Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation

Alfian Ma'arif
Department of Electrical Engineering
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
https://orcid.org/0000-0002-3482-971X

Iswanto
Department of Electrical Engineering
Universitas Muhammadiyah Yogyakarta
Yogyakarta, Indonesia
http://orcid.org/0000-0001-8459-3920

Nia Maharani Raharja
Department of Electrical Engineering
Universitas Islam Negeri Sunan Kalijaga
Yogyakarta, Indonesia
nia.raharja@uin-suka.ac.id

Phisca Aditya Rosyady
Department of Electrical Engineering
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
ahmadradityac@ee.uad.ac.id

Ahmad Raditya Cahya Baswara
Department of Electrical Engineering
Universitas Ahmad Dahlan
Yogyakarta, Indonesia
ahmadradityac@ee.uad.ac.id

Aninditya Anggari Nuryono
Department of Electrical Engineering
Universitas Gadjah Mada
Yogyakarta, Indonesia
anindityanuryono@gmail.com

***Abstract*—The research proposes controlling DC motor angular speed using the Proportional Integral Derivative (PID) controller and hardware implementation using a microcontroller. The microcontroller device is Arduino Uno as data processing, the encoder sensor is to calculate the angular speed, and the motor driver is L298. Based on the hardware implementation, the proportional controller affects the rise time, overshoot, and steady-state error. The integral controller affects overshoot and undershoot. The derivative controller affects overshoot insignificantly. The best parameter PID is Kp=1, Ki=0.3, and Kd=0.1 with system response characteristic without overshoot and undershoot. Using various set point values, the controller can make the DC motor reach the reference signal. Thus, the PID controller can control, handle, and stabilize the DC motor system.**

***Index Terms*—*DC Motor*, *PID Controller*, *Arduino Uno*, *Proportional-Integral-Derivative*, *Linear Control*.**

I. INTRODUCTION

The Direct Current (DC) motor is the device that converts the electric force to be the mechanic force [1]. It has a lot of applications such as in robotics [2] [3] [4] and industrial automation. DC motor is very famous because it is easy to control, simple, and can give a good performance.

The problem of the DC motor is how to control and stabilize the angular speed in reference value. It needs a controller to control the voltage input. There are some controllers implemented in the system, such as PID Controllerand Fuzzy Logic Controller (FLC) [5]. Some researchers have researched modeling, control, simulation [6] [7]. However, the research is just about the simulation, not the hardware implementation. The simulation is different from hardware implementation. There is a lot of the troubleshooting in the hardware implementation. Thus, the hardware implementation is more complicated than simulation.

Based on the background, the research would propose controlling DC Motor using the PID controller. The research will use the hardware implementation using low-cost microcontroller Arduino Uno [8] [9] [10] [11]. The objective of the research is to observe the PID controller Implementation and educate about PID controller characteristics in hardware implementation.

The research structure is as follows. The first is the proposed system. The second is the methodology, the third is result and discussion, and the last is the conclusion.

II. PROPOSED SYSTEM

The diagram block system in Fig. 1. The motor driver is L298. The microcontroller is the Arduino Uno as a data processor. The DC motor type is JGA25-370, with an encoder sensor included. The DC motor has spesifications reaching 210RPM speed in 6volt voltage. There are two encoders in the DC motor, encoder A and encoder B. The encoder is used to calculate the angular speed. The serial monitor is the Laptop that is used as an angular speed interface.

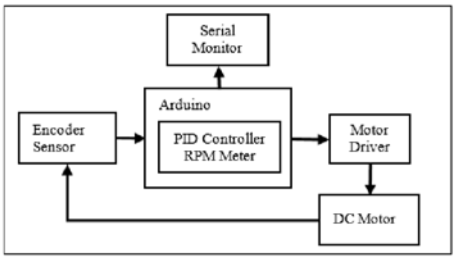

Fig. 1. The Block Diagram System

The control system block diagram is shown in Fig. 2. The Set Point (SP block) is the reference value that must be

978-1-7281-9702-9/20/$31.00 ©2020 IEEE

reached by DC Motor. The Proportional Integral Derivative (PID Block) controller must control the DC motor to reach the set point value.

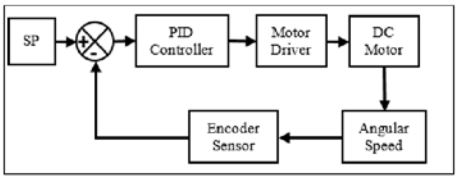

Fig. 2. The Control System Block Diagram

The output of PID is a control signal that is input for the Motor Driver. The motor driver's function is to step up the voltage. The feedback is the encoder sensor that used pulse to calculate the angular speed in RPM (Radian per Minute). The output of the PID controller is an 8-bit Pulse Width Modulation (PWM) signal with a value between 0-255.

