and RMSE error metrics of CGO, PSO, and ABC algorithms for reference tracking.

	θ_1 angle		θ_2 angle	
	MSE	RMSE	MSE	RMSE
CGO	32.33	5.68	154.38	12.42
PSO	58.98	7.67	238.79	15.45
YAKA	32.74	5.72	155.36	12.46

As a result of the optimization process, six optimization parameters k_{p_1} , k_{i_1} ,k_{d_1} ,k_{p_2} ,k_{i_2} ,k_{d_2} were obtained for each optimization algorithm. The PID parameters found are given in Table 3.

Table 3. The PID parameters found for square motion.

	Controller Parameter	CGO	PSO	ABC
PID θ_1	k_{p_1}	19853	15.016	17667
	k_{i_1}	799	17675	0
	k_{d_1}	1082	3446	0
PID θ_2	k_{p_2}	19775	19374	20000
	k_{i_2}	0	9252	1835
	k_{d_2}	1118	3776	1157

Each metaheuristic optimization algorithm, CGO, PSO, and ABC, which determines the PID parameters obtained as a result of the simulation performed on the robot arm controlled in real time thanks to the Simulink support package for Arduino, has been tested separately on the real system. Table 2 shows the differences in the controller parameters obtained by the CGO, PSO, and ABC algorithms. These differences come from the structure of the optimization algorithms.

The angle value that each motor should make for 8 seconds in order for the two-limbed robot arm to perform the reference motion is given in Figure 7. The results obtained by realizing the PID controller parameters specified in Table 2 on the real system are given in Figure 15 (a) for θ_1 and Figure 15 (b) for θ_2 for the CGO algorithm, Figure 16 (a) for θ_1 and Figure 16 (b) for θ_2 for the PSO algorithm, Figure 17 (a) for θ_1, and Figure 17 (b) for $\theta_2$for the ABC algorithm.

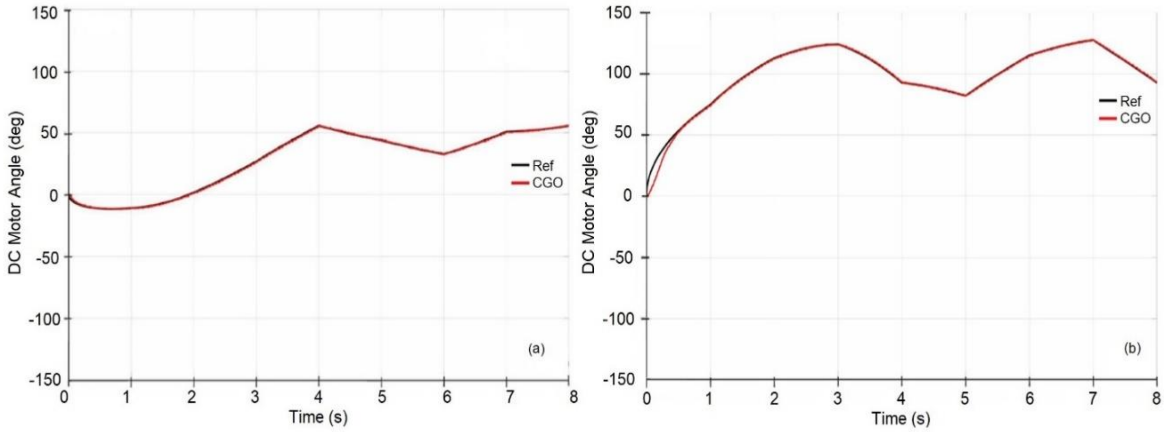

Figure 15. Reference tracking of PID parameters obtained by CGO for square orbit at angle.

(a) θ_1 (b) θ_2.

