Tuning of PID Controller Parameters with Genetic Algorithm Method on DC Motor

Eka Widya Suseno [a,1], Alfian Ma'arif [a,2,*]

[a] Universitas Ahmad Dahlan, Bantul, Yogyakarta, Indonesia
[1] eka1600022028@webmail.uad.ac.id; [2] alfianmaarif@ee.uad.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received 06 February 2021
Revised 15 February 2021
Accepted 21 February 2021

Keywords
PID controller;
Genetic Algorithm (GA);
DC Motor;
Arduino;
Tuning;
Matlab

ABSTRACT

Proportional Integral Derivative (PID) controllers are used in general to control a system, for example a DC motor system. The difficulty of using the controller is parameter tuning, because the tuning parameters still use the trial and error method to find the PID parameter constants, namely Proportional Gain (KP), Integral Gain (KI) and Derivative Gain (KD). In this case, the genetic algorithm method is used which can give better results in each iteration. Genetic algorithms are one of the smart methods inspired by the process of natural selection, the process that causes biological evolution, this concept is applied to tuning PID parameters. This research uses the Matlab simulation method and applies the simulation results to the DC motor hardware using the Arduino Uno. The genetic algorithm method gives a system that has a better steady time and a smaller maximum spike than the Trial and Error method. The test process produced the two best data with an overshoot value = 2, settling time = 13.5 and rise time of 2.7872 and the PID parameter value for mutation of 1 was KP = 3.7500; KI = 1.3184 and KD = 0.2051. Then the value of the best PID parameter on Crossover is 0.4, which is KP = 4.2090; KI = 1.2012 and KD = 0.2539 with an overshoot value = 2, settling time = 18 and rise time = 2.6462.

1. Introduction

Proportional Integral Derivative (PID) controllers are conventional controllers that are still widely used in the industrial world compared to more modern types of controls [1]. The PID controller has 3 parameters, namely KP (Proportional Constants), KI (Integral Constants), KD (Derivative Constants) in designing the controllers, the goal is to get the optimal system response according to the desired design specifications [2]. There are several methods of PID tuning, including the Ziegler-Nichols method [3] and the Fuzzy method [4][5]. Most of these tuning methods require a broad knowledge of the control system. So the need for an alternative to tuning automation and practitioners that do not require in-depth control system knowledge.

http://dx.doi.org/10.31763/ijrcs.v1i1.249

To overcome the problem, can be used methods such as intelligent Optimization Algorithm [6][7] or the Heuristic Method [8][9].

In this final project research, Genetic Algorithm is used to determine PID controller parameters so that the controller can produce a satisfactory system response [10]. Genetic Algorithm is an optimization technique based on natural evolution through mutation, crossover and selection processes [11]. Based on the previous research, genetic algorithm have better result than another algorithm [12]. After obtaining the system model, the Genetic Algorithm is used off-line by simulation to determine the PID controller parameters. Then, the parameter will implement in the DC motor System that handle by Microcontroller Arduino. The implementation is used to know the best method to determine the PID controller parameter id DC motor system.

2. Methods

3.1. Control System Block Diagram

In the design of this system, a genetic algorithm is used to determine the PID controller parameters. The PID controller block diagram with transfer function is shown in Fig. 1.

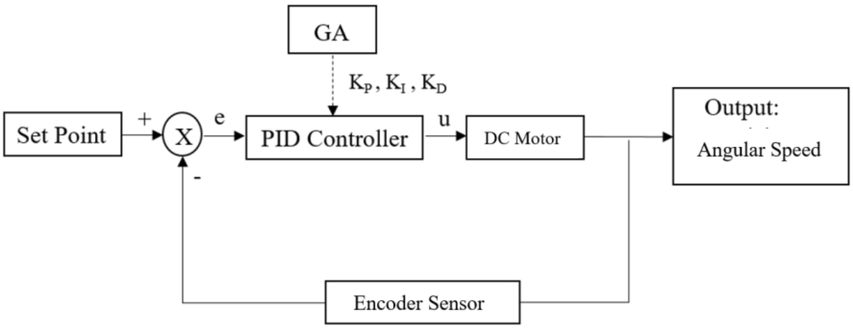

Fig. 1. PID controller tuning block diagram

Based on Figure 1, the workings of the system created are input in the form of a PID parameter tuning using a Genetic Algorithm (GA) as a control method to find PID parameter constants. The plant used is a feedback control system whose output is a speed from a DC motor.

3.2. System Design

In addition to the PID controller tuning block diagram in this study, a Hardware block diagram and hardware wiring were also made to show an explanation of the hardware used. Hardware block diagram is shown in Fig. 2.

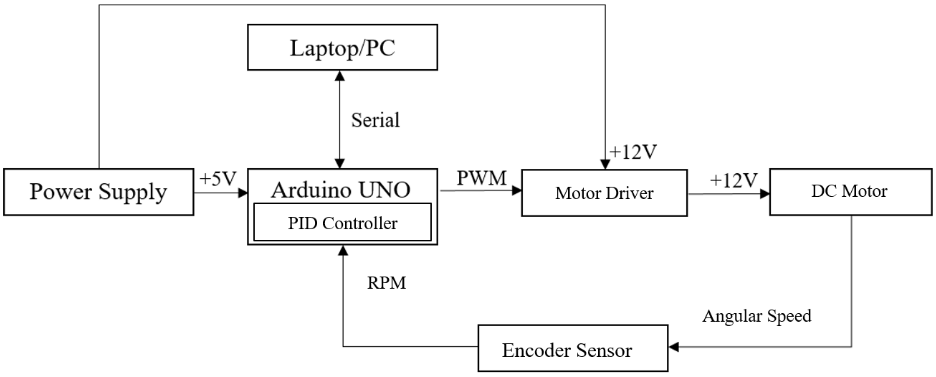

Fig. 2. Hardware Block Diagram

Figure 2 shows a block diagram of a DC motor modeling using the Arduino Uno R3 board as the system controller. The Arduino board gets a voltage supply from the power supply that has been prepared of 12 volts which is distributed to the motor driver to drive the DC motor, then the rotating motion of the DC motor will be recorded by the encoder which will be given to the Arduino Uno R3 so that the results can be seen in the Arduino Ide software on the menu serial monitor and serial plotter.

3.3. Wiring Diagram

The wiring diagram or wiring diagram of the system created can be seen in Fig. 3. Hardware design is made to control DC motors, and motor drivers to see the system response and also as a means of collecting the data needed to model DC motors using a system identification modeling approach, the data to be used are in the form of voltage and speed of the DC motor.

