DC Motor Rotary Speed Control with Arduino UNO Based PID Control

Rikwan [1], Alfian Ma'arif [2,*]
[1, 2] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
Email: [1] Rikwanudding007@gmail.com, [2] alfianmaarif@ee.uad.ac.id
*Corresponding Author

***Abstract*— Along with the development of the times, DC motors are often used in industrial equipment or household appliances, but in DC motors they often experience a decrease due to the given load, it requires a controller. This research uses PID (Proportional Derivative) controller. In this study, the DC motor can be controlled despite the load using the trial and error method. This study uses Arduino UNO software for testing using parameters $Kp = 1.5$, $Ki = 0.87$, $Kd = 0.27$, parameter y is the parameter value Kp, Ki, Kd obtained from the system response according to the software used. the value of rise time = 0.9925 Tested, Time = 2.7368, Overshoot = 1.3333 and Steady State Error = 0**

Keywords*— DC Motors; Arduino UNO PID (Proportional Integral Derivative).

I. Introduction

In today's era, the need for motors that have great efficiency, large torque, large and variable speeds, and relatively low maintenance budgets continues to grow. However, the motors that are universally produced at this time, are DC motors and induction motors, so these needs have not been met. In DC motors, the motor drive uses DC voltage on the rotor [1].

But on the other hand, DC motors require large maintenance costs. Large maintenance costs arise because DC motors use brushes for commutation. The brush quickly faces destruction because when the motor turns around, on the brush, arching will arise on the brush due to the commutation process. Induction motors require lower maintenance costs and have greater speed than DC motors [2][3].

With the development of technology, various microcontrollers have become widespread in the market today. Starting from the cost, limitations on expertise or memory types as well as other facilities provided from the manufacturer that are easy and suitable to apply as desired. The use of microcontrollers in the process of control or automatic control has benefited many areas in terms of efficiency, accuracy, and effectiveness, as well as in terms of accuracy, reliability, and speed of the creation process can continue to grow [4][3].

DC motor is one of the easy-to-apply motors. This convenience makes DC motors often used in all kinds of purposes, including industrial equipment or household appliances. However, in its application, the speed of the DC motor often occurs due to a load so that the speed is unstable [5][13].

PID (Proportional–Integral– Derivative) controllers are universal controllers used in various industrial processes. The popularity of PID controllers results from their good performance over a wide range of conditions due to the simplicity of the PID, which allows engineers to operate them directly. To implement the PID controller, three parameters must be set in the controlled process which include proportional gain, integral gain, and derivative gain [6][7][14].

Today, the tech world often uses Differential Integral Proportional (PID) control as a feedback controller. Because it is tested to be able to share control performance well, can speed up system reactions, and eliminate offsets, although it has a simple but easy-to-understand algorithm [8][15].

In the study, the speed of DC motors is often abnormal due to external constraints or changes in parameters from the fabrication, so it is necessary to try the controller design. The controller designed using PID consists of 3 types of settings combined, namely the P (Proportional), I (Integral), and D (Derivative) controls. Switching to the PID control set on the keypad and the best time (Time Sampling) obtained to obtain stability from the coveted DC motor speed ATMega328 Microcontroller [6][16].

This study explains the adjustment of a single-phase induction motor in this study utilizing a Proportional Integral Derivative (PID) controller to adjust the motor speed to maintain a constant speed along with load changes. This speed setting is controlled by PWM pulses to obtain a frequency corresponding to the reference speed. In this study, no-load testing obtained motor rise time will accelerate at smaller reference speeds [9][17].

In this study, it was focused on determining the value of PID control parameters that worked optimally using the manual trial error tuning method using Aspen Hysis V8,6. Research was conducted to determine whether PID control could work optimally in stabilizing errors if given disturbances in temperatures that had been set in the reactants [10][18] .

In this study, we will design a DC motor control model using several kinds of controls, namely PID control tuned with autotuning Matlab, PID led with Ant Colony Optimization (ACO) to control speed [11][19].

This study uses a PID control system using a feedback encoder as feedback based on microcontroller Arduino UNO and its application to a no-load DC motor by using visual studio application as an interface that functions to input set point values in PID parameters and present data in graphs in real time [12][20].

II. METHODS

The design of this system is carried out with two stages of design, namely hardware design and software design. Hardware design required block diagrams, circuit diagrams, and tool designs. The design of the software contains a flow chart of the method used.