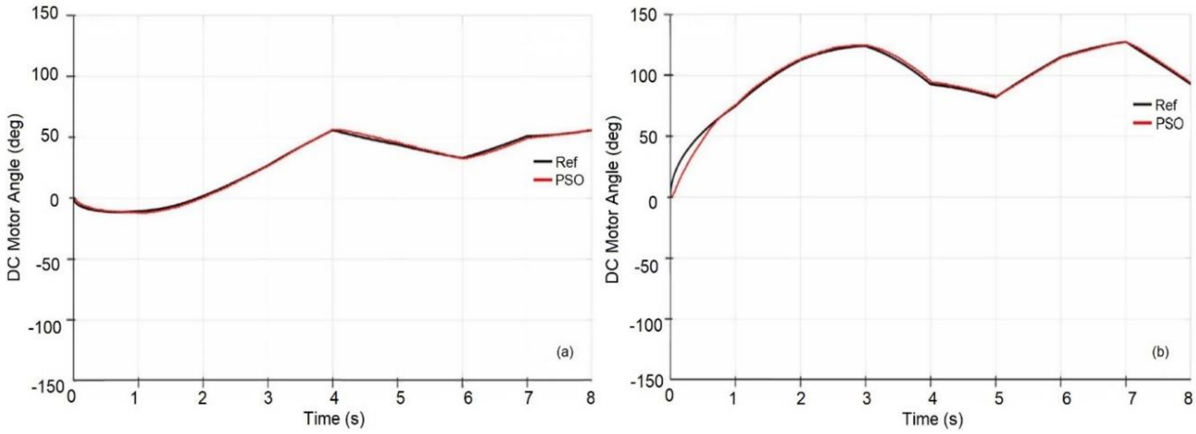

Figure 16. Reference tracking of PID parameters obtained by PSO for square orbit at angle (a) θ_1 (b) θ_2.

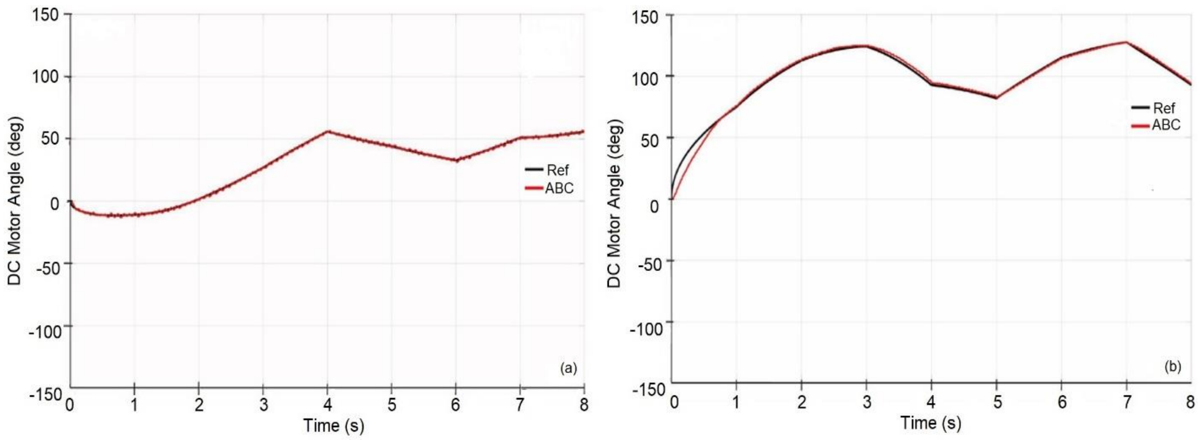

Figure 17. Reference tracking of PID parameters obtained by ABC for square orbit at angle (a) θ_1 (b) θ_2.

When the experimental data obtained from real-time measurements of the DC motors, which are the actuators of the two-limbed robot arm, are analyzed in Figures 15, 16, 17; PSO and ABC algorithms gave very close results to the reference as mentioned in the literature. For the square motion, which is one of the most difficult motions for a two-limbed robot arm, all three algorithms gave very successful results in real-time control. However, the recently popular and promising CGO algorithm gave slightly better results than the PSO and ABC algorithms. This is the same as the simulation result. This is due to the fact that the CGO algorithm finds the parameters of the PID controller closer to the correct values.

4. Conclusion and Suggestions

Many studies in the literature on robot control have been done with servo motors or stepper motors. The DC motor used in this study gave a very successful result and demonstrated that a DC motor with certain parameters can be used for robot control. The transfer function of the DC motor was obtained from the catalog values, and the controller feedback was provided by the encoder. There are many algorithms to determine the parameters of the PID controller used to control the two-limbed robot arm. In this study, the CGO algorithm is used for the first time to determine the PID parameters. The CGO, PSO, and ABC algorithms are compared as a result of the simulation of the transfer function of the DC motor. The parameters of the PID controller obtained as a result of the simulation were tested on the two-limbed robot arm.

As a result of the simulation process, the CGO algorithm gave a result closer to the reference than the PSO and ABC algorithms. In order for the two-limbed DC motors to perform square motion, the change in the angle value of the DC motors for 8 s was calculated, and the control was performed for each motor. When the motors run for 8 s, the end part of the robot draws a square motion. Thanks to the metaheuristic optimization algorithms used, the CGO algorithm gave more successful results than the PSO and ABC algorithms in the PID controller parameters for the two-limbed robot arm. The success of metaheuristic optimization algorithms is not the same for every problem. The success of an optimization algorithm in one problem does not mean that it will be successful in another. This study has shown in the literature that the CGO metaheuristic optimization algorithm can be used to determine the coefficients of a PID controller.