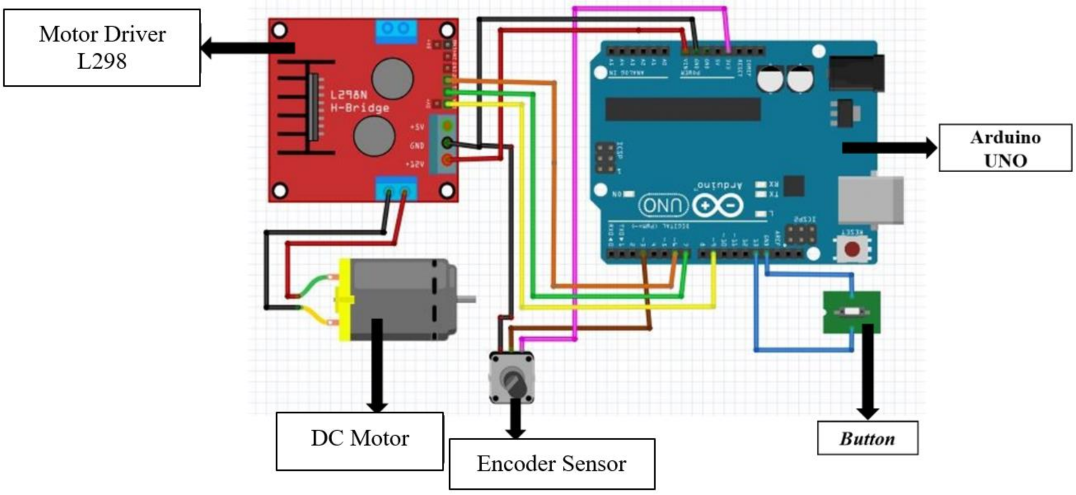

Fig. 3. Wiring Hardware

3.4. Genetic Algorithm

The Genetic Algorithm is depicted in the form of a flowchart shown in Fig. 4. Based on Figure 4, the main standard genetic algorithm includes the following four operators. GA selection is a selection operation that will select the parent solution. During the reproductive phase of GA, individuals are selected from the population and recombined, resulting in offspring which in turn will comprise the next generation. The crossover takes two parents and cuts the parent chromosome string at several randomly selected positions, to produce two “head” segments and two “tail” segments. The mutations are applied to each child individually, after the crossover.

3.Result and Discussion

In the section, the DC Motor Transfer Function model is taken from [13] as

$$\mathrm{G(s)} = \frac{\omega(s)}{v(s)} = \frac{0.01}{0.005s^2 + 0.06s + 0.1001} \tag{1}$$

where the input is voltage $v(s)$ and output is angular speed $\omega(s)$.

3.1. Testing using the Trial and Error Method

Tests with trial and error methods or trial and error methods on the values of KP, KI and KD were carried out to determine the comparison of the PID parameters obtained. Testing with this method uses a DC Motor Plant in Matlab as shown in Fig. 5.

The results of testing using the Trial and Error method are shown in Table 1. The test was carried out five times with the results in the form of a graphic on the DC motor Plant. The graph is shown in Fig. 6 and Fig. 7. Based on the result, the trial and error method have poor performance and response.