In this study, a process of identifying problems in the DC motor was carried out, where the rotation of the DC motor needs to be controlled at speed in accordance with the function and use of the motor. The factors that are considered in this study are motor speed control and calculation of controlling values. To get optimal results, a study was carried out and subsequently the appropriate hardware was made and the manufacture of software to operate the system from the driver motor and DC motor. After that, tool testing is carried out to obtain data to be processed and compared with previous studies. The research flow chart of the automatic feeding system can be seen in Fig. 1.

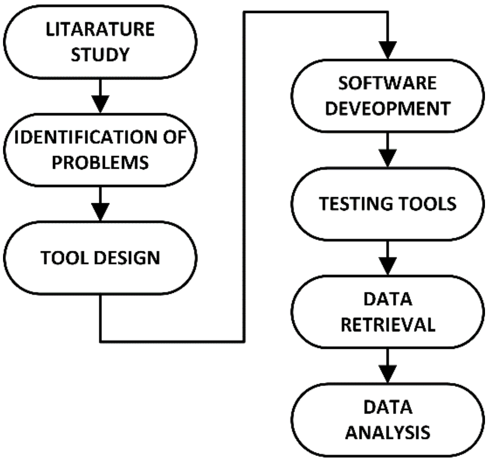

Fig. 1. Research Flowchart

A. *Hardware Design*

The hardware design of this dc motor's angular velocity control system rests on the Arduino Uno controller which controls all sensors and actuators. The hardware design block diagram can be seen in Fig. 2.

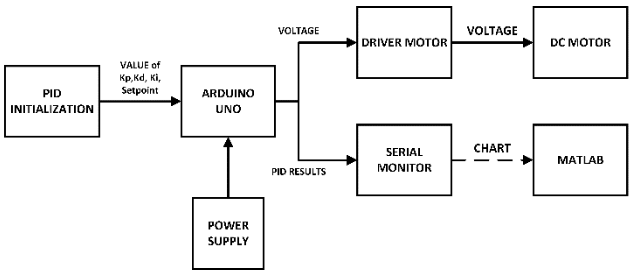

Fig. 2. System design block diagram

The design of this DC motor angular velocity control system uses an Arduino UNO microcontroller to regulate a DC motor that has an encoder in it, Arduino will send a voltage signal to the driver motor to be subsequently discharged to the DC motor according to the voltage calculation results of the PID controller. The voltage value in the DC motor is influenced by the PID controller, depending on the setpoint value and the value of each Proportional-integral-derivative controller, the result of the PID controller is the angular velocity of the DC motor.

PID control results in addition to the rotation results on the DC motor, this study also displays the PID control results through a graph displayed by the serial monitor, data from the serial monitor will later be processed to in the application MATLAB will be used to calculate the results of the PID controller so that the values of the PID control parameters are obtained, such as the calculation of risetime, settling time, overshoot, undershoot and its peak value. Hardware design begins with creating a wiring diagram of the components used and continues with hardware assembly. Assembly is n the process of connecting the wiring of all t-gauge so that the reading of the value of the driver motor and the rotation of the DC motor is obtained. The wiring diagram of the entire system can be seen in Fig. 3.

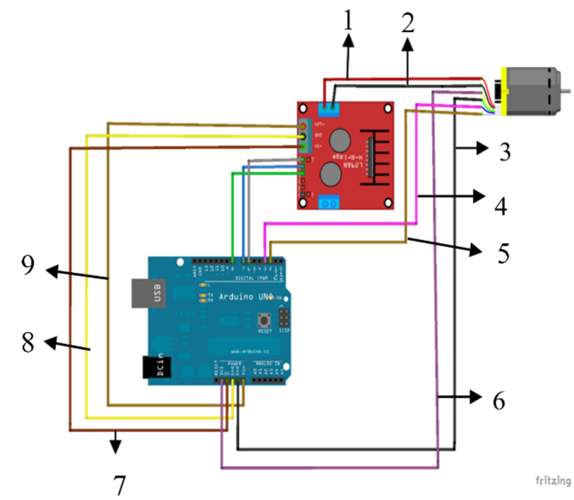

Fig. 3. Wiring diagram

Fig. 3 is the wiring circuit of this system, all components are connected by the Arduino UNO with a power supply coming from a 12 Volt adapter. The motor driver and DC motor are directly connected with the Arduino pins. The pins used on the Arduino UNO can be seen in Table I.