III. Methodology

A. Wiring Diagram

The wiring diagram is shown in Fig. 3, and the Input-Output PIN is shown in Table I. There are three kinds of voltage generated from a power supply: 3.3volt, 5volt, and 12volt. The encoder sensor used the 3.3volt. The L298 motor driver used the 5V to activate the IC and used 12V to supply the DC motor. The PIN motor direction is arranged by PIN 7 and 8. The PWM PIN is arranged by PIN 8. The encoder data is connected with PIN 2 and 3.

TABLE I
Arduino Uno PIN Number

PIN NUMBER	I/O PIN FUNCTION	PIN NUMBER	I/O PIN FUNCTION
6	PWM PIN	2	Encoder A
7	Motor Driver Direction	3	Encoder B
8	Motor Driver Direction		

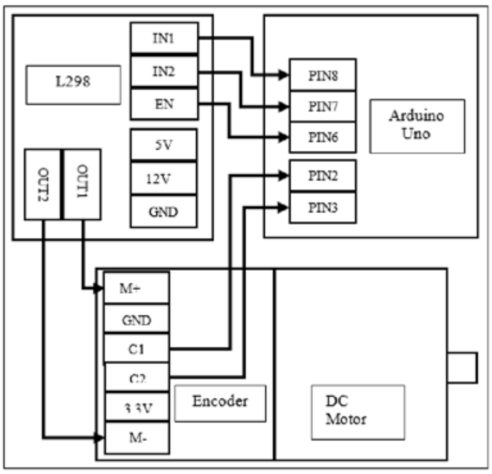

Fig. 3. The Wiring Diagram

B. Speed Meter

The code of the angular speed meter is shown in Listing. 1. The principle of angular speed meter is to count the pulse from the encoder then multiply the counted pulse by an encoder constant to get the RPM value. Based on the specifications, the DC Motor reaches the 210RPM in 6volt voltage. The encoder constant can be found based on the DC motor 6volt power voltage and the calculated pulses.

```
currentMillis = millis();
 if (currentMillis - previousMillis >
     interval)
 {
  rpm =
      (float)(encoderValue*ENCODER_CONSTANT);
  previousMillis = currentMillis;
  Serial.println(rpm); }
```

Listing 1. The Angular Speed Meter

C. Proportional Integral Derivative (PID) Controller

The proportional integral derivative (PID) controller is the most used controller in robotics and industrial. It is because the controller is easy to understand, easy to be implemented in simulation [12], and hardware implementation [13], and has a simple structure but could give an excellent response. The PID controller signal in time-domain [19] could be written as

$$u = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)dt + K_p T_d \frac{de(t)}{dt} \quad (1)$$

Or it can be written as

$$u = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (2)$$

Where

$$K_i = \frac{K_p}{T_i}, K_d = K_p T_d$$

Where the variable *Kp* is the proportional constant, the variable *Ki* is the integral constant, the variable *Kd* is the derivative constant. The PID code is shown in Listing. 2. The code is based on the PID equation in (2).

```
error = sp - rpm;
sum_error = sum_error + error;
motorSpeed = ((kp*error) + (ki*sum_error) +
    (kd *(error-last_error)));
if(motorSpeed > 255) motorSpeed = 255;
else if(motorSpeed < 0) motorSpeed = 0;
analogWrite(PWM, motorSpeed);
last_error = error;
```

Listing 2. The Proportional Integral Derivative Controller

IV. Result and Discussion

There is some examination in this section. The examinations are the open-loop test, controller parameter test, and tracking control test.

A. Open Loop Test

The open-loop test is shown in Fig. 4. The x-axis is time and the y-axis is angular speed. The open-loop test observes the system response (angular speed) in RPM using an encoder sensor with a constant input voltage in PWM.

Based on the result with various PWM, the speed meter could calculate the angular speed. The angular speed characteristics are rising initially and stable in specific RPM or steady for a long time. Thus, the angular speed meter can calculate the RPM, and the system has a stable system response.