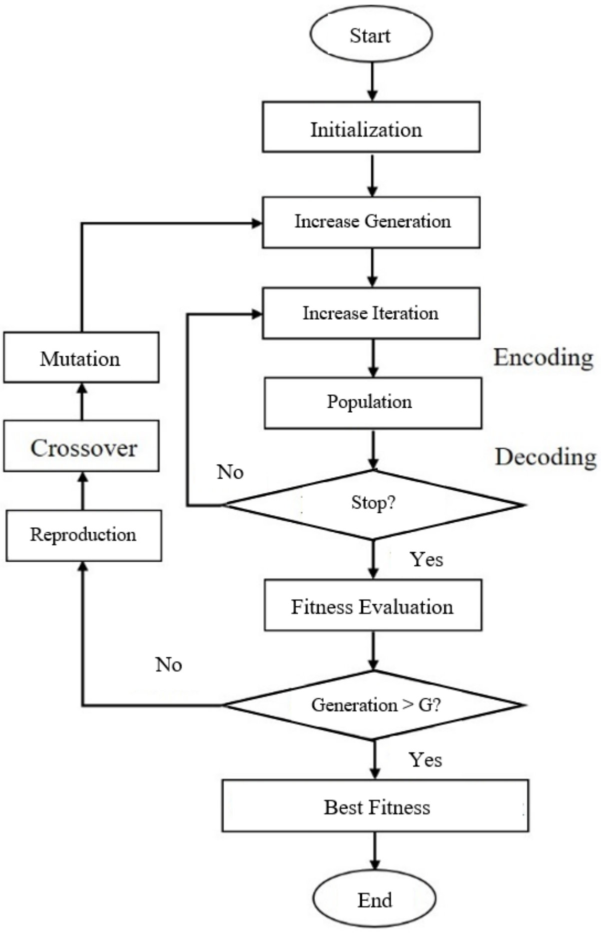

Fig. 4. Genetic algorithm program flowchart

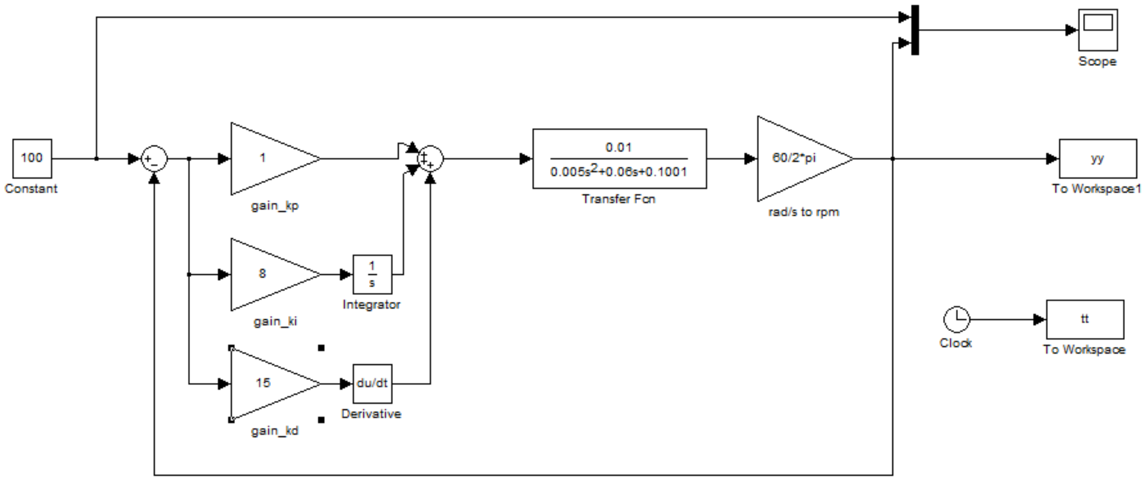

Fig. 5. DC Motor Plant at Matlab

Table 1. Testing 1 uses the Trial and Error Method

No.	PID Controller Parameter			Rise time	Settling Time	Overshoot
	KP	KI	KD			
1	1	8	15	1.4440	NaN	85.6825
2	10	25	15	0.8129	NaN	49.2620

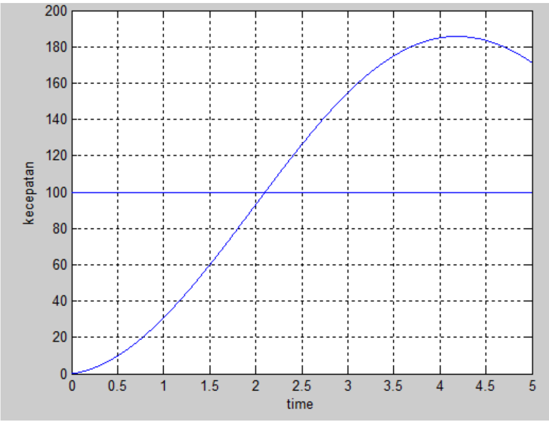

Fig. 6. Data Graph number 1 Trial and Error Testing

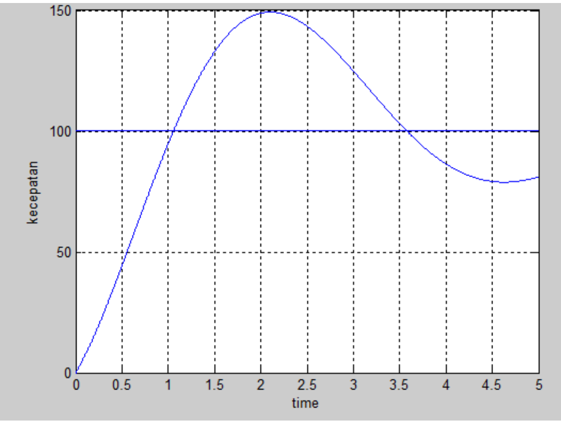

Fig. 7. Data Graph number 2 Trial and Error Testing

3.2. Testing using the Genetic Algorithm (GA) Method

Testing using the Genetic Algorithm method is carried out with three kinds of tests which are influenced by the function of the genetic algorithm itself. The functions of genetic algorithms include the mutation process, the process of crossing over and the number of generation variations. The testing process is run with Matlab.

3.2.1. Tests using GA with mutation effect

The effect of mutation is carried out by the process of entering a mutation value of 0.8 and 1. Then entering a crossover value of 0.6 in all the data tested and using the number of generations of 100.The fitness results and PID parameter for the DC Motor Transfer Function in the model selection equation are shown in Fig. 8 and Fig. 9.

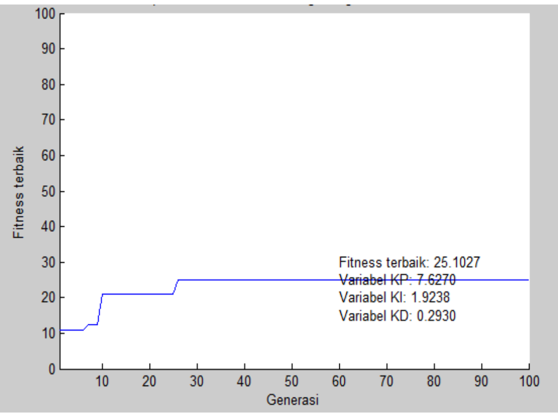

Fig. 8. The Fitness results and PID parameter with a mutation constant are 0.8

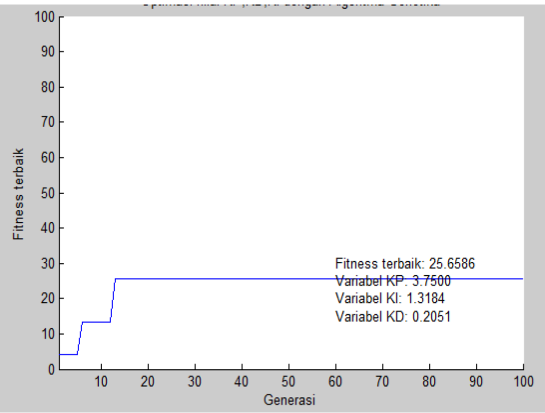

Fig. 9. The fitness results and PID parameter with a mutation constant are 1

The next test is to enter the PID parameter values (KP, KI, KD) obtained from the tuning results of mutations at the Matlab DC Motor Plant. The results displayed from the test are in the form of a graph shown in Fig. 10 and Fig. 11.

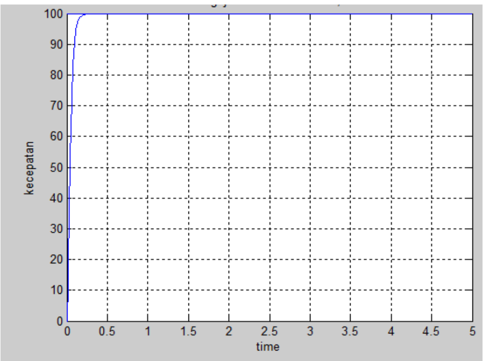

Fig. 10. Simulation results with a mutation constant are 0.8

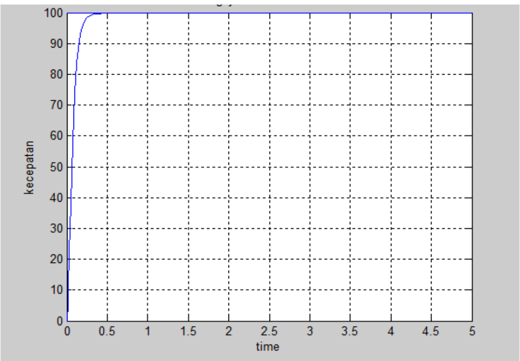

Fig. 11. Simulation results with a mutation constant are 1

The results shown in Fig. 10 and Fig. 11 can be considered better than the graph using the Trial and Error method. The results of the test using the GA Method with this mutation effect are shown in Table 2.

