ISSN 2355-3286

Modeling and Controlling the Actuator Joint Angle Position on the Robot Arm Base Using Discrete PID Algorithm

Machdiar Rohman[1], Dede Irawan Saputra[2]
Universitas Jenderal Achmad Yani, Program Studi Teknik Elektro
Cimahi, Indonesia
[1]machdiarrohman@gmail.com, [2]dedeirawan.saputra@lecture.unjani.ac.id

Accepted 09 June 2021
Approved 22 September 2021

***Abstract*— DC motor is a motor that is easy to apply. Its application in robotic DC motor control often occurs errors due to the existing load, so that the DC motor becomes inaccurate. The control used is PID (proportional integral differential). This PID control system works by processing calculations based on the control variables Kp, Ki, and Kd to achieve the conditions according to the expected setpoint. To make a DC motor position control device can be controlled with a PID controller. In practice, the variable to be controlled in this research is position control in the form of degrees. With the Arduino Mega controller, the motor driver as a DC motor rotation controller, the DC motor is given feedback in the form of an encoder sensor, the software used is the Arduino IDE. The results showed that PID control can correct errors and transient responses with a time constant value of 1.50 seconds, a rise time of 1.60 seconds, a settling time of 2.30 seconds and a delay time of 1.20 seconds and a peak time of 1.6 seconds and an error value of 0.33% through tuning parameter Kp = 16 Ki = 0.001 Kd = 16.**

***Index Terms*— Motor DC, Discrete PID, System response**

I. Introduction

DC motor is a type of drive that is often used in all fields, one of which is the field of robotics and control [1]. DC motors in the field of robotics can be controlled to regulate speed [2], position or a combination of them [3]. DC motor has a fast response, but still has a steady state error [4]. To reduce steady state error value of DC motor as position control, a controller is needed. The PID controller is the most widely used because it is simple and easy to learn and tuning its parameters. More than 95% of processes in the industry use these controllers. This controller is a combination of proportional (P), integral (I), and derivative (D) controllers [5], [6].

In most speed control systems, what is desired is to keep the rotational speed constant for all load conditions, not set the rotational speed to vary over time as the reference input changes. Not so with the positioning system. In these systems, system accuracy is usually measured not only by steady-state error for step inputs, but also by steady-state error for time-varying inputs. This is usually termed as following error. The addition of a PID controller can actually improve performance of positioning system in terms of magnitude of this following error [7].

As for some control positions for application of DC motors in the field of robotics, namely controlling position of motor on the goods moving robot with input in form of images that are read by camera [8]. Then, control the PID Training Kit ELABO TS 3400 Using a Position Sensor which is implemented on a DC motor whose control uses a potentiometer [9] Next, the use of a DC motor on an arm flip folding machine that uses PID control [10]. In this study, the application of a DC motor to the basic motion of the robot arm will be carried out to determine the position of the robotic base arm using the discrete PID method by providing input via a serial monitor.

The design of base arm robot is carried out by using actuator base arm robot in form of a brushed DC motor to achieve the angular position of encoder sensor [11] with motor control in form of the PID method [12]. There are several approaches that can be implemented, one of which is discrete PID [13]. The design of the PID controller in discrete domain is expected to approach the application of using a microcontroller and is expected to be able to increase transition response and reduce the value of base movement error with a predetermined constant on the PID controller. The determination of initial design carried out by transfer function modeling process using identification system of open loop and close loop approaches [14] with System Identification Matlab tools [15].

II. Design System

A. Movement Mechanism Design

At this point components that used in the robotic arm are computer controller which functions is to determine the input angle which inputted through the serial monitor, then there is the Arduino Mega microcontroller which will give command, then DC motor functions as an actuator that will move the robot arm and the last is in the form of a controller. PID controller to determine the final position of the robot arm as shown in the open loop block diagram in Fig. 1. Then in Fig. 2. is a block diagram of a close loop system where the system has a hall sensor that functions as a feedback sensor that can make motor determine rotation direction and angle of the motor.