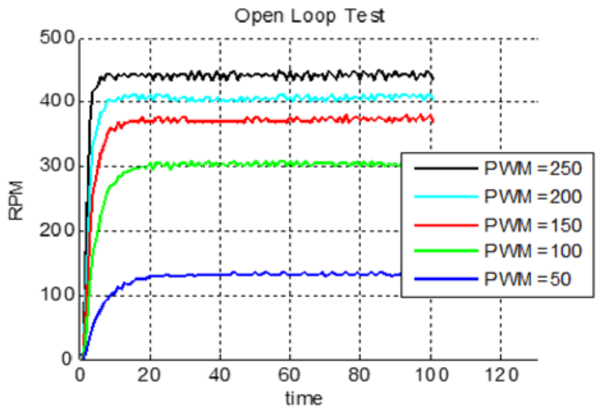

Fig. 4. Open Loop Test Result

The PWM, the voltage in DC Motor, and angular speed in RPM calculation are shown in Table II.

TABLE II
THE PWM, VOLTAGE AND RPM CALCULATION

PWM (8-bit)	VOLTAGE (volt)	RPM VALUE
50	3	127
100	6.7	297
150	8.2	368
200	9.1	405
250	9.8	440

Based on the examination, to get the desired RPM, the voltage must be adjusted. The method is not efficient because the calibration must be done while the RPM is changing. Because of that, the controller is needed to make the angular speed always reach the reference.

B. Proportional Integral Derivative Parameter Test

The proportional, Integral, and Derivative constants have different system response characteristics. It could be in rising time, settling time, overshoot, undershoot, peak time, and the steady-state error. The first test is to analyze the effect of the increased proportional value. The result is shown in Fig. 5. The system response is shown in Table III.

Based on Fig. 5 and Table III, it could be known that increased proportional value can decrease the rise time value, increase overshoot value, and decrease the steady-state error.

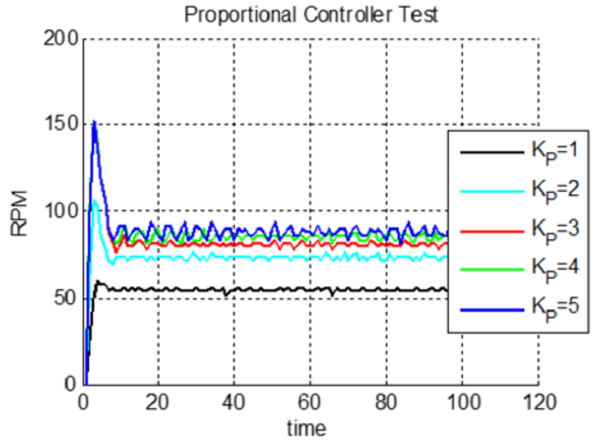

Fig. 5. The system's response of Proportional control

TABLE III
PROPORTIONAL CONTROL SYSTEM RESPONSE

Value	System Response			
	Rise Time	*Setting Time*	*Overshoot*	*Steady State Error*
Kp=1	-	-	-	46
Kp=2	1.4943	-	6	28
Kp=3	0.8333	-	50	18
Kp=4	0.8163	-	52	16
Kp=5	0.8511	-	52	14

The result does not have a settling time because the response system cannot reach the set point.

The second test is to analyze the effect of increased integral value. The result is shown in Fig. 6. The system response is in Table IV. It could be known that increased integral value can decrease the rise time value, increase overshoot value, and eliminate the steady-state error.

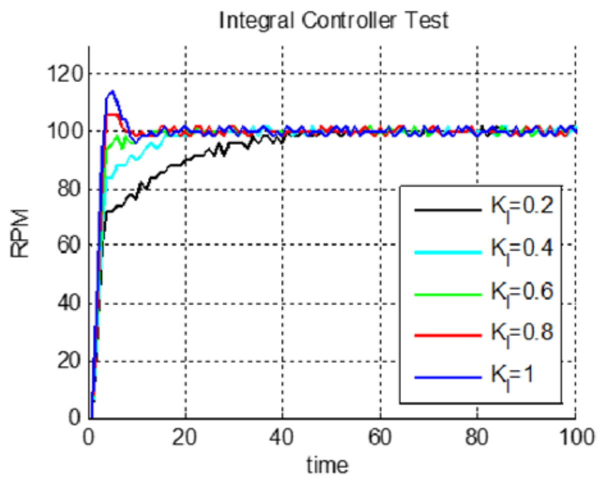

Fig. 6. The system's response of Integral control

The third is analyzing the effect of increased derivative value. The result is shown in Fig. 7. The system response is in Table V. Based on Fig. 7 and Table V, it could be known

TABLE IV
INTEGRAL CONTROL SYSTEM RESPONSE

Table Head	System Response			
	Rise Time	***Setting Time***	***Overshoot***	***Steady State Error***
Ki=0.1	18.6154	39.500	2	2
Ki=0.2	7.1667	15	2	0
Ki=0.3	2.4837	10	2	0
Ki=0.4	1.9949	6	6	2
Ki=0.5	1.7219	10	16	0

that increased derivative value can decrease the overshoot a little bit. The derivative controller does not affect the response system significantly, probably because of the use of less sample time.