Table 2. Test 2 uses the GA method with mutation effect

No.	Mutation	Cross-over	Gene-rations	Best Fitness	KP gain	KI Gain	KD Gain	Rise time	Settling Time	Overshoot
1	0.8	0.6	100	25.1027	7.6270	1.9238	0.2930	0.0813	0.1500	0
2	1	0.6	100	25.6586	3.7500	1.3184	0.2051	0.1288	0.2428	0

From the results in Table 2, it is obtained that the overshoot value is 0. The overshoot obtained is optimal with the PID parameter value that meets the requirements, namely KP ≥ KI ≥ KD.

3.2.2. Tests using GA with crossover effect

Effect of crossovers do with the process of entering a value crossover 0.2 and 0.4. Next, enter a mutation value of 0.6 on all tested data and use the number of generations of 100. The fitness results and PID parameter for the DC Motor Transfer Function in the model selection equation are shown in Fig. 12 and Fig. 13.

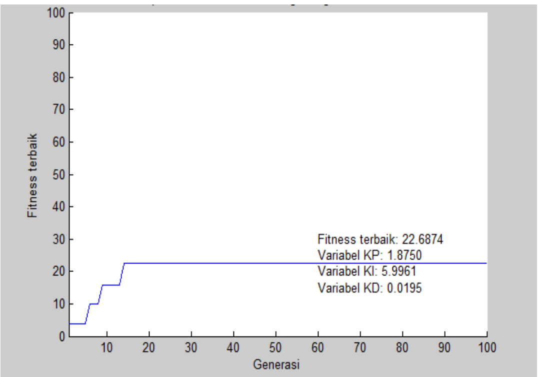

Fig. 12. The fitness results and PID parameter with the crossover constant are 0.2

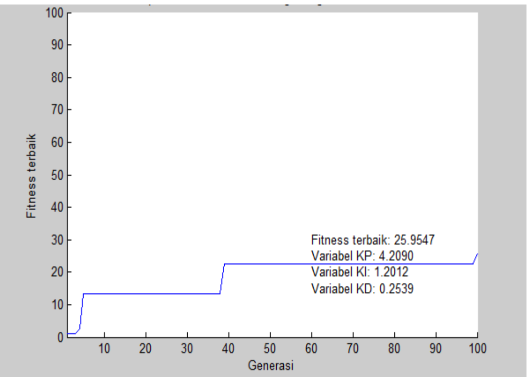

Fig. 13. The fitness and PID parameter results with the crossover constant are 0.4

The next test is to include the value of the PID parameters (KP, KI, KD) obtained from the tuning results crossovers in Matlab DC Motor Plant. The results displayed from the test are in the form of a graphic shown in Fig. 14 and Fig. 15.

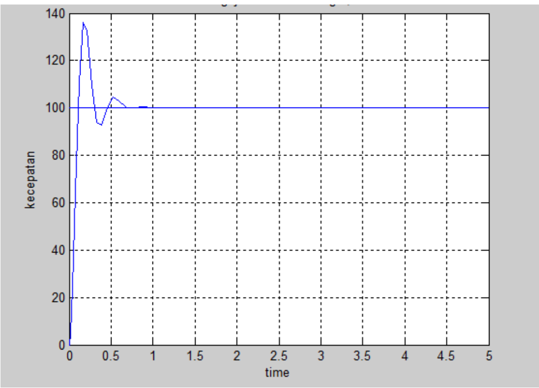

Fig. 14. The simulation results with the crossover constant are 0.2

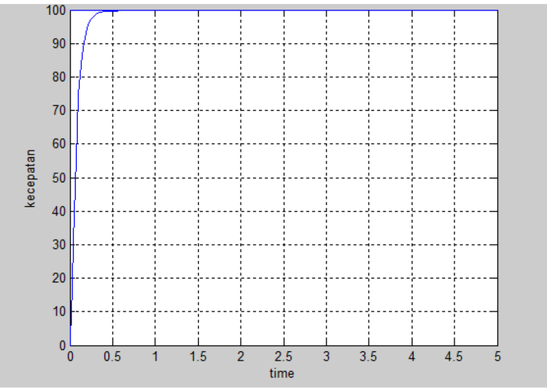

Fig. 15. The simulation results with the crossover constant are 0.4

The results shown in Fig. 14 and Fig. 15 can be considered better than the graph with the Trial and Error method, but not better than the graph using the GA Mutation method, which displays the graph without more overshoot. The results of the test using the GA method with the crossover effect are shown in Table 3.

Table 3. Testing 3 uses the GA Method with the crossover Effect

No.	Mutation	Cross-over	Gene-rations	Best Fitness	KP gain	KI Gain	KD Gain	Rise time	Settling Time	Overshoot
1	0.6	0.2	100	22.6874	1.8750	5.9961	0.0195	0.0746	0.6191	36.1315
2	0.6	0.4	100	25.9547	4.2090	1.2012	0.2539	0.1444	0.2737	0

From the results in Table 3, it is obtained an overshoot value of 0. The overshoot obtained is optimal with the PID parameter values that meet the requirements, namely KP ≥ KI ≥ KD.

3.2.3. Tests using GA with the Influence of the Number of Generations of Variation

The effect of the number of generations largely determines the PID parameter, this number of generations does not absolutely have to be bigger or smaller because it is determined by the movement of the best fitness value. The number of generations used in this simulation consists of five data, namely the number of generations of 120 and the number of generations of 150. Next, enter the value of mutation and crossover 0.6 in all data tested. The simulation results for the Transfer Function DC Motor in the model selection equation are shown in Fig. 17 and Fig. 18.

The next test is to enter the PID parameter values (KP, KI, KD) obtained from the simulation results of the Number of Generation Variations at the Matlab DC Motor Plant. The results displayed from the test are in the form of graphs shown in Fig. 18 and Fig. 19.

The results shown in the graph above can be considered better than the graph with the Trial and Error method and the graph with the GA Mutation and crossover methods because they display more overshoot. The results of testing using the GA method with the effect of the number of generations of variations are shown in Table 4.

