IJRCS
International Journal of Robotics and Control Systems
Vol. 2, No. 2, 2022, pp. 435-447
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs
ASCEE

Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor

Eka Suci Rahayu [a], Alfian Ma'arif [a,1,*], Abdullah Cakan [b]

[a] Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[b] Konya Technical University, Konya, Turkey
[1] alfianmaarif@ee.uad.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received October 07, 2021
Revised May 11, 2022
Accepted June 29, 2022

Keywords
PID Control;
PSO;
Angular Speed;
Arduino;
Tuning

ABSTRACT

The use of DC motors is now common because of its advantages and has become an important necessity in helping human activities. Generally, motor control is designed with PID control. The main problem that is often discussed in PID is parameter tuning, namely determining the value of the Kp, Ki, and Kd parameters in order to obtain optimal system performance. In this study, one method for tuning PID parameters on a DC motor will be used, namely the Particle Swarm Optimization (PSO) method. Parameter optimization using the PSO method has stable results compared to other methods. The results of tuning the PID controller parameters using the PSO method on the MATLAB Simulink obtained optimal results where the value of Kp = 8.9099, K = 2.1469, and Kd = 0.31952 with the value of rise time of 0.0740, settling time of 0.1361 and overshoot of 0. Then the results of hardware testing by entering the PID value in the Arduino IDE software produce a stable motor speed response where Kp = 1.4551, Ki= 1.3079, and Kd = 0.80271 with a rise time value of 4.3296, settling time of 7.3333 and overshoot of 1.

1. Introduction

The use of DC motors is now common and has become an important necessity in helping human activities, for example, industrial applications [1], moving an item such as running a conveyor machine [2], sucking water from underground to the surface using a water pump [3], cooling a room by turning a fan [4], robotics [5][6], and electric vehicle [7]. Motor control is needed to get the movement of the rotational speed [8][9] or motor position [10] to match a predetermined value. Generally, motor control is designed with PID control [11].

Proportional Integral Derivative (PID) controllers are increasingly being used in controlling DC motors [12], Quadrotor [13], Magnetic Levitation System [14], Quadcopter [15], Inverted Pendulum [16], and Converter [17][18]. Some of the advantages of PID are simple structure, good stability, strong resistance, and ease of implementation in software or hardware [19][20]. PID Control has been implemented in many systems such as temperature control [21], Aircraft [22], and Transport robots [23].

The main problem that is often discussed in PID is parameter tuning [24][25], namely determining the parameter values of the Proportional Constant (K_p), Integral Constant (K_i), and Derivative Constant (K_d) in order to obtain optimal system performance [26][27]. One technique that is often

https://doi.org/10.31763/ijrcs.v2i2.476

used is trial-error conventional control [28], but for this method, it is difficult to adjust the parameters, so the parameter search takes a long time, the control accuracy is not good, and the parameters used are not optimal.

In recent years, researchers have used many intelligent methods for tuning PID parameters, such as the Flower pollination algorithm, Teaching learning based optimization [29], Artificial Bee Colony Algorithm [30][31], Grey Wolf Optimization [32], Firefly Algorithm [33], Differential Evolution [34], Genetic Algorithm [35], Sine Cosine Algorithm [36][37], Water Wave Optimization [38]. Researchers began to study the intelligent behavior of animals to be applied in solving optimization problems such as Whale Optimizer Algorithm [39], Fish Migration Optimization Algorithm [40], Grey Wolf Optimizer [41], Artificial Bee Colony Algorithm [42], Bat Algorithm [43], Harris Hawk Optimization [44][45]. Several optimization methods based on conventional methods and intelligent methods have been widely used to optimize PID parameters on DC motors [46][47][48][49][50]. In this study, one of the smart methods for tuning PID parameters on a DC motor will be used, namely the Particle Swarm Optimization (PSO) method [51]. Several research references said that parameter optimization using the PSO method has stable results compared to other methods [52][53][54].

Some control methods proposed in previous research for DC Motor [55][56][57] were applied only to simulation systems, so further validation is required. This paper applied the control method not only to simulation but also to hardware implementation. Thus, the research contributes to simulation-based PID tuning and implementing the control method in the hardware system.

2. Method

2.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, which was inspired by the social behavior of a flock of birds or fish [58]. In Particle Swarm Optimization, the swarm is assumed to have a certain size, with each particle starting position at a random location in multidimensional space. Each particle is assumed to have two characteristics, namely position, and velocity. Each particle moves in a certain space or space and remembers the best position found with respect to the value of the objective function. Each particle conveys information or its best position to the other particles and adjusts the position and speed of each based on the information received about the good position [59]. Particle Swarm Optimization (PSO) is one of the evolutionary computing techniques in which the population in PSO is based on a search algorithm and begins with a random population called a particle.

2.2. Proportional Integral Derivative (PID) Controller

The main problem in this study is tuning the PID parameter. Therefore, the PSO algorithm method is used to find the best parameter values (viewed from the rise time, settling time, overshoot, and steady-state error) [52]. The PSO algorithm method is used to avoid trial and error because it will take a long time. In designing the PID control system, what needs to be done is to adjust the parameters P, I, or D so that the system output signal response to a certain input is as desired. The PID controller equation in Laplace form is [60]

$$\frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$
$$= K_p + \frac{K_i}{s} + K_d s$$

Where K_p is proportional gain value, K_i is the integral gain value, K_d is the derivative gain value, E is the error between the reference value and feedback value, U is the control signal value. The discrete form of the PID Controller is

$$u(k) = K_p e(k) + K_i \sum_{n=0}^{N} e(k) + K_d(e(k) - e(k-1))$$

Where k is the discrete step at time t.

The independent effect of increasing parameter value in PID control is shown in Table 1. For example, while the K_i and K_d is fixed, increasing K_p alone can decrease rise time, increase overshoot, slightly increase settling time, decrease the steady-state error, and decrease stability margins [61].