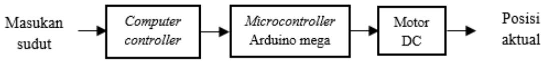


Fig. 1. Open loop arm robot system

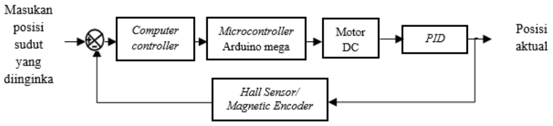

Masukan posisi sudut yang diinginka
Computer controller
Microcontroller Arduino mega
Motor DC
PID
Posisi aktual
Hall Sensor/ Magnetic Encoder

Fig. 2. Close loop arm robot system using PID controller

The schematic diagram of robotic arm shown in Fig. 3. consists of a robotic arm base, Arduino mega, encoded DC motor, L298N H-Bridge module driver, buck converter, and 12V power supply. The power supply is used as a voltage source for motor driver and buck converter. Buck converter convert 12V voltage source into 5V voltage. The 5V voltage source is used for the encoder voltage source from the encoder contained in the DC motor. The motor driver will provide a voltage source to DC motor and input pin of the motor driver will be connected to PWM pin on Arduino Mega, and output pin of the motor driver will be connected to DC motor. Furthermore, motor encoder contained in the DC motor will be connected to pins 2 and 3 of the Arduino as an interrupt pin.

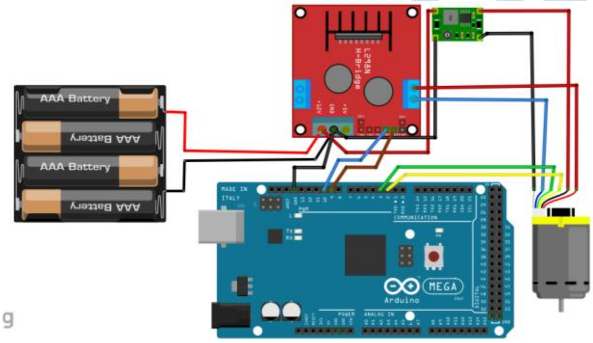


Fig. 3. Schematic diagram base arm robot

First base arm robot will initialize the DC motor, then angle value is entered on serial monitor. After angle value is entered, the robot is ready to use. Furthermore, the robot that has received angle input value, will be applied to the PID controller constant and the input will be calculated and adjusted to the input on serial monitor that has been inputted previously. Then the robot will move with a calculated angle and the movement process ends the flow process of the robotic arm system as shown in Fig. 4.

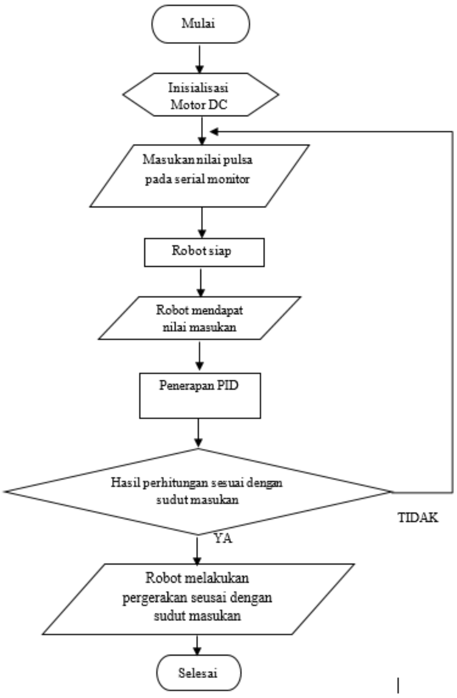

Mulai
Inisialisasi Motor DC
Masukan nilai pulsa pada serial monitor
Robot siap
Robot mendapat nilai masukan
Penerapan PID
Hasil perhitungan sesuai dengan sudut masukan
TIDAK
YA
Robot melakukan pergerakan seusai dengan sudut masukan
Selesai