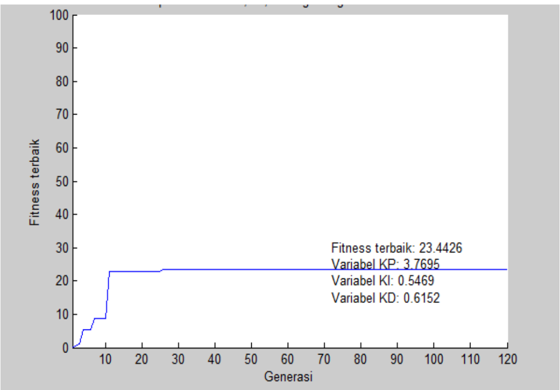

Fig. 16. The Fitness Results Number of Generations 120

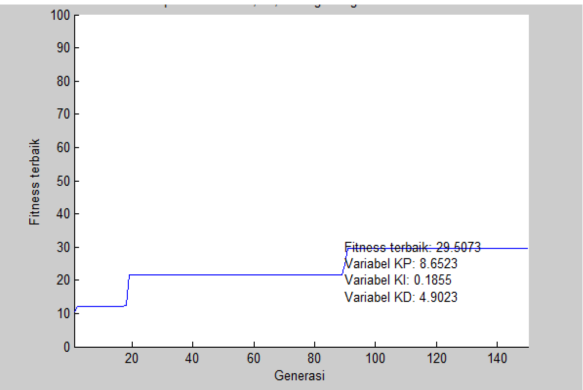

Fig. 17. The Fitness Results Number of Generation 150

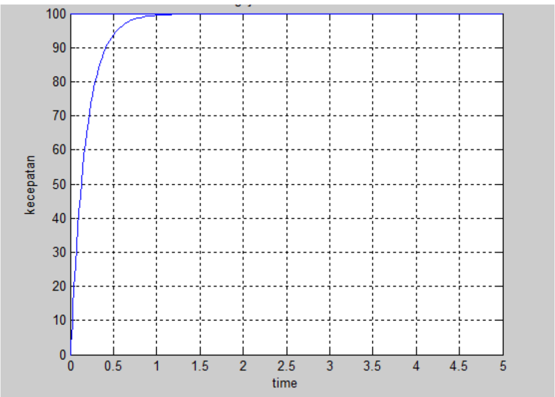

Fig. 18. Graph of Simulation Results Number of Generations 120

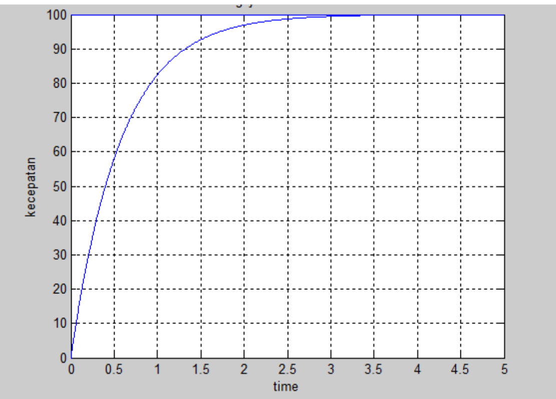

Fig. 19. Graph of Simulation Results for the Number of Generations 150

Table 4. Testing 4 uses the GA method with the effect of the number of generations of variations

No.	Mutation	Cross-over	Gene-rations	Best Fitness	KP gain	KI Gain	KD Gain	Rise time	Settling Time	Overshoot
1	0.6	0.6	120	23.4426	3.7695	0.5469	0.6152	0.3889	0.7122	0
2	0.6	0.6	150	29.5037	8.6532	0.1855	4.9025	1.2614	2.2480	0

From the results in Table 4, it is obtained an overshoot value of 0 on all data. The overshoot obtained is optimal with the PID parameter values that meet the requirements, namely KP ≥ KI ≥ KD.

3.3. Hardware Testing

The hardware used in this test is shown in Fig. 20. Hardware testing is carried out after the testing process 1 to test 4 gets results in the form of 6 data for PID parameter values (KP, KI, KD) using the Trial and Error and GA methods, then the data is tested on a tool that has been designed to get optimal results in accordance with the aim of the researcher. The hardware testing process is run with the Arduino application and to display system response information using Matlab. The results of this hardware test are shown in Table 5.

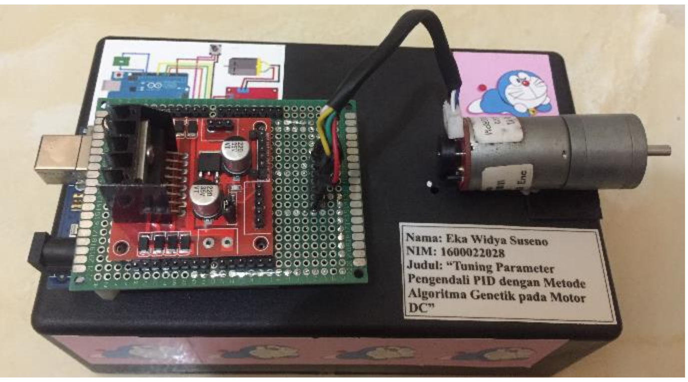
Nama: Eka Widya Suseno
NIM: 1600022028
Judul: "Tuning Parameter Pengendali PID dengan Metode Algoritma Genetik pada Motor DC"

Fig. 20. PID Control Hardware with DC Motor

Table 5. Comparison of PID Parameters on Testing 1 to 4 using Hardware

No.	Methods	KP Gain	KI Gain	KD Gain	Rise time	Settling Time	Overshoot
1	Trial and Error 1	1	8	15	2.2882	99.8000	25
2	Trial and Error 2	10	25	15	1.7756	NaN	20
3	Genetic Algorithm with Mutation 0.8	7.6270	1.9238	0.2930	1.3754	42.2500	12
4	Genetic Algorithm with Mutation 1	3.7500	1.3184	0.2051	2.7872	13.5000	2
5	Genetic Algorithm with crossover 0.2	1.8750	5.9961	0.0195	2.2917	94.2500	10
6	Genetic Algorithm with crossover 0.4	4.2090	1.2012	0.2539	2.6462	18.0000	2
7	Genetic Algorithm with number of Generations 120	3.7695	0.5469	0.6152	1.3200	24.3333	7
8	Genetic Algorithm Influence Number of Generation 150	8.6532	0.1855	4.9025	3.4228	NaN	15

Hardware testing is carried out 8 times according to the amount of data from the previous test. The sequence of these tests is in accordance with tests 1 to 4. The results of the hardware testing are in the form of a DC motor speed graph which will then be compared to get the smallest overshoot value. The graph is shown in Fig. 22 to Fig. 29.