TABLE I. ARDUINO UNO INPUT AND OUTPUT PINS

No.	Pi	Information
1	+ Driver	+ Driver to motor Encoder +
2	- Driver	- Driver to Motor Encoder -
3	GND	GND Arduino to GND Encoder
4	D3	Battery D3 to Battery Encoder Chanel B motor DC
5	D2	Pin D3 to pin Encoder Chanel A motor DC
6	3,3	VCC 3,3 Volt Encoder
7	5	VCC Motor Driver 5 Volt
8	GND	Ground Motor Driver & Ground Encoder
9	Wire	VCC 12 Volt Motor Driver

B. *3D Design*

Mechanical manufacturing in this system begins with 3D creation to get an estimate of the size of the hardware. The design includes the entire mechanics of the DC motor speed system in the form of a black plastic box, this system will have two boxes, of which the first box contains a DC motor, the second box will contain arduino UNO and a driver motor. The two boxes at the top will be pinned to be paired with a jack tire, later this system will be uninstalled, so that users can easily understand and use this system. The two boxes will later be connected to each other with a cable through the jack hole at the top of the box. The 3D design of the overall system can be seen in Fig. 4.

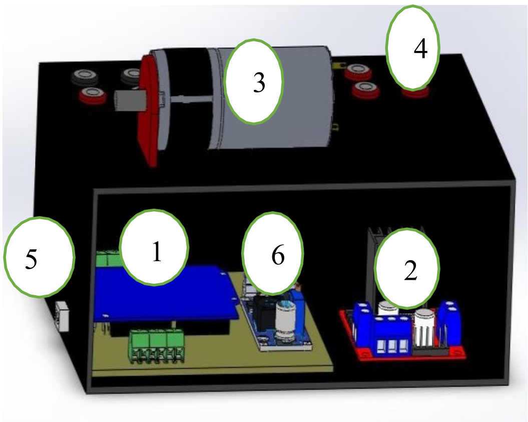

Fig. 4. 3D Design

Fig. 4 is a 3D design of the control box of the DC motor rotary speed control system. The frame-shaped system is made of plastic boxes and arranged in such a way following a 3D design. A more detailed explanation can be seen in Table II.

TABLE II. SYSTEM CONTROL BOX DESCRIPTION

No	Component	Information
1	Arduino	As a system control center
2	Engine Drive (L298N)	As a link between arduino and DC motor
3	PG28	As an Object of Research on Speed Control
4	Banana Jack	As a place for connections between cables and between boxes
5	Adapt Jack	System Power Supply from USB
6	Step Down (LM2596)	As a voltage lowering from Power Supply 12 V to 5 V Arduino Uno

C. *Software design*

Software design, that is, the program for the Arduino UNO controller is created through the Arduino IDE application and downloaded via a USB cable to the node board. Fig. 5 is an explanation of the system flow chart.

Based on Fig. 5, it can be understood that the motor speed control system uses the PID control method, starting with the inclusion of the values of the coefficients P, I and D and the desired setpoint value. The program will process the PID input and later the physical output is in the form of the angular speed of the DC motor, the speed of the DC motor is basically obtained from the voltage given after being given a PID controller. The speed of the DC motor will always be monitored and detected by the encoder sensor, the encoder will always calculate what speed the motor is rotating, if the speed of the motor is different from the PID value (Reference Value) then the encoder will send the next error value, the error value will be processed and corrected, the repair results will again be issued by the Arduino in the form of DC motor rotation in accordance with the PID value (reference value).