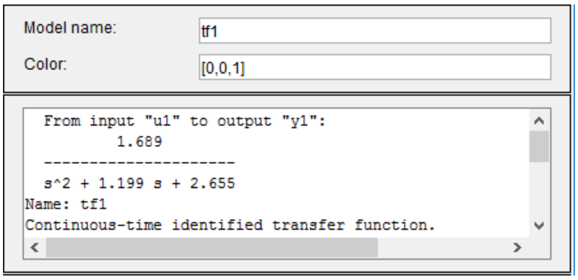
Fig. 4. Flowchart of arm robot control system using PID controller

B. *Software Design*

The software design is carried out using a system identification toolbox to create a mathematical model in the form of a transfer function from designed base arm robot and simulink is used to simulate the control system with PID algorithm.

1. *System identification toolbox* (SIT)

SIT is used to determine a mathematical model in form of a transfer function of DC motor actuator contained in the robot's base arm. The transfer function equation will be used to simulate encoder position control system with PID algorithm on simulink. Fig. 5. shows the transfer function generated by SIT.



Model name: tf1

Color: [0,0,1]

```
  From input "u1" to output "y1":
          1.689
  ---------------------
  s^2 + 1.199 s + 2.655
Name: tf1
Continuous-time identified transfer function.
```

Fig. 5. Transfer function parameter result from modeling

The transfer function generated by SIT in continuous domain is a transfer function in the form of a close loop. The system is converted into an open loop system as shown in equation 1 so that it can be modeled in discrete form using the c2d function in Matlab with a sample time of 0.1 seconds as shown in equation 2.

$$G(s) = \frac{1{,}689}{s^2+1{,}199s} \qquad (1)$$

$$G(z) = \frac{0{,}007912z + 0.007404}{z^2 + 1{,}82z + 0{,}8195} \qquad (2)$$

2. *Simulink*

Simulation on the MATLAB simulink software is carried out to generate data that will be used as a comparison with implementation of the designed base arm robot. There are 4 simulations used, namely close loop control simulations in discrete and continuous domains and PID close loop control simulations in discrete and continuous domains as shown in Fig. 6 to Fig. 9.

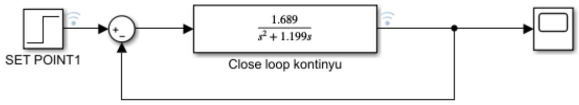


Fig. 6. Continuous domain close loop control simulation

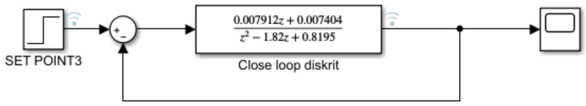


Fig. 7. Discrete domain close loop control simulation

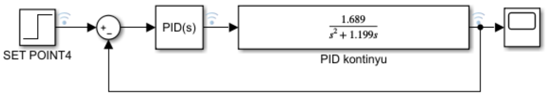


Fig. 8. Continuous PID close loop control simulation

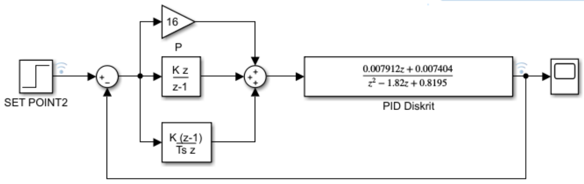


Fig. 9. Discrete PID close loop control simulation

III. Result and Discussion

The robotic base arm is made as shown in Fig. 10. The manufacture of this robotic base arm is intended to run three main systems, namely the open loop control system for base arm robot, the close loop control system for base arm robot, and the close loop control system for PID base arm robot.

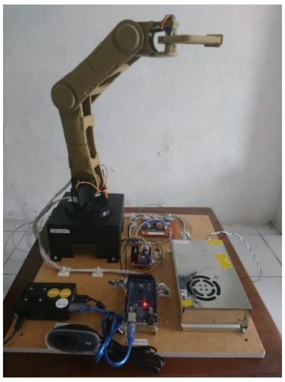


Fig. 10. Design result of the robotic base arm

In this process an effort is made to improve the system performance of the base arm robot. The following are results of trials and comparisons that have been carried out on each system.