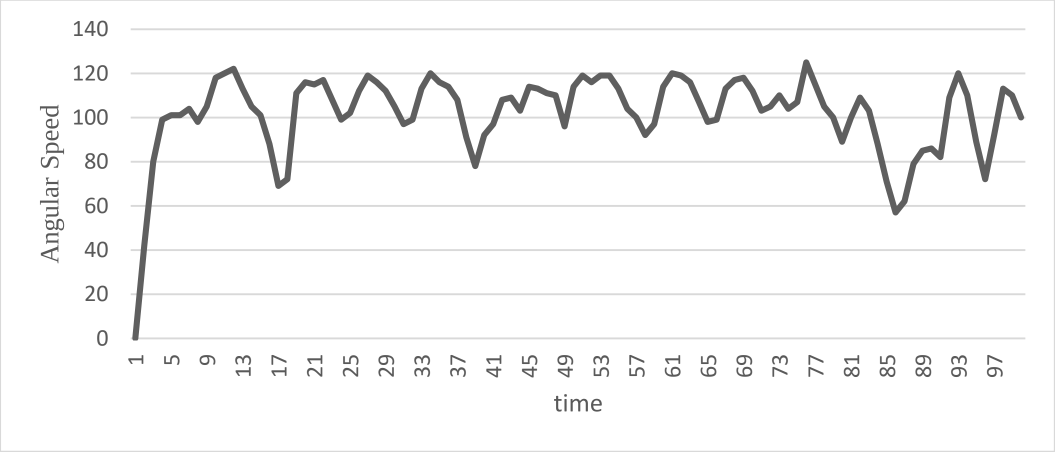

Fig. 21. Graph of Trial and Error Method 1 on Hardware Testing

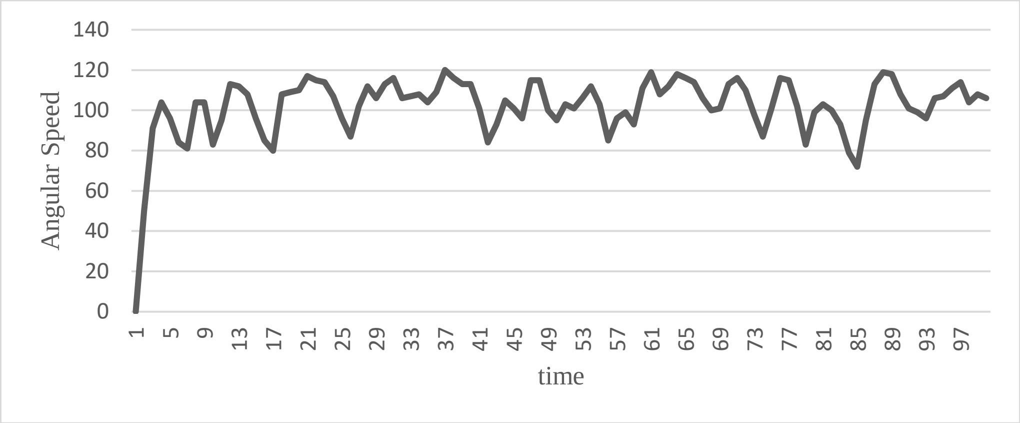

Fig. 22. Graph of Trial and Error 2 Method on Hardware Testing

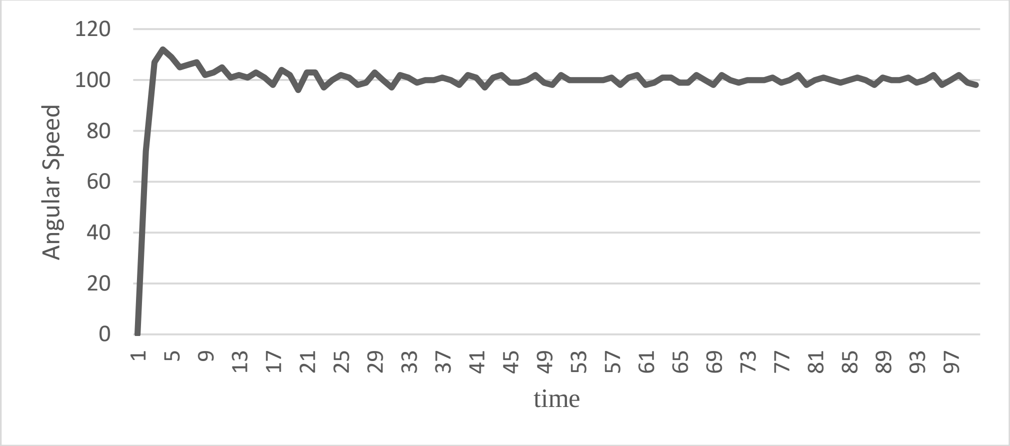

Fig. 23. Graph of the GA Mutation Method 0.8 on Hardware Testing

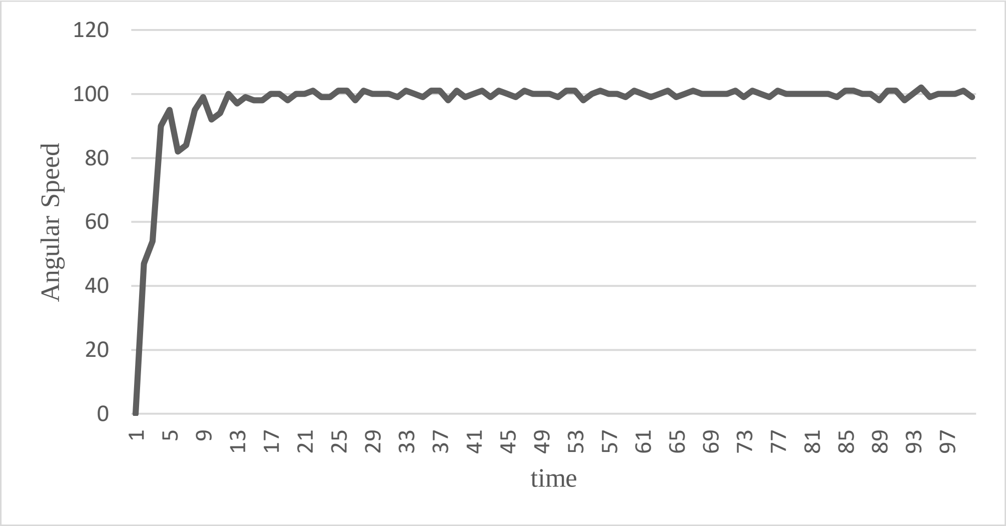

Fig. 24. Graph of the GA Mutation Method 1 on Hardware Testing

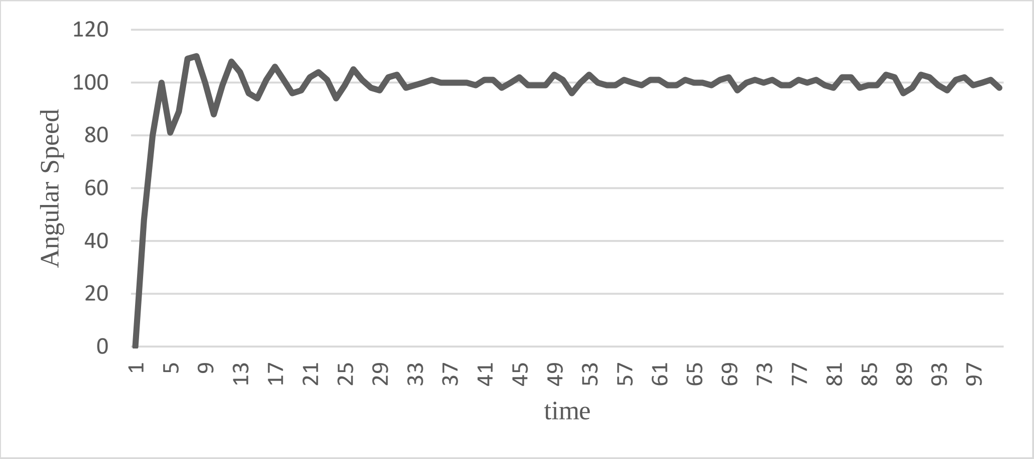

Fig. 25. Graph GA crossover 0.2 method on Hardware Testing

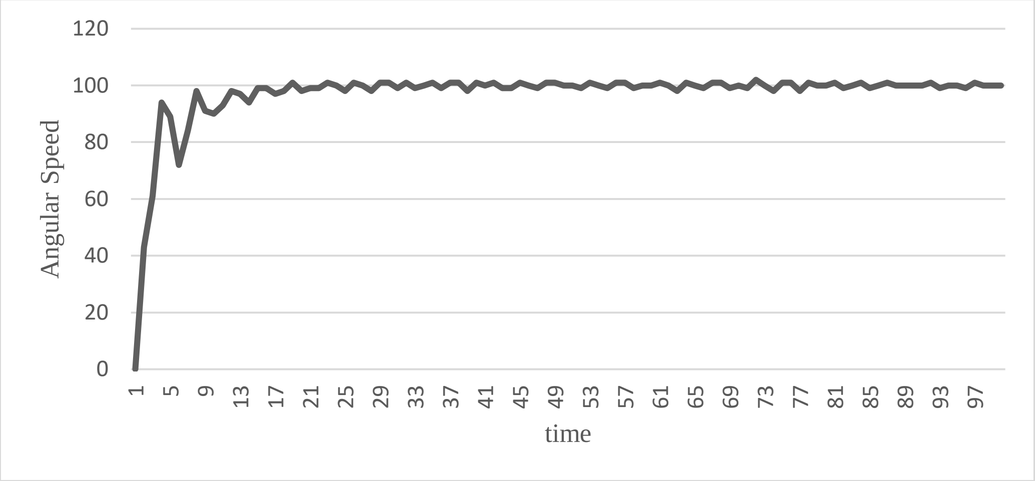

Fig. 26. Graph GA crossover 0.4 Method on Hardware Testing

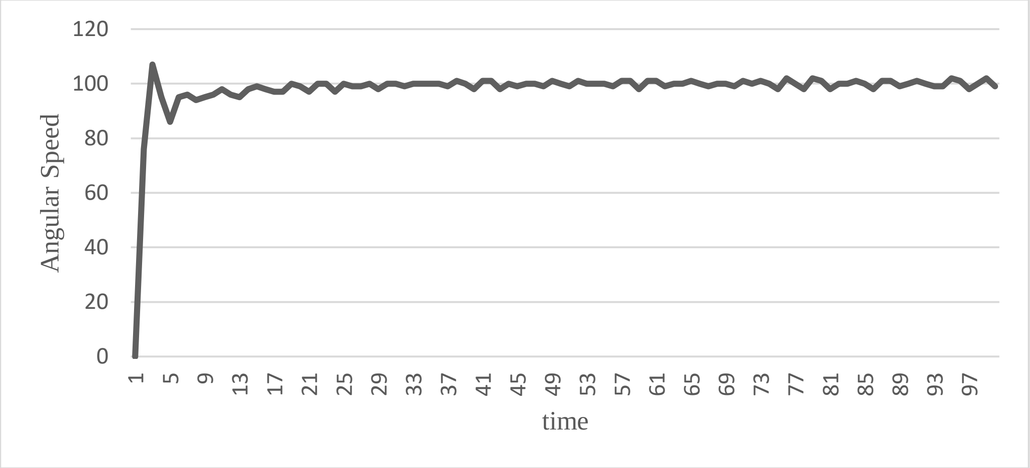

Fig. 27. Graph GA Method Number of Generation 120 on Hardware Testing

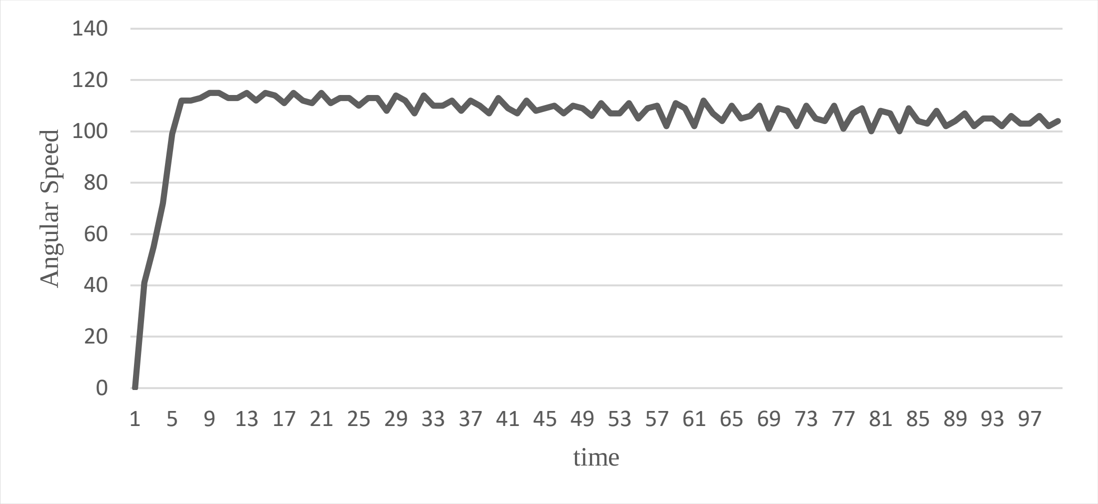

Fig. 28. Graph of GA Method Number of Generation 150 on Hardware Testing

The best PID parameters (Fig. 25) are KP = 3.7500; KI = 1.3184 and KD = 0.2051 with the overshoot value = 2, undershoot = 0 and, settling time = 13.5 and rise time = 2.7872. The other best PID parameters (Fig. 27) are KP = 4.2090; KI = 1.2012 and KD = 0.2539 with the overshoot value = 2, undershoot = 0 and for settling time = 18 and rise time = 2.6462.

4. Conclusion

After conducting research and processing the data in this study, the following conclusions can be drawn. The genetic algorithm for tuning PID controlling parameters has found the optimal solution with an overshoot below 10%, with the number of generations of 100 on the mutation effect of 0.4 and the crossover effect of 0.8. As well as the effect of the number of generations of variations of 120 and 150. The genetic algorithm method gives a system that has a better steady time and a smaller maximum spike than the Trial and Error method used in the simulation. The hardware testing process with the genetic algorithm method on a DC motor can run well, so that the best data is obtained in the form of an overshoot