Table 1. Effect of Independent P, I, and D tuning

	Rise Time	**Overshoot**	**Settling Time**	**Steady State Error**	**Stability**
Increasing K_p	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_i	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_d	Small Decrease	Decrease	Decrease	Minor Change	Improve

2.3. DC Motor Model

A plant in modeling is a system that may be composed of various components/elements that are interconnected in carrying out an action. Depending on the components/elements used, a system can be of mechanical, pneumatic, electrical, or electro-mechanical type. DC motor electrical modeling obtained the following equation as

$$V_R + V_l + V_{emf} = V_s$$

$$iR + L\frac{di}{dt} + K_e w = V_s$$

Using the Laplace transform, the last equation to be equation as

$$IR + LIs + K_e W = V_s$$

$$I(R + Ls) = V_s - K_e W$$

$$I = \frac{V_s - K_e W}{R + Ls}$$

DC motor mechanical modeling using Newton's law of rotation approach obtained the following equation. Rotational motion can be written by the equation

$$\Sigma\tau = J\dot{\omega}$$

$$T - fk = J\dot{\omega}$$

$$K_t i - b\omega = J\dot{\omega}$$

Where fk is the friction, ω is the angular speed, $\dot{\omega} = \frac{d\omega}{dt}$ is the angular acceleration.

Using the Laplace transform then obtained

$$W(Js + b) = K_t I$$

Combination of mechanical and electrical equations to find the angular acceleration as

$$W(Js + b) = K_t \frac{V_s - K_e W}{R + L_s}$$

$$K_t V - K_t K_e W = W(Js + b)(R + Ls)$$

$$K_t V = W(Js + b)(R + Ls) + K_t K_e W$$

$$W = \left[\frac{K_t}{(Js + b)(Ls + R) + K_e K_t}\right] V$$

In general, the torque generated by a DC motor is proportional to the armature current and the magnetic field strength. In this example, we assume that the magnetic field is constant so that the motor torque (T) is proportional to the armature current (i) only by a constant factor (K_t).

2.4. System Design

In designing this system, it is done with reference to the theories and previous research, namely designing a PID control system on a DC motor with the PSO method. This research was conducted at the system design stage, both in software design and hardware design. The hardware design includes designing a DC Motor Modeling block diagram shown in Fig. 1.

Fig. 1 is a block diagram of DC motor modeling using an Arduino Uno R3 board as a control system. The power supply provides voltage to the Arduino board, which is then channeled to the motor driver, which is to drive the DC motor, then the rotary motion of the DC motor will be read by the Encoder, which will be given to the Arduino Uno so that the results can be seen in the Arduino IDE software on the serial monitor and serial plotter menus. To see the wiring diagram of the designed system, see Fig. 2.

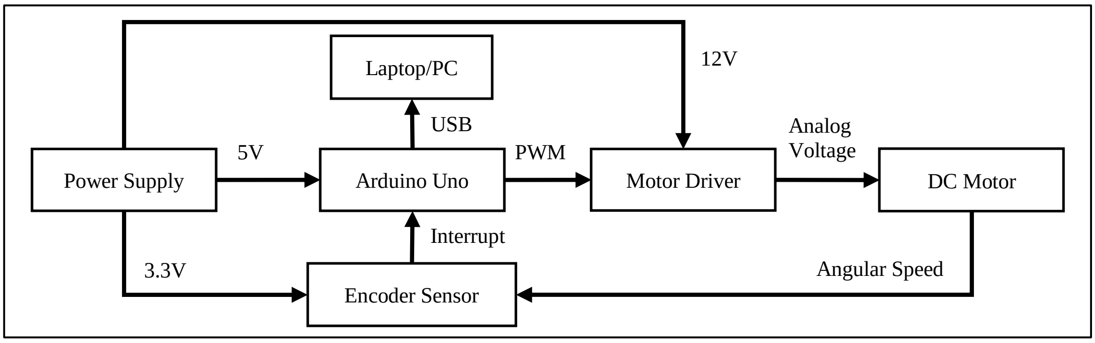
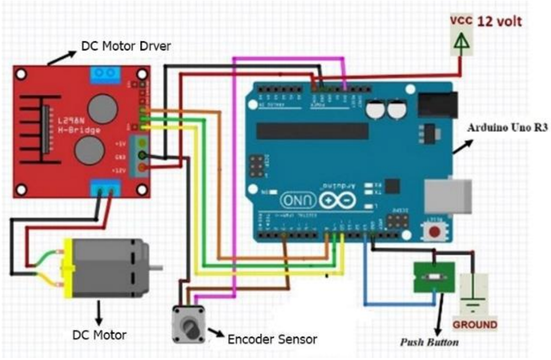


Fig. 1. Diagram Blok Hardware

In the wiring diagram Fig. 2, the Hall-effect OH42E encoder sensor outputs to Pin 2 Arduino, the Button is connected to Pin 13 Arduino and ground, input ENA motor driver from Arduino Pin10 and motor driver input IN1 and IN2 for motor direction from pin 8 and Pin 9 Arduino Uno, then the DC motor is connected to the output of M1 and M2 of the motor driver. The hardware design that will be made for data collection of DC motor modeling using system identification is shown in Fig. 3.



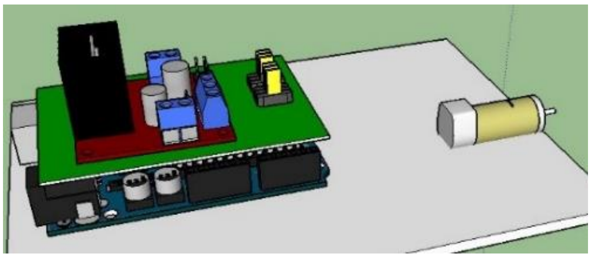


Fig. 2. Wiring Diagram

Fig. 3. Hardware Design

Software design includes block diagram design of PID control system with PSO algorithm. The data flow relationship between the PSO algorithm and the PID control is shown in Fig. 4. The variable in the figure is an error (e), rise time (t_r), settling time (t_s), overshoot time (O_s), and steady state error (sse).

In designing this device, the Particle Swarm Optimization Algorithm is used to determine the most optimal PID control parameters, with input in the form of PID parameter tuning using the Particle Swarm Optimization (PSO) algorithm to find the PID parameter constants. The working system is applied according to the block diagram of the PID control system shown in Fig. 4. Usually, after the PID parameters are calculated using the tuning method, these parameters need to be tuned again to get the best results. The plant used is a feedback control system whose output is the speed of a DC motor.