A. *Open Loop Control System Base Arm Robot*

Testing response of open loop position control on the base arm robot is carried out to determine enhancement pulses response generated by encoder to be converted into angles when given a PWM signal as input. Data from sampling result of open loop control response test on base arm robot, is processed using system identification toolbox (SIT) and produces a mathematical model with transfer function in equation 1.

After obtaining transfer function equation from modeling process in system identification toolbox (SIT), an open loop position control simulation was performed on MATLAB simulink. The results of position control simulation can be seen in Fig. 11 and actual condition of open loop control is shown in Fig. 12.

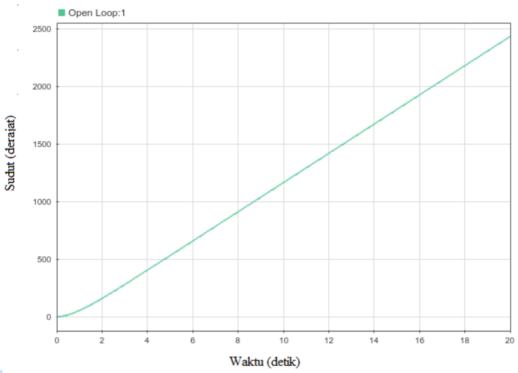


Fig. 11. Angle response graph of open loop position control simulation

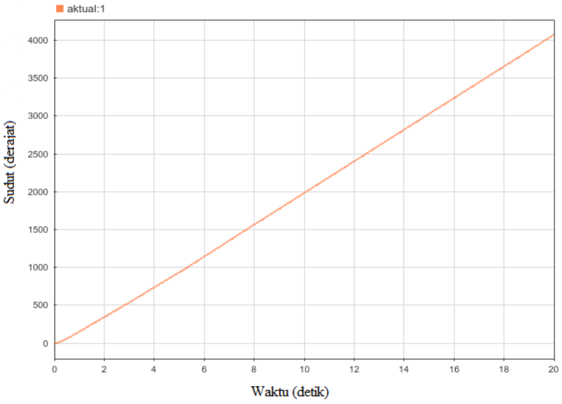


Fig. 12. Actual angle response graph of open loop position control

Based on Fig. 11 and Fig. 12 test results, it was found that response of open loop position control base arm robot from increasing angle of encoder could not reach steady state. With these results, the plant model used is assumed to be changed to a close loop condition to achieve steady state conditions.

B. Close Loop Control System Base Arm Robot

Closed loop position control system is made after system response from open loop position control system cannot reach a steady state. Testing of close loop position control system is carried out with the aim of knowing the characteristics of close loop response, the test is carried out by comparing real system response with simulation response in continuous and discrete domains performed on Simulink. In close loop experiment, motor angle position was tested by setting a setpoint value of 90º. Sampling time used in simulation setpoint is 0.1 seconds as shown in Fig. 13, Fig. 14, and Fig. 15.