value below 10% and PID parameters that meet the requirements. The testing process produced the two best data with an overshoot value = 2 and the best PID parameter value at mutation 1 was KP = 3.7500; KI = 1.3184 and KD = 0.2051 and the best PID parameter value in crossover 0.4 is KP = 4.2090; KI = 1.2012 and KD = 0.2539.

References

[1] D. Somwanshi, M. Bundele, G. Kumar, and G. Parashar, “Comparison of fuzzy-PID and PID controller for speed control of DC motor using LabVIEW,” in *Procedia Computer Science*, 2019, vol. 152, pp. 252–260. https://doi.org/10.1016/j.procs.2019.05.019

[2] M. A. Taut, G. Chindris, and D. Pitică, “PID Algorithm used for DC Motor Control,” in *2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging, SIITME 2018 - Proceedings*, 2019, pp. 365–372. https://doi.org/10.1109/SIITME.2018.8599230

[3] E. Apriaskar, N. Azis Salim, and D. Prastiyanto, “Performance Evaluation of Balancing Bicopter using P, PI, and PID Controller,” *Jurnal Teknik Elektro*, vol. 11, no. 2, pp. 44–49, Dec. 2019. https://doi.org/10.15294/jte.v11i2.23032

[4] A. Y. Al-Maliki and K. Iqbal, “FLC-based PID controller tuning for sensorless speed control of DC motor,” in *Proceedings of the IEEE International Conference on Industrial Technology*, 2018, vol. 2018-Febru, pp. 169–174. https://doi.org/10.1109/ICIT.2018.8352171

[5] E. Flores-Morán, W. Yánez-Pazmiño, and J. Barzola-Monteses, "Genetic algorithm and fuzzy self-tuning PID for DC motor position controllers," in *Proceedings of the 2018 19th International Carpathian Control Conference, ICCC 2018*, 2018, pp. 162–168. https://doi.org/10.1109/CarpathianCC.2018.8399621

[6] J. Agarwal, G. Parmar, and R. Gupta, "Comparative Analysis of PID Controller for Speed Control of DC motor with Intelligent Optimization Algorithms," in *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 2018, pp. 273–277. https://doi.org/10.1109/ICACCCN.2018.8748475