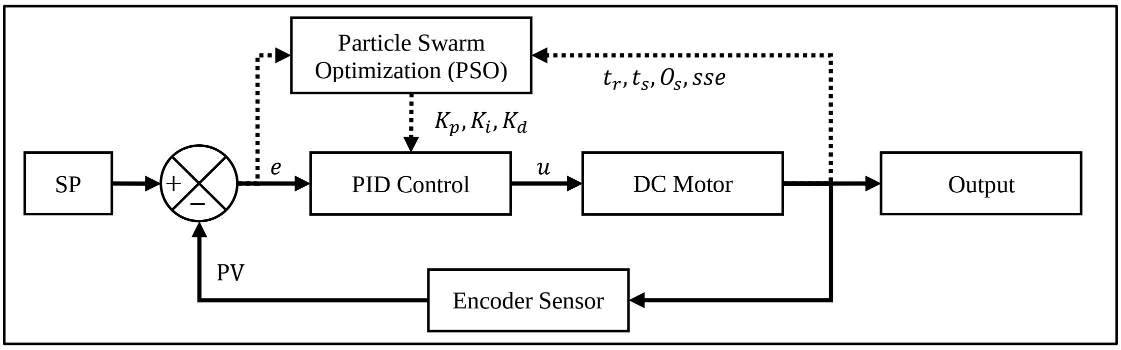


Fig. 4. Block Diagram of PID control system with PSO algorithm

After the data is identified, the next step is the design of the Simulink software using MATLAB software, which is used to see the response to the data obtained on the DC motor modeling block that has been made. The Simulink model can be seen in Fig. 5.

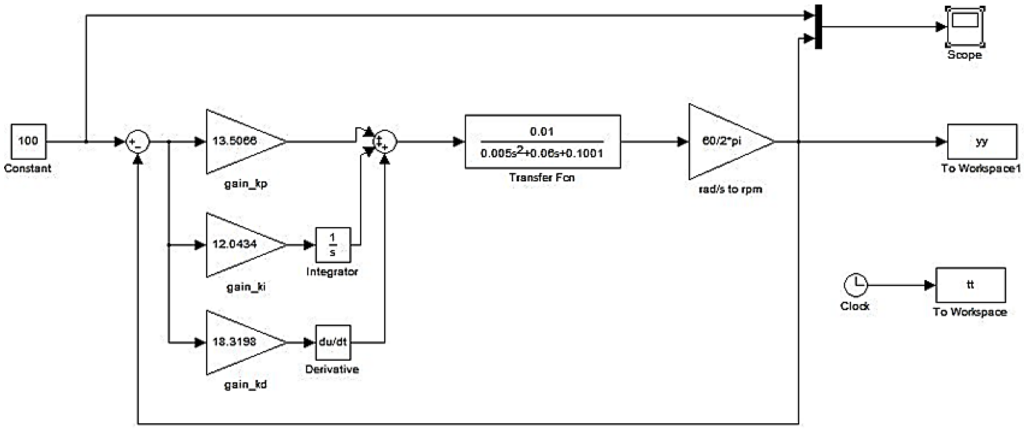


Fig. 5. Simulation in Simulink MATLAB

3. Result and Discussion

In this system, testing is carried out to find out the results of the system design and the test plan made in the previous chapter. Testing this system is done by testing the software (software) and hardware (hardware). The testing phase is carried out using the trial-error method, testing the PSO (Particle Swarm Optimization) algorithm method, and testing Hardware.

3.1. Trial and Error Method

In testing with the trial-error method, the researcher uses Simulink from MATLAB, which is to enter the values of K_p, K_i, and K_d using the trial-and-error method. The following simulation uses the trial-error method with the input values of $K_p = 10$, $K_i = 25$, and $K_d = 17$. The graph is obtained as shown in Fig. 6. It is obtained that the rise time = 0.8692, settling time = NaN, and overshoot = 51.1442. From the three graphs in the test using the trial-error method, the values for rise time, settling time, and overshoot are shown in Table 2. It can be seen that the test three times taking data using the trial-error method showed poor results with a fairly high overshoot. The overshoot that looks the best among the five data is $K_p = 18.7$, $K_i = 4.8$, and $K_d = 15.5$, with a rise time of 1.2717, settling time of NaN, and overshoot of 11.7563.

Table 2. Testing Result of Trial-Error Method

No	PID Controller Parameters			Rise Time	Settling Time	Overshoot
	K_p	K_i	K_d			
1.	10	25	17	0.8692	NaN	51.1442

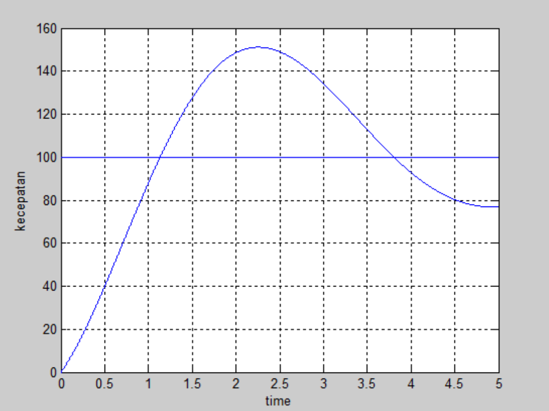


Fig. 6. Trial-Error Simulation Graph Result

3.2. PSO Method

In testing the tuning method with the PSO (Particle Swarm Optimization) algorithm, there are several stages of testing. The testing phase carried out is testing the parameters of the PSO algorithm, including testing the cognitive acceleration constant (c1), testing the social acceleration constant (c2), and testing the number of G values (generation).

In testing the cognitive acceleration constant (c1), it is entering the parameter value c1 with a value of 0.8. The graph can be seen as shown in Fig. 7. It shows the tuning results with the PSO algorithm, the rise time = 0.0740, the settling time = 0.1361, and the overshoot = 0.

In testing the social acceleration constant (c2), which is entering the parameter value c2 with a value of 0.8. The graph can be seen in Fig. 8. It shows the system response from the PID controller, which is tuned using the PSO algorithm with a rise time = 0.3628 seconds, settling time = 2.6483 seconds, and overshoot = 7.3809.