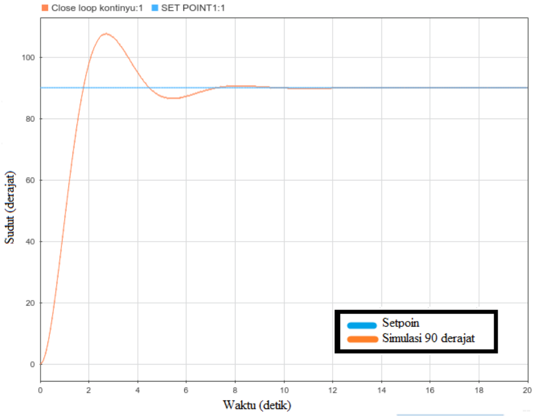


Fig. 13. Response of 90º continuous close loop control simulation

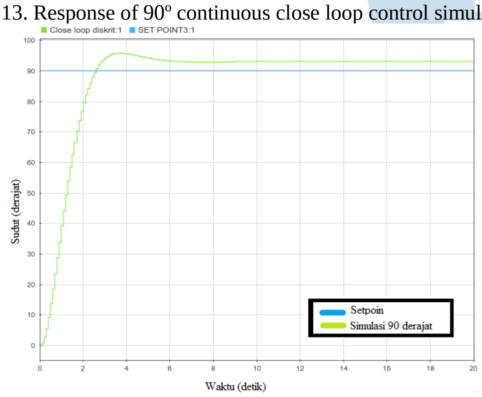


Fig. 14. Response of 90º discrete close loop control simulation

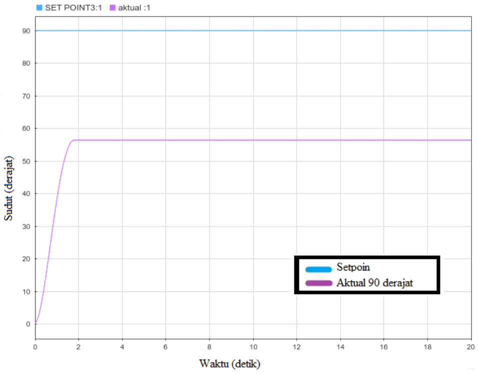


Fig. 15. Actual response of 90º close loop control

Responses that shown from continuous and discrete close loop simulation reaches desired setpoint. However, actual response of motor shows that rotation angle of the motor is in a steady state and cannot reach 90º setpoint

C. Close Loop PID Control System Base Arm Robot

Testing response of PID close loop position control is done by comparing data from simulation test results with data from implementation test results. As in close loop experiment, PID close loop experiment was carried out by setting motor angle position setpoint value of 90º. Sampling time used in simulation is 0.1 seconds.

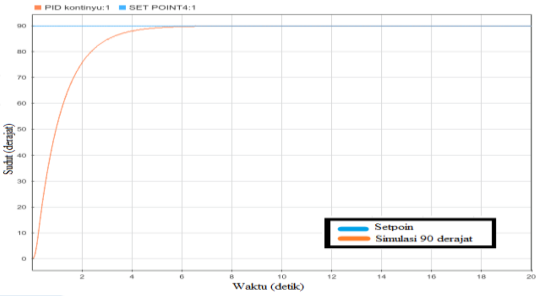


Fig. 16. Response of 90º continuous close loop PID control simulation

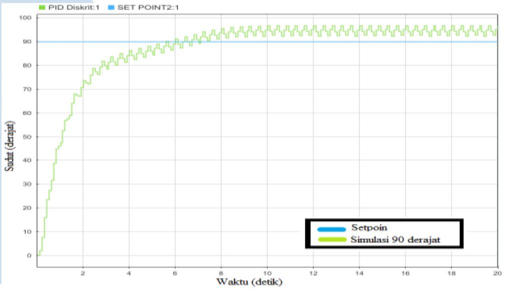


Fig. 17. Response of 90º discrete close loop control simulation

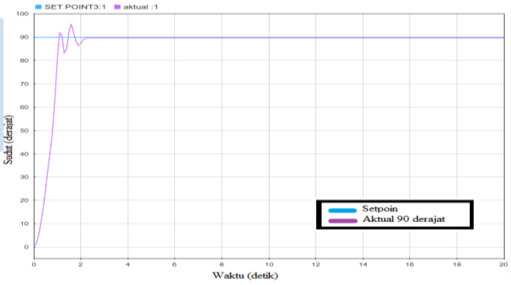


Fig. 18. Actual response of 90º close loop PID control

Responses that shown from Fig. 16 to Fig. 18 reaches a steady state at specified setpoint, which is an angle of 90°.

D. Comparison of system

Comparison of actual transient response, close loop and close loop PID at 90º angle from the system was carried out.