[7] S. Ekinci, D. Izci, and B. Hekimoglu, "PID Speed Control of DC Motor Using Harris Hawks Optimization Algorithm," in *2nd International Conference on Electrical, Communication and Computer Engineering, ICECCE 2020*, 2020. https://doi.org/10.1109/ICECCE49384.2020.9179308

[8] A. Ahmed, R. Gupta, and G. Parmar, "GWO/PID Approach for Optimal Control of DC Motor," in *2018 5th International Conference on Signal Processing and Integrated Networks, SPIN 2018*, 2018, pp. 181–186. https://doi.org/10.1109/SPIN.2018.8474105

[9] D. Maddi, A. Sheta, D. Davineni, and H. Al-Hiary, "Optimization of PID Controller Gain Using Evolutionary Algorithm and Swarm Intelligence," in *2019 10th International Conference on Information and Communication Systems, ICICS 2019*, 2019, pp. 199–204. https://doi.org/10.1109/IACS.2019.8809144

[10] L. Angel, J. Viola, and M. Vega, "Hardware in the loop experimental validation of PID controllers tuned by genetic algorithms," in *4th IEEE Colombian Conference on Automatic Control: Automatic Control as Key Support of Industrial Productivity, CCAC 2019 - Proceedings*, 2019. https://doi.org/10.1109/CCAC.2019.8921229

[11] K. Hakiim, A. Dharmawan, and F. Faizah, "Optimasi Kendali PID menggunakan Algoritma Genetika untuk Penerbangan Quadrotor," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 7, no. 2, p. 173, Oct. 2017. https://doi.org/10.22146/ijeis.23937

[12] D. K. Meena and S. Chahar, "Speed control of DC servo motor using genetic algorithm," in *IEEE International Conference on Information, Communication, Instrumentation and Control, ICICIC 2017*, 2018, vol. 2018-Janua, pp. 1–7. https://doi.org/10.1109/ICOMICON.2017.8279122

[13] A. Ma'arif, H. Nabila, Iswanto, and O. Wahyunggoro, "Application of Intelligent Search Algorithms in Proportional-Integral-Derivative Control of Direct-Current Motor System," in *Journal of Physics: Conference Series*, 2019, vol. 1373, no. 1, pp. 1–10. https://doi.org/10.1088/1742-6596/1373/1/012039