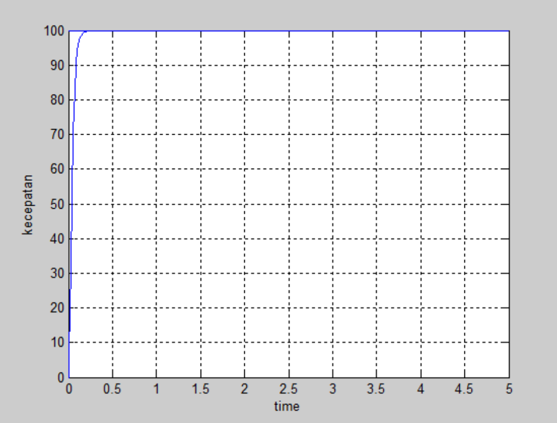


Fig. 7. Result of c1 PSO Testing

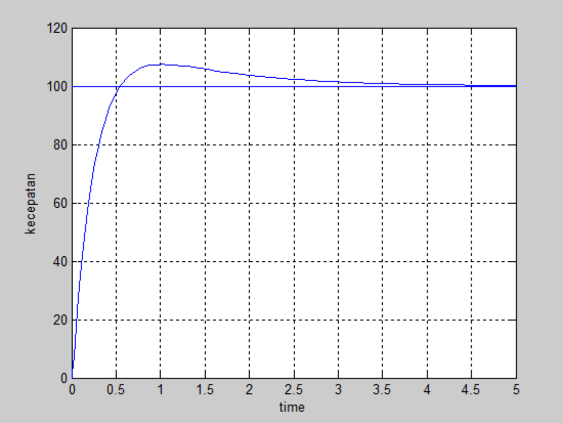


Fig. 8. Result of c2 PSO Testing

In testing the number of G values (generations), the G parameter value is with a value of 30. The graph can be seen in Fig. 9. It shows the system response from the PID controller, which is tuned using the PSO algorithm with the rise time = 0.6521 seconds, settling time = 3.9944, and overshoot = 14.5578.

From the three graphs in the c1 test, c2 testing, and testing the number of G values using the tuning method with the PSO algorithm, the rise time, settling time, and overshoot values are obtained in Table 3. It can be seen that the results of the test-taking data using the tuning method with the PSO algorithm showed better results than the previous method, namely the trial-error method. From the experiment on testing the cognitive acceleration constant (c1), there were very good results, where $c1 = 0.8$, $K_p = 8.9099$, $K_i = 2.1469$ and $K_d = 0.31952$ with rise time = 0.0740, settling time = 0.1361, and overshoot = 0. Then on testing the social acceleration constant (c2), there are quite good results, where $c2 = 0.8$; $K_p = 8.9099$; $K_i = 2.1469$ and $K_d = 0.31952$ with rise time is 0.3628; settling time is 2.6483, and overshoot is 7.3809. In testing the number of G values (generations), there are quite good results, with a value of $G = 30$; $K_p = 2.204$; $K_i = 1.7331$ and $K_d = 0.99411$ with rise time = 0.6521, settling time = 3.9944 and overshoot = 14.5578.

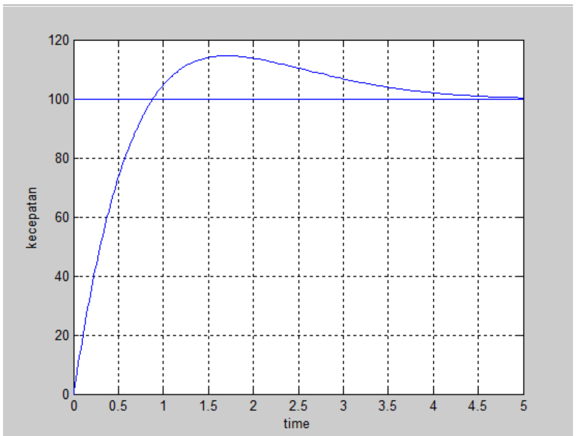


Fig. 9. Result of G PSO Testing

Table 3. Test Results c1, c2, and G

No.	C1	C2	G	K_p	K_i	K_d	Rise Time	Settling Time	Over-shoot
1.	0.8	0.9	10	8.9099	2.1469	0.31952	0.0740	0.1361	0
2.	0.9	0.8	10	1,9929	1.645	0.37808	0.3628	2.6483	7.3809
3.	0.7	0.9	30	2.204	1.7331	0.99411	0.6521	3.9944	14.5578

3.3. Hardware Examination

In this hardware test, there are several stages of testing carried out, namely testing by entering the PID parameter values that have been obtained from simulations on MATLAB software by tuning using the trial-error method and the PSO algorithm method.

3.3.1. Trial-Error Method

The following hardware testing is done by entering the K_p, K_i, and K_d values obtained from the trial-error method on the Arduino IDE software, which is complete with the PID controller program listing. Fig. 10 is a graph of hardware testing that produces a speed response for a motor with a not-good rise time, settling time, and overshoot value. From the input of the trial-error value, the first one produces a graph with a rise time of 1.7403 seconds, settling time = NaN, and overshoot = 19.

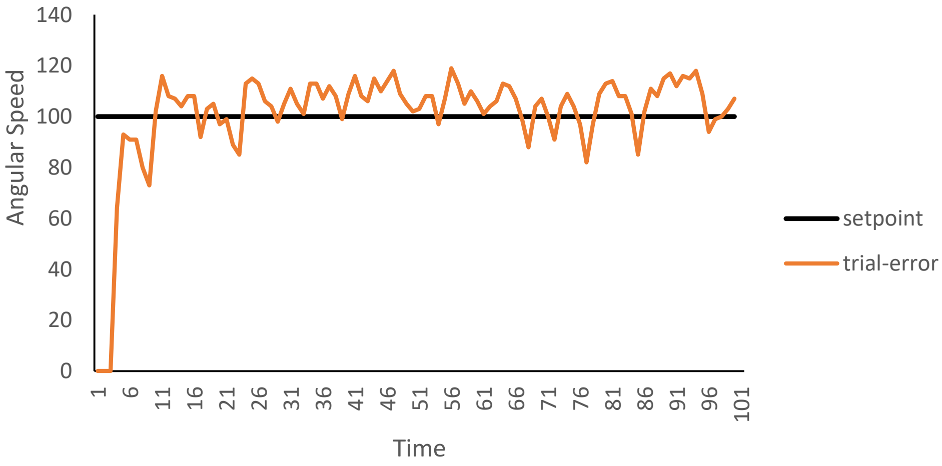


Fig. 10. Response of Trial-Error Method

3.3.2. Cognitive Acceleration Constant (c1) PSO Testing

The following hardware test is to enter the values of K_p, K_i and K_d obtained from the PSO c1 method on the Arduino IDE software, which is complete with the PID controller program listing, a graph like Fig. 11 is obtained.