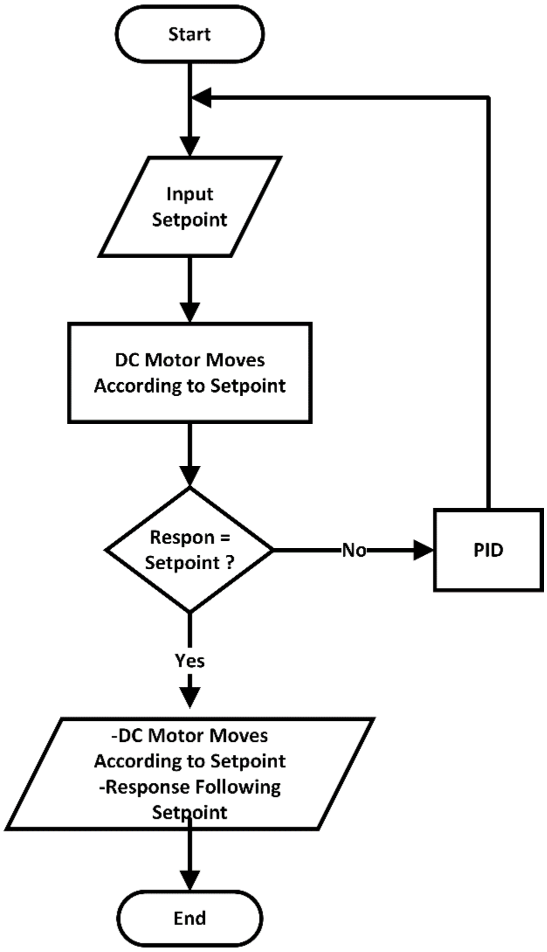

Fig. 5. System flow chart

III. RESULT AND DISCUSSION

A. *DC Motor Testing*

In testing the PG28 DC motor, it was carried out by giving a PWM speed value, where in this experiment using PWM 255 can be seen on Fig. 6. By using a tachometer where the motor test results are as expected on Fig. 7.

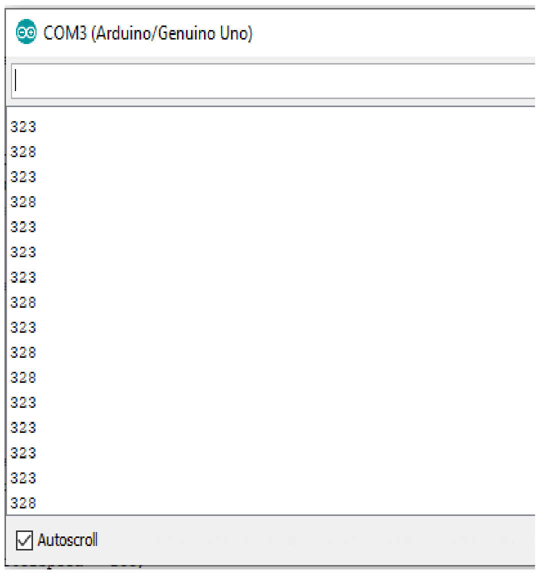

Fig. 6. Rpm 255 rpm Arduino

Fig. 7. Rpm 255 rpm tachometer

B. Overall System Testing

a) PID testing

The approval of the DC motor speed control design system using PID control where in this test by changing the values of Kp, Ki and Kd, the results can be seen in Fig. 8.

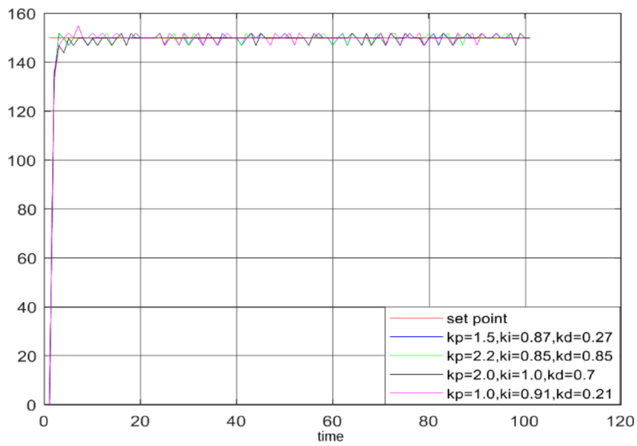

Fig. 8. Graphic control PID

It can be seen Fig. 8 of the comparison chart of PID values that change the value of Kp, Ki and Kd values obtained the best results as in Fig. 8 where the results obtained are as desired stably at the given set point value.

TABLE III. PID CONTROL

Kp	WHICH	KD	Rise Time (Tr)	Overshot (Mp)	Peak Time	Settling Time	Error
1.5	0.87	0.27	0.9925	1.3333	3	2.7368	0
2.2	0.85	0.85	0.8824	1.3333	16	4.5000	0
2.0	1.0	0.7	1.0301	3.3333	3.3333	7.4000	0
1.0	0.91	0.21	0.9925	1.3333	3	2.7368	0

It can be seen in Table III by changing the value of Kp, Kp and Kd value can be seen the results of the Rise Time which is small and there is no error.

b) Load PID Testing

The approval of the DC motor speed control design system using PID control where in this test by changing the values of Kp, Ki and Kd with a load, the results can be seen as follows.