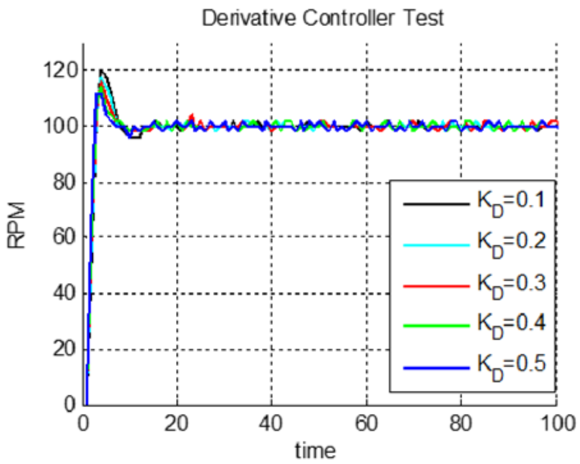

Fig. 7. The system's response of Derivative control

TABLE V
DERIVATIVE CONTROL SYSTEM RESPONSE

Value	System Response			
	Rise Time	***Setting Time***	***Overshoot***	***Steady State Error***
Kd=0.1	1.5417	11.5000	20.0000	0
Kd=0.2	1.5385	9.5000	18.0000	0
Kd=0.3	1.5014	22.3333	16.000	2
Kd=0.4	1.4676	6	14.000	2
Kd=0.5	1.3702	9.5000	12.000	2

Finally, the best proportional integral derivative parameter value is shown in Table VI. The result of the system response is shown in Fig. 8.

The setpoint as the reference value is 100RPM. The first best parameter gives the system response without overshoot, and the second-best parameter gives the system response with the little overshoot.

C. Tracking Control Test

The test of various setpoint is shown in Fig 9. The system response is shown in Table VII.

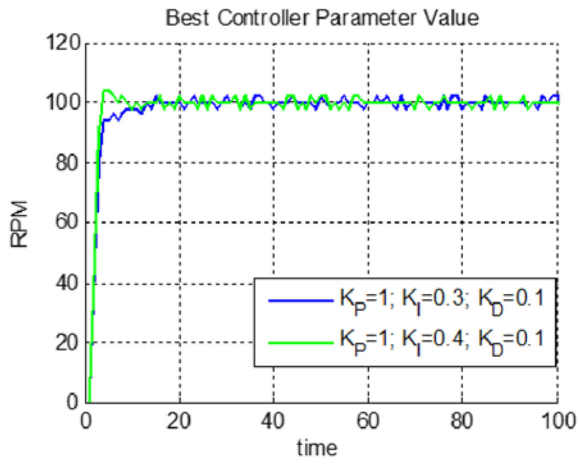

Fig. 8. The system's response of PID controller

TABLE VI
BEST PARAMETER PID

Value	System Response			
	Rise Time	***Setting Time***	***Overshoot***	***Steady State Error***
Kp=1 Ki=0.3 Kd=0.1	2.4511	11.5000	2.0	2
Kp=1 Ki=0.4 Kd=0.1	1.7500	5	4.0	2

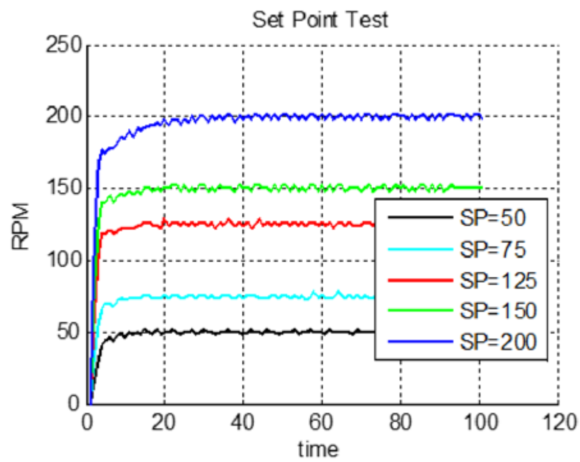

Fig. 9. The system's response to various set point

TABLE VII
SYSTEM RESPONSE TO VARIOUS SET POINT

Set point	System Response			
	Rise Time	***Setting Time***	***Overshoot***	***Steady State Error***
SP=50	3.4000	100.5200	8.3333	52
SP=75	3.3077	100.2400	2.6316	24
SP=125	2.2949	99.7600	3.2258	24
SP=150	2.4602	14.2500	1.3333	50
SP=200	5.8394	100.0100	2.0202	98

Based on the result, the controller could reach the reference value. The manual tuning uses 100 RPM as setpoint, thus the best response system with fast rise time only in 100 RPM or below the set point. Overall, the controller can stabilize the DC motor in the various set point below the 20 of time. So the controller is useful to control, stabilize the DC motor system.