Fig. 11 is a hardware test graph that produces a motor speed response with good rise time, settling time, and overshoot values. From the input value c1 = 0.7, it produces a graph with a rise time of 4.3296 seconds; settling time = 7.3333 seconds, and overshoot = 1.

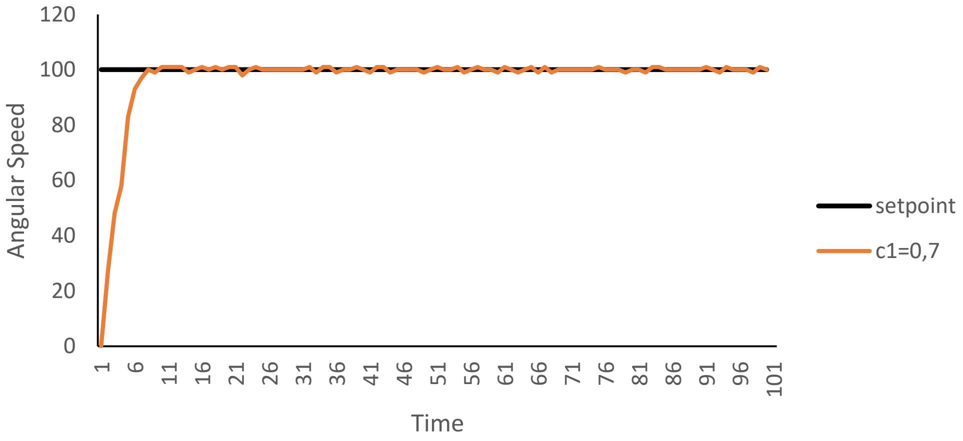


Fig. 11. Response of c1 PSO Testing

3.3.3. Social Acceleration Constant (c2) PSO Testing

The following hardware test is to enter the values of K_p, K_i, and K_d obtained from the PSO c2 method on the Arduino IDE software, which is complete with the PID program listing. Fig. 12 is a hardware test graph that produces a motor speed response with a fairly good rise time, settling time, and overshoot value. From the input value c2 = 0.8, it produces a graph with a rise time of 4.3288 seconds, a settling time of 9.5000 seconds, and an overshoot of 2.

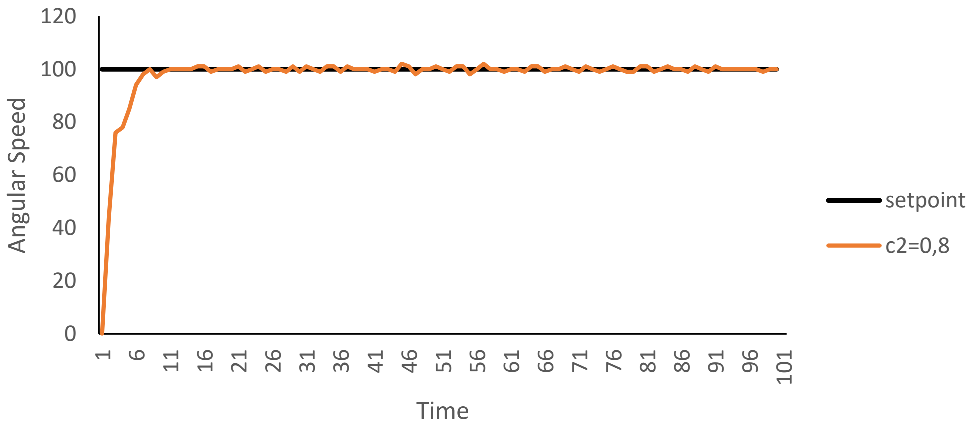


Fig. 12. Response of c2 PSO Testing

3.3.4. Generation (G) PSO Testing

The following hardware test is to enter the values of K_p, K_i, and K_d obtained from the PSO G method on the Arduino IDE software, which is complete with the PID program listing. Fig. 13 is a hardware test graph that produces a motor speed response with a fairly good rise time, settling time, and overshoot value. From the input value, G = 50 produces a graph with a rise time of 1.5979 seconds, settling time = 5.6667 seconds, and overshoot = 2.

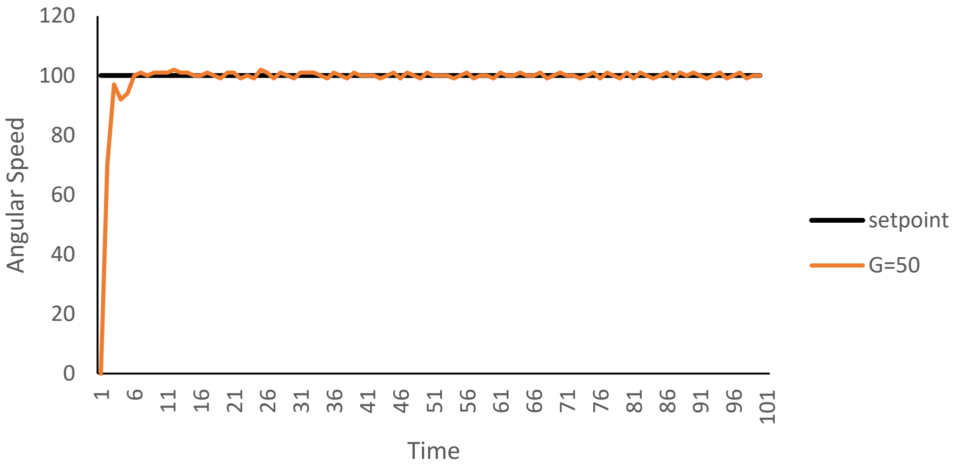


Fig. 13. Response of G=50 PSO Testing

From the 4 testing stages used in this hardware test, the following are the results of the comparison of the rise time, settling time, and overshoot values shown in Table 3. It can be seen that the comparison of the simulation test and hardware testing shows the results of the rise time, settling time, and overshoot values that are different. In testing the trial-error method, the results of the overshoot are unstable, with values of K_p = 50, K_i = 25, and K_d = 30 with rise time = 1.7403, settling time = NaN, and overshoot = 19. In testing the PSO method, the overshoot results are stable with K_p = 1.4551, K_i = 1.3079 and K_d = 0.80271 with rise time = 4.3288, settling time = 7.3333 and overshoot=1. In testing the PSO c2 method, the overshoot results are seen to be stable with a value of K_p = 1.9929, K_i = 1.645, and K_d = 0.37808 with rise time = 4.3288, settling time = 9.5000 and overshoot=2. In testing the PSO Generation method, the overshoot results are stable with K_p = 0.80636, K_i = 0.8774 and K_d = 0.62891 with rise time = 1.5979, settling time = 5.6667 and overshoot = 2.