The controller part is of great importance in the studies to be carried out on robotic systems. Among the methods to be used for the problem of determining controller parameters, methods with a small number of parameters should be chosen. Since the angular loss of any axis affects the holder end, the material to be used with the smallest error rate should be selected. Motor selection should be made according to the torque value required for the system. If the motor to be used as a drive element is a DC motor, the step range of the encoder should be high, the value of the motor speed should be low, and the cycle ratio of the reducer should be high.

Acknowledgment

This study was conducted at Necmettin Erbakan University, Graduate School of Natural and Applied Sciences, Mechatronics Master's thesis written within the scope of the master's program at the Department of Engineering prepared.

Contributions of the authors

All authors contributed equally to the study.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

References

[1] W. S. Barbosa, M. M. Gioia, V. G. Natividade, R. F. Wanderley, M. R. Chaves, F. C. Gouvea and F. M. Gonçalves, "Industry 4.0: examples of the use of the robotic arm for digital manufacturing processes," *International Journal on Interactive Design and Manufacturing (IJIDeM),* vol. 14, pp.1569-1575, 2020.

[2] S. K. Valluru and M. Singh, "Optimization strategy of bio-inspired metaheuristic algorithms tuned PID controller for PMBDC actuated robotic manipulator," *Procedia Computer Science*, vol.171, pp. 2040-2049, 2020.

[3] K. J. Åström and T. Hägglund. "The future of PID control," *Control engineering practice*, vol. 9, no. 11, pp.1163-1175, 2001.

[4] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems," *Heliyon*, vol. 8, no. 5, p.e09399, 2022.

[5] S. Kucuk and Z. Bıngul, *Robot kinematics: Forward and inverse kinematics, London, UK*: INTECH Open Access Publisher, 2006, pp. 117-148.

[6] S. Maheriya and P. Parikh, "A review: Modelling of Brushed DC motor and Various type of control methods," *Journal for Research*, vol. 1, 2016.

[7] X. Hu, "Particle swarm optimization tutorial, titan.csit.rmit.edu, 2006. [Online]". Available: https://titan.csit.rmit.edu.au/ e46507/publications/pso-tutorial-seal06.pdf. [Accessed: March 22, 2021].

[8] J. Tang, G. Liu and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems," *Applications and trends. IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp.1627-1643, 2021.

[9] E. Boğar, "Optimizasyon kuramında yeni bir metasezgisel yaklaşım: Ergen kimlik arama algoritması (AISA) ve mühendislik uygulamaları," Doktora tezi, Pamukkale Üniversitesi Fen bilimleri Enstitüsü, Denizli, 2021.

[10] R. Eberhart and J. Kennedy, "Particle swarm optimization," *in Proceedings of the IEEE international conference on neural networks* vol. 4, pp. 1942-1948, November 1995.

[11] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," *In 2007 IEEE swarm intelligence symposium*, pp. 120-127, April 2007.

[12] M. Azab, "Global maximum power point tracking for partially shaded PV arrays using particle swarm optimization," *International Journal of Renewable Energy Technology*, vol. 1, no.2, pp. 211-235, 2009.

[13] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization*, vol. 39, pp. 459-471, 2007.

[14] H. O. Erkol, "Ters sarkaç sisteminin yapay arı kolonisi algoritması ile optimizasyonu," *Politeknik Dergisi*, vol. 20, no. 4, pp.863-868, 2017.

[15] S. Hassan, B. Abdelmajid, Z. Mourad, S. Aicha and B. Abdenaceur, "An advanced MPPT based on artificial bee colony algorithm for MPPT photovoltaic system under partial shading condition. International," *Journal of Power Electronics and Drive Systems*, vol. 8, no. 2, p. 647, 2017.

[16] S. Talatahari and M . Azizi, "Chaos game optimization: a novel metaheuristic algorithm," *Artificial Intelligence Review*, vol. 54, no. 2 pp. 917-1004, 2021.

[17] M. Barakat, "Novel chaos game optimization tuned-fractional-order PID fractional-order PI controller for load–frequency control of interconnected power systems," *Protection and Control of Modern Power Systems*, vol. 7, no. 1, 2022.