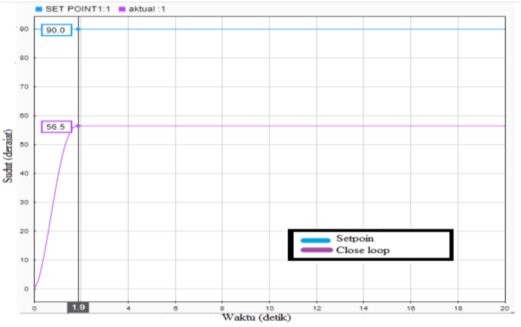


Fig. 19. Close loop position control transient response graph

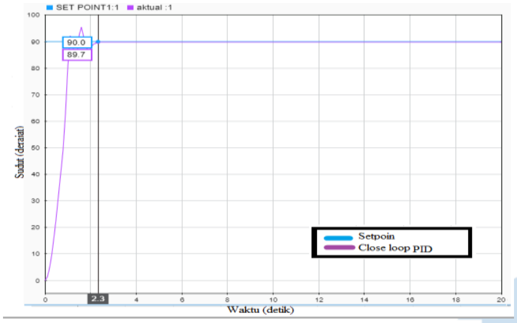


Fig.

Fig. 20. Close loop PID position control transient response graph

Transient response of close loop control system shown in Fig. 19 has an error value between actual condition and setpoint of 33.5º and reaches a steady state condition at 1.90 seconds with a value of 56.5º. While transient response of close loop PID control system shown in Fig. 20. has an error value between actual condition and setpoint of 0.3º and reaches a steady state condition at 2.30 seconds with a value of 89.7º. Comparison error value response characteristics of close loop position control algorithm and PID close loop position control can be seen in Table 1 below.

TABLE 1. RESPONSE COMPARISON OF *CLOSE LOOP* AND *CLOSE LOOP* PID

	Error Value	*Steady state*	*Steady state Value*
Close loop	33,5º	1,90 second	56,5º
Close loop PID	0,3º	2,30 second	89,7º

Transient response characteristics that can be measured include:

a. Time constant, time required for motor to reach 63.2% of maximum angle.
b. Rise time, time required for motor has increased by 10% - 90% at steady state.
c. Settling time, time required for motor to enter stable area as big as 5% of steady state.
d. Delay time, time required to reach 50% of steady state.
e. Peak time, time required to reach resulting peak value.