Table 4. Test results comparison of PID values

No	Method	Gain K_p	Gain K_i	Gain K_d	Rise Time	Settling Time	Over-shoot
1.	Trial-Error	50	25	30	1.7403	NaN	19
2.	C1	1.4551	1.3079	0.80271	4.3296	7.3333	1
3.	C2	1.9929	1.645	0.37808	4.3288	9.5000	2
4.	G	0.80636	0.8774	0.62891	1.5979	5.6667	2

4. Conclusions

After doing research and testing, both software and hardware testing, it can be concluded the following things. The PSO algorithm is effective in tuning the PID parameter and implementation in hardware. Tuning the PID controller parameters using the PSO method on Simulink MATLAB obtained optimal results where the value of K_p = 8.9099; K_i = 2.1469 and K_d = 0.31952 with a rise time value = 0.0740, settling time = 0.1361 and overshoot = 0. The results of hardware testing by entering the PID value in the Arduino IDE software produce a stable motor speed response where K_p = 1.4551, K_i = 1.3079, and K_d = 0.80271 with a rise time value = 4.3296; settling time=7.3333 and overshoot = 1. A comparison of simulation testing and hardware testing with the same K_p, K_i, and K_d parameter values show different rise time, settling time, and overshoot values. If the overshoot in the simulation test gets a value of zero, in the hardware test, it gets a value greater than the Simulink value.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research was funded by Universitas Ahmad Dahlan grant number PD-212/SP3/LPPM-UAD/2020

Acknowledgment: The authors would like to thank Naufal Rahmat Setiawan, Siti Fatimah Anggraini, Eka Widya Suseno, Mila Diah Ika Putri, Ryan Istiarno, that helped to finish this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] I. S. Okoro and C. O. Enwerem, "Robust control of a DC motor," *Heliyon*, vol. 6, no. 12, 2020, https://doi.org/10.1016/j.heliyon.2020.e05777.

[2] M. L. Dezaki, S. Hatami, A. Zolfagharian, and M. Bodaghi, "A pneumatic conveyor robot for color detection and sorting," *Cognitive Robotics*, vol. 2, pp. 60–72, 2022, https://doi.org/10.1016/j.cogr.2022.03.001.

[3] M. M. Ahmed, W. S. Hassanien, and M. A. Enany, "Modeling and evaluation of SC MPPT controllers for PVWPS based on DC motor," *Energy Reports*, vol. 7, pp. 6044–6053, 2021, https://doi.org/10.1016/j.egyr.2021.09.055.

[4] Hendra, S. Pebriyanto, Hernadewita, Hermiyetti, and Yoserizal, "Applying Programmable Logic Control (PLC) for Control Motors, Blower and Heater in the Rubber Drying Processing," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 1, pp. 131–141, 2021, https://doi.org/10.26555/jiteki.v7i1.20514.

[5] M. Auzan, R. M. Hujja, M. R. Fuadin, and D. Lelono, "Path Tracking and Position Control of Nonholonomic Differential Drive Wheeled Mobile Robot," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 3, pp. 368–379, 2021, https://doi.org/10.26555/jiteki.v7i3.21017.

[6] A. Maarif, R. D. Puriyanto, and F. R. T. Hasan, "Robot Keseimbangan Dengan Kendali Proporsional-Integral-Derivatif (PID) dan Kalman Filter," *IT Journal Research and Development (ITJRD)*, vol. 4, no. 2, pp. 117–127, 2020, https://doi.org/10.25299/itjrd.2020.vol4(2).3900.

[7] R. Ashokkumar, M. Suresh, B. Sharmila, H. Panchal, C. Gokul, K. V. Udhayanatchi, K. K. Sadasivuni, and M. Israr, "A novel method for Arduino based electric vehicle emulator," *International Journal of Ambient Energy*, 2021, https://doi.org/10.1080/01430750.2020.1860129.

[8] D. Puangdownreong, "Fractional Order PID Controller Design for DC Motor Speed Control System via Flower Pollination Algorithm," *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, vol. 17, no. 1, pp. 14–23, 2019, https://doi.org/10.37936/ecti-eec.2019171.215368.

[9] E. Eker, M. Kayri, S. Ekinci, and D. Izci, "A New Fusion of ASO with SA Algorithm and Its Applications to MLP Training and DC Motor Speed Control," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3889–3911, 2021, https://doi.org/10.1007/s13369-020-05228-5.

[10] T. H. Mohamed, M. A. M. Alamin, and A. M. Hassan, "Adaptive position control of a cart moved by a DC motor using integral controller tuned by Jaya optimization with Balloon effect," *Computers & Electrical Engineering*, vol. 87, p. 106786, 2020, https://doi.org/10.1016/j.compeleceng.2020.106786.

[11] T. P. Cabré, A. S. Vela, M. T. Ribes, J. M. Blanc, J. R. Pablo, and F. C. Sancho, "Didactic platform for DC motor speed and position control in Z-plane," *ISA Transactions*, 2021, https://doi.org/10.1016/j.isatra.2021.02.020.

[12] H. E. A. Ibrahim, F. N. Hassan, and A. O. Shomer, "Optimal PID control of a brushless DC motor using PSO and BF techniques," *Ain Shams Engineering Journal*, vol. 5, no. 2, pp. 391–398, 2014, https://doi.org/10.1016/j.asej.2013.09.013.

[13] S. E. I. Hasseni, L. Abdou, and H. E. Glida, "Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms," *Evolutionary Intelligence*, vol. 14, no. 1, pp. 61–73, 2019, https://doi.org/10.1007/s12065-019-00312-8.

[14] M. H. A. Yaseen and H. J. Abd, "Modeling and control for a magnetic levitation system based on SIMLAB platform in real time," *Results in Physics*, vol. 8, pp. 153–159, 2018, https://doi.org/10.1016/j.rinp.2017.11.026.