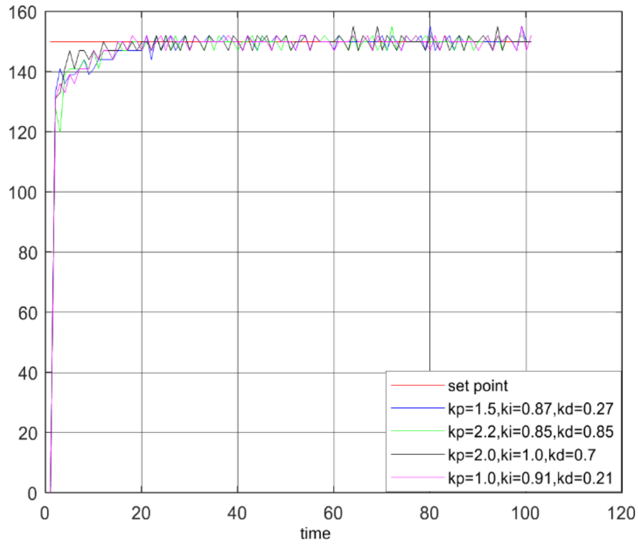

Fig. 9. Load PID control graph

It can be seen Fig. 9 of the comparison chart of PID values given the load by changing the values of Kp, Ki and Kd values obtained the best results as in Fig. 9. Hasil obtained as desired stably at the given set point value even though the resulting Rise Time result is slower than in PID which is not given a load.

TABLE IV. PID CONTROL

Kp	WHICH	KD	Rise Time (Tr)	Overshot (Mp)	Peak Time	Settling Time	Error
1.5	0.87	0.27	1.1372	3.3333	80	99.4000	0
2.2	0.85	0.85	2.6723	3.3333	72	72.2500	0
2.0	1.0	0.7	2.1355	3.3333	64	94.2500	0
1.0	0.91	0.21	.6855	3.3333	99	99.2500	-2

It can be seen in Table IV by changing the value of Kp, Ki and the value of Kd given the load can be seen the result of the Rise Time obtained is greater than the PID, larger overshot and small error.

IV. CONCLUSION

Based on the discussion and test results of the research controller of the angular velocity of the DC motor using the Arduini UNO-based PID method, conclusions were obtained.

For the work of the DC motor control tool system that has been designed and can be controlled so that it reaches the rotation that is in line with the reference value that has been given.

The correct PID value to control the speed of the DC motor rotation can be found by changing the values of KP, KI, and KD values using the trial and error method, in the trial and error process, the values used with $Kp = 1.5$, $Ki = 0.87$, $Kd = 0.27$ are obtained.

From the parameters that have been tested, the system response is obtained from the values of the Kp, Ki, Kd parameters that correspond to software testing where the value of Rise Time = 0.9925, settling time = 2.7368, overshoot = 1.3333 and steady state error = 0

ACKNOWLEDGMENT

The author would like to thank the editors and reviewers for their suggestions and input so that this journal can be resolved. To all parties involved in the research and preparation of this journal, the author also expresses his gratitude. May Allah swt repay the kindness of all parties involved and count as a worship charity

REFERENCES

[1] B. Hekimoğlu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *in IEEE Access,* vol. 7, pp. 38100-38114, 2019, https://doi.org/10.1109/ACCESS.2019.2905961.

[2] S. Khubalkar, A. Junghare, M. Aware, and S. Das, "Modeling and control of a permanent-magnet brushless DC motor drive using a fractional order proportional-integral-derivative controller," *Turkish Journal of Electrical Engineering and Computer Sciences,* vol. 25, no. 5, pp. 4223-4241, 2017, https://doi.org/10.3906/elk-1612-277.

[3] A. B. Satpe, S. V. Khobragade and S. L. Nalbalwar, "Wheelchair Control Using Hand Movement & Voice with Obstacle Avoidance," *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS),* pp. 1175-1178, 2018, https://doi.org/10.1109/ICCONS.2018.8663070.

[4] D. Handaya, and R. Fauziah, "Proportional-integral-derivative and linear quadratic regulator control of direct current motor position using multi-turn based on LabView," *Journal of Robotics and Control (JRC),* vol. 2, no. 4, pp. 332-336, 2021, https://doi.org/10.18196/jrc.24102.

[5] M. Alvarez, *et al.,* "Prototyping and control of an automatic ceramic tableware finishing device," *In APCA International Conference on Automatic Control and Soft Computing*, pp. 167-178, 2022, https://doi.org/10.1007/978-3-031-10047-5_15.