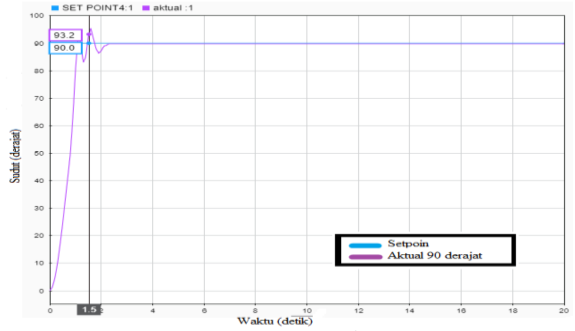


Fig. 21. Time constant implementation of PID close loop position control

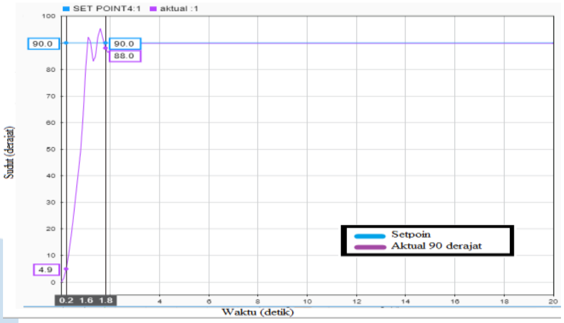


Fig. 22. Rise time implementation of PID close loop position control

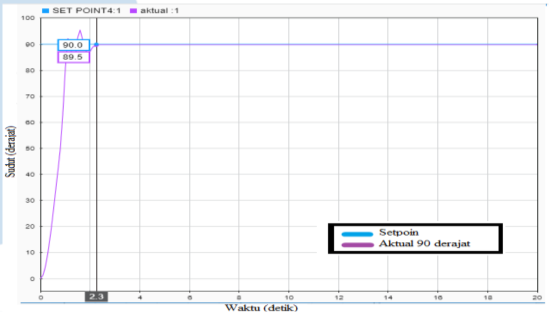


Fig. 23. Settling time implementation of PID close loop position control

Based on Fig. 21, Fig. 22 and Fig. 23, characteristics of transient response in the implementation of PID close loop position control have a time constant of 1.5 seconds, a rise time of 1.6 seconds, and a settling time of 2.3 seconds.

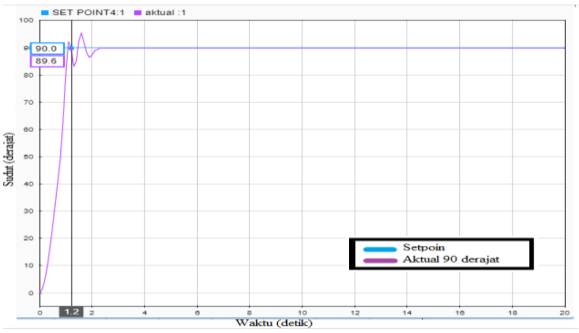


Fig. 24. Delay time implementation of PID close loop position control

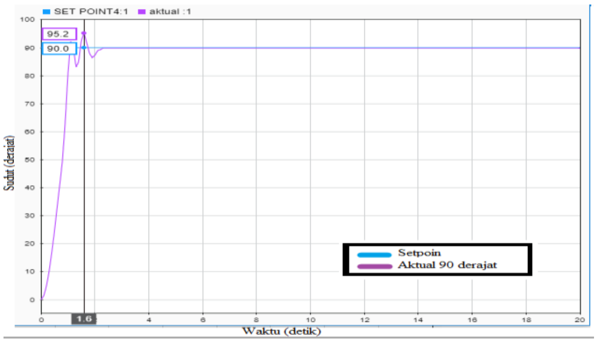


Fig. 25. Peak time implementation of PID close loop position control

Characteristics of transient response in the implementation of PID close loop position control shown in Fig. 24 and Fig. 25 have a delay time value of 1.2 seconds and a peak time of 1.6 seconds. Overall transient response resulting from the system is shown in Table 2.

TABLE 2. CHARACTERISTICS OF TRANSIENT RESPONSE *CLOSE LOOP* PID POSITION CONTROL

	Time constan	***Rise time***	***Settling time***	***Delay time***	***Peak time***
Actual	1,5 second	1,6 second	2,3 second	1,2 second	1,6 second

Then the last test was carried out in form of testing the response to difference setpoint value on PID close loop control with results that can be seen in Fig. 26.

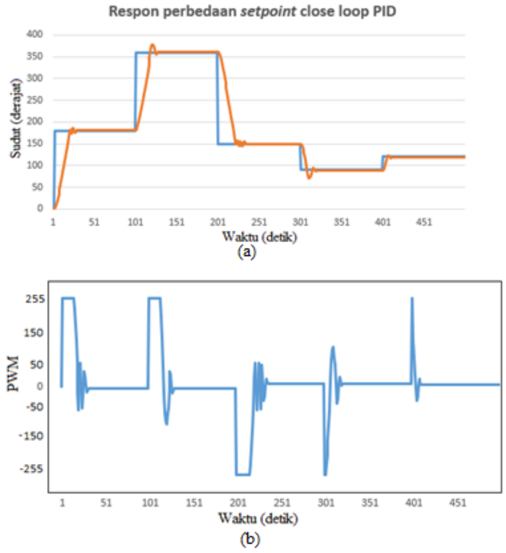


Fig. 26. Angle output response test graph (a) and PWM output graph (b) PID close loop control setpoint value

Angle control response in each setpoint increase experiment shown in Fig. 26, it can be seen that when the setpoint is at a value of 180°, 360° and 120° the response to an increase in angle tends to occur small and stable oscillations which are marked by an increase in the PWM value towards 255 but when the setpoint is at the value of 150° and 90°, the response to an increase in angle begins to slow down and oscillations tend to be large and are clearly visible when the setpoint is at 90°. The PWM output is able to adjust the value with a nominal value of 255 to -255 in order to maintain stability of output angle with the setpoint specified in PID close loop control.