[15] J. Yoon and J. Doh, “Optimal PID control for hovering stabilization of quadcopter using long short term memory,” *Advanced Engineering Informatics*, vol. 53, p. 101679, 2022, https://doi.org/10.1016/j.aei.2022.101679.

[16] J. J. Wang and T. Kumbasar, “Optimal PID control of spatial inverted pendulum with big bang-big crunch optimization,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 822–832, 2020, https://doi.org/10.1109/JAS.2018.7511267.

[17] D. Bakria, M. Azzouzi, and D. Gozim, “Chaos Control and Stabilization of a PID Controlled Buck Converter Using the Spotted Hyena Optimizer,” *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7922–7926, 2021, https://doi.org/10.48084/etasr.4585.

[18] T. Kobaku, R. Jeyasenthil, S. Sahoo, R. Ramchand, and T. Dragicevic, “Quantitative Feedback Design-Based Robust PID Control of Voltage Mode Controlled DC-DC Boost Converter,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 286–290, 2021, https://doi.org/10.1109/TCSII.2020.2988319.

[19] 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, https://doi.org/10.1016/j.heliyon.2022.e09399.

[20] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, “A review of PID control, tuning methods and applications,” *International Journal of Dynamics and Control*, vol. 9, pp. 818–827, 2021, https://doi.org/10.1007/s40435-020-00665-4.

[21] A. Kherkhar, Y. Chiba, A. Tlemçani, and H. Mamur, “Thermal investigation of a thermoelectric cooler based on Arduino and PID control approach,” *Case Studies in Thermal Engineering*, vol. 36, p. 102249, 2022, https://doi.org/10.1016/j.csite.2022.102249.

[22] F. Xu, X. Liang, M. Chen, and W. Liu, “Robust Self-Learning PID Control of an Aircraft Anti-Skid Braking System,” *Mathematics*, vol. 10, no. 8, p. 1290, 2022, https://doi.org/10.3390/math10081290.

[23] P. Božek and Y. Nikitin, “The Development of an Optimally-Tuned PID Control for the Actuator of a Transport Robot,” *Actuators*, vol. 10, no. 8, p. 195, 2021, https://doi.org/10.3390/act10080195.

[24] J. Sun, H. Zhou, X. Ma, and Z. Ju, “Study on PID tuning strategy based on dynamic stiffness for radial active magnetic bearing,” *ISA Transactions*, vol. 80, pp. 458–474, 2018, https://doi.org/10.1016/j.isatra.2018.07.036.

[25] Y. Fan, J. Shao, G. Sun, and X. Shao, “Improved Beetle Antennae Search Algorithm-Based Lévy Flight for Tuning of PID Controller in Force Control System,” *Mathematical Problems in Engineering*, vol. 2020, 2020, https://doi.org/10.1155/2020/4287315.

[26] D. Potnuru, K. Alice Mary, and C. Sai Babu, “Experimental implementation of Flower Pollination Algorithm for speed controller of a BLDC motor,” *Ain Shams Engineering Journal*, vol. 10, no. 2, pp. 287–295, 2019, https://doi.org/10.1016/j.asej.2018.07.005.

[27] B. B. Alagoz, F. N. Deniz, and M. Koseoglu, “An efficient PID-based optimizer loop and its application in De Jong’s functions minimization and quadratic regression problems,” *Systems & Control Letters*, vol. 159, p. 105090, 2022, https://doi.org/10.1016/j.sysconle.2021.105090.

[28] W. Farag, “Complex Trajectory Tracking Using PID Control for Autonomous Driving,” *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 2, pp. 356–366, Sep. 2019, https://doi.org/10.1007/s13177-019-00204-2.

[29] S. Ulusoy, S. M. Nigdeli, and G. Bekdaş, “Novel metaheuristic-based tuning of PID controllers for seismic structures and verification of robustness,” *Journal of Building Engineering*, vol. 33, p. 101647, 2021, https://doi.org/10.1016/j.jobe.2020.101647.

[30] H. Du, P. Liu, Q. Cui, X. Ma, and H. Wang, “PID Controller Parameter Optimized by Reformative Artificial Bee Colony Algorithm,” *Journal of Mathematics*, vol. 2022, pp. 1–16, 2022, https://doi.org/10.1155/2022/3826702.

[31] H. Wang, H. Du, Q. Cui, and H. Song, “Artificial bee colony algorithm based PID controller for steel stripe deviation control system,” *Science Progress*, vol. 105, no. 1, pp. 1–22, 2022, https://doi.org/10.1177/00368504221075188.

[32] J. Bhookya, M. Vijaya Kumar, J. Ravi Kumar, and A. Seshagiri Rao, "Implementation of PID controller for liquid level system using mGWO and integration of IoT application," *Journal of Industrial Information Integration*, vol. 28, p. 100368, 2022, https://doi.org/10.1016/j.jii.2022.100368.

[33] B. N. Kommula and V. R. Kota, "Direct instantaneous torque control of Brushless DC motor using firefly Algorithm based fractional order PID controller," *Journal of King Saud University - Engineering Sciences*, vol. 32, no. 2, pp. 133–140, 2020, https://doi.org/10.1016/j.jksues.2018.04.007.

[34] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, J. Álvarez-Gallegos, and M. Aldape-Pérez, "Bio-inspired adaptive control strategy for the highly efficient speed regulation of the DC motor under parametric uncertainty," *Applied Soft Computing*, vol. 75, pp. 29–45, 2019, https://doi.org/10.1016/j.asoc.2018.11.002.

[35] M. M. Gani, M. S. Islam, and M. A. Ullah, "Optimal PID tuning for controlling the temperature of electric furnace by genetic algorithm," *SN Applied Sciences*, vol. 1, no. 8, pp. 1–8, 2019, https://doi.org/10.1007/s42452-019-0929-y.

[36] J. Bhookya and R. K. Jatoth, "Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine-algorithm," *Evolutionary Intelligence*, vol. 12, no. 4, pp. 725–733, 2019, https://doi.org/10.1007/s12065-019-00290-x.

[37] B. Hekimoğlu, "Sine-cosine algorithm-based optimization for automatic voltage regulator system," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 6, pp. 1761–1771, 2018, https://doi.org/10.1177/0142331218811453.

[38] Y. Zhou, J. Zhang, X. Yang, and Y. Ling, "Optimization of PID Controller Based on Water Wave Optimization for an Automatic Voltage Regulator System," *Information Technology and Control*, vol. 48, no. 1, pp. 160–171, 2019, https://doi.org/10.5755/j01.itc.48.1.20296.