Compared to the open-loop test, it will be tough to get specific RPM using voltage adjustment. Also, it will need more time and effort. Thus, the PID controller can handle the system to get efficient control of various references.

V. Conclusion and Future Work

The research is about controlling the DC motor system using a Proportional Integral Derivative controller. The hardware implementation is held by using Arduino Uno. Based on the result, the PID Controller can control, handle, and stabilize the DC motor system at various setpoints. The best parameter of PID is $Kp = 1$, $Ki = 0.3$, $Kd = 0.1$ with a characteristic response without overshoot. The other best parameter of PID is $Kp = 1$, $Ki = 0.4$, $Kd = 0.1$ with a little bit overshoot.

The tuning of the PID controller parameter is still using the trial and error. It needs further research about the autotuning of the PID controller or using other tuning methods such as the Ziegler Nichols, Genetic Algorithm, or Coefficient Diagram Method. The angular speed meter still has much error when calculating the RPM. Thus, it needs further research about the best angular speed meter method, such as using the current sensor, voltage sensor, two-pulse method, and etc.

Acknowledgment

The author would like to thank the editor that helps article processing and editing until the article is published. Also, the author would like to thank the reviewer that give advice, comment, and suggestion to improve the quality of the paper.

References

[1] 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.

[2] A. Latif, K. Shankar, and P. T. Nguyen, "Legged Fire Fighter Robot Movement Using PID," Journal of Robotics and Control (JRC), vol. 1, no. 1, pp. 15-–18, 2020.

[3] A. Latif, H. A. Widodo, R. Rahim, and K. Kunal, "Implementation of Line Follower Robot based Microcontroller ATMega32A," Journal of Robotics and Control (JRC), vol. 1, no. 2, pp. 70—74, 2020.

[4] N. Rinanto, I. Marzuqi, A. Khumaidi, and S. T. Sarena, "Obstacle Avoidance using Fuzzy Logic Controller on Wheeled Soccer Robot," Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, vol. 5, no. 1, pp. 26—35, 2019.

[5] Z. Tir, O. Malik, M. A. Hamida, H. Cherif, Y. Bekakra, and A. Kadrine, "Implementation of a fuzzy logic speed controller for a permanent magnet dc motor using a low-cost Arduino platform," in 2017 5th International Conference on Electrical Engineering - Boumerdes, ICEE-B 2017, 2017, vol. 2017–January, pp. 1—4.

[6] A. Ghareaghaji, "A Comparison between Fuzzy–PSO Controller and PID–PSO Controller for Controlling a DC Motor," Bulletin of Electrical Engineering and Informatics, vol. 4, no. 2, pp. 130—135, 2015.

[7] R. Akbari-Hasanjani, S. Javadi, and R. Sabbaghi–Nadooshan, "DC motor speed control by self–tuning fuzzy PID algorithm," Transactions of the Institute of Measurement and Control, vol. 37, no. 2, pp. 164-–176, 2015.

[8] I. Prasojo, A. Maseleno, O. Tanane, and N. Shahu, "Design of Automatic Watering System Based on Arduino," Journal of Robotics and Control (JRC), vol. 1, no. 2, pp. 59-–63, Jan. 2020.

[9] M. W. Hariyanto, A. H. Hendrawan, and R. Ritzkal, "Monitoring the Environmental Temperature of the Arduino Assistance Engineering Faculty Using Telegram," Journal of Robotics and Control (JRC), vol. 1, no. 3, pp. 96—101, 2020.

[10] N. H. Wijaya, D. F. Novela, N. Shahu, and M. U. Sattar, "Arduino–based Mini Shaker for Automatic Chemical Solution Mixer," Journal of Robotics and Control (JRC), vol. 1, no. 6, pp. 220-–223, 2020.

[11] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID System Regulator for Mini Conveyor Drive Based–on Matlab," Journal of Robotics and Control (JRC), vol. 1, no. 6, pp. 185-–190, 2020.

[12] 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.

[13] A. Febriawan and W. S. Aji, "Rotating Control on Robots Indonesian Abu Robot Contest with PID and IMUBNO055 Controls," Buletin Ilmiah Sarjana Teknik Elektro, vol. 2, no. 1, pp. 14—23, 2020.

[14] K. Ogata, Modern Control Engineering. Prentice Hall, 2010.