[6] S. H. Hussein, and M. A. A. Al-Rawe, "Real time speed control of the dc motor in labview," *In 2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*, pp. 137-141, 2020, https://doi.org/10.1109/ICEEE49618.2020.9102625.

[7] U. Pande and Y. V. Hote, "Real time Design of PID Controller for Process control system," *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 278-283, 2020, https://doi.org/10.1109/ICSTCEE49637.2020.9276776.

[8] S. Widodo, M. M. Amin, A. Sutrisman, E. Cofriyanti, and R. M. Puji, "Implementation of Parking Portal Door Security System Using RFID and Password Based on Microcontrollers in Sriwijaya State Polytechnic," *In Journal of Physics: Conference Series*, vol. 1500, no. 1, p. 012114, 2020, https://doi.org/10.1088/1742-6596/1500/1/012114.

[9] J. Lin, Z. Gong, Z. Chen, and Y. Huang, "Design of DC motor speed auto-tuning system based on LabVIEW," *In International Conference on Electronic Information Technology (EIT 2022)*, vol. 12254, pp. 105-112, 2022, https://doi.org/10.1117/12.2640289.

[10] S. Balamurugan and A. Umarani, "Study of Discrete PID Controller for DC Motor Speed Control Using MATLAB," *2020 International Conference on Computing and Information Technology (ICCIT-1441),* pp. 1-6, 2020, https://doi.org/10.1109/ICCIT-144147971.2020.9213780.

[11] G. Capasso, *et al.,* "SOXS control electronics design," *In Software and Cyberinfrastructure for Astronomy V*, vol. 10707, pp. 715-724, 2018, https://doi.org/10.1117/12.2312780.

[12] A. F. Rifa'i, H. Rudiansyah, Nuryanti and R. Milanto, "Design and Implementation of Auto-limit Riding Speed System for Electric Motorcycles Based on Battery Level with Fuzzy PID," *2022 7th International Conference on Electric Vehicular Technology (ICEVT)*, pp. 27-33, 2022, https://doi.org/10.1109/ICEVT55516.2022.9925008.

[13] H. Suryoatmojo, *et al.,* "Robust speed control of brushless dc motor based on adaptive neuro fuzzy inference system for electric motorcycle application," *International Journal of Innovative Computing Information and Control,* vol. 16, no. 2, pp. 415-428, 2020, https://doi.org/10.1016/j.asoc.2015.04.014.

[14] C. N. Khanh, *et al.,* "Novel Adaptive Control Method for BLCD Drive of Electric Bike for Vietnam Environment," *In IOP Conference Series: Materials Science and Engineering*, vol. 819, no. 1, p. 012017, 2020, https://doi.org/10.1088/1757-899X/819/1/012017.

[15] Y. B. Koca, Y. Aslan, and B. Gökçe, "Speed control based PID configuration of a DC motor for an unmanned agricultural vehicle," *In 2021 8th International Conference on Electrical and Electronics Engineering (ICEEE)*, pp. 117-120, 2021, https://doi.org/10.1109/ICEEE52452.2021.9415908.

[16] M. Shouran and M. Habil, "Tuning of PID Controller Using Different Optimization Algorithms for Industrial DC Motor," *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE),* pp. 756-759, 2021, https://doi.org/10.1109/ICACITE51222.2021.9404616.

[17] A. Alkrwy, A. A. Hussein, T. H. Atyia, and M. Khamees, "Adaptive tuning of PID controller using crow Search algorithm for DC motor," *In IOP Conference Series: Materials Science and Engineering*, vol. 1076, no. 1, p. 012001, https://doi.org/10.1088/1757-899X/1076/1/012001.

[18] A. Ma'arif, H. Nabila, and O. Wahyunggoro, "Application of intelligent search algorithms in proportional-integral-derivative control of direct-current motor system," I*n Journal of Physics: Conference Series*, vol. 1373, no. 1, p. 012039, 2021, https://doi.org/10.1088/1742-6596/1373/1/012039 .

[19] S. Tiwari, A. Bhatt, A. C. Unni, J. G. Singh and W. Ongsakul, "Control of DC Motor Using Genetic Algorithm Based PID Controller," 2018 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE), pp. 1-6, 2018, https://doi.org/10.23919/ICUE-GESD.2018.8635662.

[20] M. Ahmed, *et al.,* "Improved PID Controller for DC Motor Control," *In IOP Conference Series: Materials Science and Engineering*, vol. 1052, no. 1, p. 012058, 2021, https://doi.org/10.1088/1757-899X/1052/1/012058.