[39] F. Loucif, S. Kechida, and A. Sebbagh, "Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 1, pp. 1–11, 2019, https://doi.org/10.1007/s40430-019-2074-3.

[40] B. Guo, Z. Zhuang, J. S. Pan, and S. C. Chu, "Optimal Design and Simulation for PID Controller Using Fractional-Order Fish Migration Optimization Algorithm," *IEEE Access*, vol. 9, pp. 8808–8819, 2021, https://doi.org/10.1109/ACCESS.2021.3049421.

[41] J. Agarwal, G. Parmar, R. Gupta, and A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsystem Technologies*, vol. 24, no. 12, pp. 4997–5006, 2018, https://doi.org/10.1007/s00542-018-3920-4.

[42] K. Vanchinathan and N. Selvaganesan, "Adaptive fractional order PID controller tuning for brushless DC motor using Artificial Bee Colony algorithm," *Results in Control and Optimization*, vol. 4, p. 100032, 2021, https://doi.org/10.1016/j.rico.2021.100032.

[43] L. Chaib, A. Choucha, and S. Arif, "Optimal design and tuning of novel fractional order PID power system stabilizer using a new metaheuristic Bat algorithm," *Ain Shams Engineering Journal*, vol. 8, no. 2, pp. 113–125, 2017, https://doi.org/10.1016/j.asej.2015.08.003.

[44] S. Kadry and V. Rajinikanth, "Design of PID Controller for Magnetic Levitation System using Harris Hawks Optimization," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 6, no. 2, p. 70, 2021, https://doi.org/10.26555/jiteki.v6i2.19167.

[45] A. H. Mary, A. H. Miry, and M. H. Miry, "An Optimal Robust State Feedback Controller for the AVR System-Based Harris Hawks Optimization Algorithm," *Electric Power Components and Systems*, vol. 48, no. 16–17, pp. 1684–1694, 2021, https://doi.org/10.1080/15325008.2021.1908456.

[46] L. Jia and X. Zhao, "An Improved Particle Swarm Optimization (PSO) Optimized Integral Separation PID and its Application on Central Position Control System," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 7064–7071, 2019, https://doi.org/10.1109/JSEN.2019.2912849.

[47] A. Parnianifard, A. S. Azfanizam, M. K. A. Ariffin, M. I. S. Ismail, M. R. Maghami, and C. Gomes, "Kriging and Latin hypercube sampling assisted simulation optimization in optimal design of PID controller for speed control of DC motor," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 5, pp. 1471–1479, 2018, https://doi.org/10.1166/jctn.2018.7379.

[48] Y. G. Rashid and A. M. A. Hussain, "Implementing optimization of PID controller for DC motor speed

control," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 657–664, 2021, https://doi.org/10.11591/ijeecs.v23.i2.pp657-664.

[49] S. Ekinci, B. Hekimoğlu, and D. Izci, "Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor," *Engineering Science and Technology, an International Journal*, vol. 24, no. 2, pp. 331–342, 2021, https://doi.org/10.1016/j.jestch.2020.08.011.

[50] M. S. N. Krishna konijeti and M. L. Bharathi, "Extraction of maximum power from solar with BLDC motor driven electric vehicles based HHO algorithm," *Advances in Engineering Software*, vol. 170, p. 103137, 2022, https://doi.org/10.1016/j.advengsoft.2022.103137.

[51] A. K. Kashyap and D. R. Parhi, "Particle Swarm Optimization aided PID gait controller design for a humanoid robot," *ISA Transactions*, vol. 114, pp. 306–330, 2021, https://doi.org/10.1016/j.isatra.2020.12.033.

[52] Z. Xiang, D. Ji, H. Zhang, H. Wu, and Y. Li, "A simple PID-based strategy for particle swarm optimization algorithm," *Information Sciences*, vol. 502, pp. 558–574, 2019, https://doi.org/10.1016/j.ins.2019.06.042.

[53] H. Feng, W. Ma, C. Yin, and D. Cao, "Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller," *Automation in Construction*, vol. 127, p. 103722, 2021, https://doi.org/10.1016/j.autcon.2021.103722.

[54] S. M. H. Mousakazemi and N. Ayoobian, "Robust tuned PID controller with PSO based on two-point kinetic model and adaptive disturbance rejection for a PWR-type reactor," *Progress in Nuclear Energy*, vol. 111, pp. 183–194, 2019, https://doi.org/10.1016/j.pnucene.2018.11.003.

[55] P. Dutta and S. K. Nayak, "Grey Wolf Optimizer Based PID Controller for Speed Control of BLDC Motor," *Journal of Electrical Engineering & Technology*, vol. 16, no. 2, pp. 955–961, 2021, https://doi.org/10.1007/s42835-021-00660-5.

[56] M. A. Shamseldin, R. Barbosa, and I. Jesus, "Optimal Coronavirus Optimization Algorithm Based PID Controller for High Performance Brushless DC Motor," *Algorithms*, vol. 14, no. 7, p. 193, 2021, https://doi.org/10.3390/a14070193.

[57] B. Naidu Kommula and V. Reddy Kota, "Design of MFA-PSO based fractional order PID controller for effective torque controlled BLDC motor," *Sustainable Energy Technologies and Assessments*, vol. 49, p. 101644, 2022, https://doi.org/10.1016/j.seta.2021.101644.

[58] H. R. R. Zaman and F. S. Gharehchopogh, "An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems," *Engineering with Computers*, pp. 1–35, 2021, https://doi.org/10.1007/s00366-021-01431-6.

[59] Z. Qi, Q. Shi, and H. Zhang, "Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-Based DC motor subject to stochastic delays," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5637–5646, 2020, https://doi.org/10.1109/TIE.2019.2934030.

[60] X. Zhou, J. Zhou, C. Yang, and W. Gui, "Set-Point tracking and Multi-Objective Optimization-Based PID control for the goethite process," *IEEE Access*, vol. 6, pp. 36683–36698, 2018, https://doi.org/10.1109/ACCESS.2018.2847641.

[61] Y. Li, K. H. Ang, and G. C. Y. Chong, "PID Control System Analysis and Design," *IEEE Control Systems*, vol. 26, no. 1, pp. 32–41, 2006, https://doi.org/10.1109/MCS.2006.1580152.