IV. CONCLUSION

Based on design, measurement, and analysis. It can be concluded that the PID parameter implemented in the system transfer function has a Kp value of 16, a Ki of 0.001, and a Kd of 16 obtained through PID tuner in Matlab. The results of comparative analysis from close loop control and PID close loop control show that the PID close loop control is better than the close loop control, because it is able to optimally control the angle position which has an error value of 0.33% with a value of 0.3°. Close loop PID control has a transient response with a time constant of 1.50 seconds, a rise time of 1.60 seconds, a settling time of 2.3 seconds and a delay time of 1.2 seconds and a peak time of 1.6 seconds and for development of this research, it is recommended that the controller be developed by adding other controllers such as an anti-windup controller to reduce interference with the system. The use of Zigler-Nichole method in determining the PID parameters mathematically in future research. Addition of an upper arm robot that is integrated with the base arm robot.

ACKNOWLEDGEMENT

Thanks to Mr. Dede Irawan Saputra, S.Pd., M.T. who has guided the process of carrying out this research.

REFERENCES

[1] M. Ahmed, N. M. Tahir, A. Y. Zimit, M. Idi, K. A. Abubakar, and S. A. Jalo, “Improved PID Controller for DC Motor Control,” in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1052, no. 1, p. 012058, doi: 10.1088/1757-899x/1052/1/012058.

[2] A. Rafi, A. Tahtawi, and S. W. Jadmiko, “Kendali Kecepatan Motor DC Berbasis WNCS Menggunakan Pengendali PI Anti-Windup,” *JTERA*, vol. 4, no. 2, pp. 227–236, 2019, doi: 10.31544/jtera.v4.i2.2019.227-236.

[3] A. Rafi, A. Tahtawi, J. T. Elektro, P. N. Bandung, and K. B. Barat, “Kendali Posisi Motor DC Menggunakan Logika Fuzzy Interval Tipe 2 The Position Controlling of DC Motor Using Interval Type-2 Fuzzy Logic,” *TELKA*, vol. 7, no. 1, pp. 1–10, 2021.

[4] K. Sharma and D. K. Palwalia, “A modified PID control with adaptive fuzzy controller applied to DC motor,” in *IEEE International Conference on Information, Communication, Instrumentation and Control, ICICIC 2017*, 2018, no. 1, pp. 1–6, doi: 10.1109/ICOMICON.2017.8279151.

[5] T.-C. C. Nichitela and M.-G. G. Unguritu, “DC Motor Speed Control Using a Discrete PID Control Law,” *Ann. Univ. Craiova. Ser. Comput. Electron. Mechatronics*, vol. 14, no. 1, pp. 31–36, 2017.

[6] A. P. Singh, U. Narayan, and A. Verma, “Speed Control of DC Motor using Pid Controller Based on Matlab,” in *Innovative Systems Design and Engineering*, 2013, vol. 4, no. 6, pp. 22–28, [Online]. Available: http://www.iiste.org/Journals/index.php/ISDE/article/view/6018.

[7] S. Yahya, S. W. Jadmiko, K. Wijayanto, and A. R. A.

Tahtawi, “Design and implementation of training module for control liquid level on tank using PID method based PLC,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 830, no. 3, 2020, doi: 10.1088/1757-899X/830/3/032065.

[8] N. F. Zulfardi, D. I. Saputra, and A. D. Ahkam, “Aplikasi Deteksi Benda Menggunakan Metode Image Substraction Sebagai Masukan Koordinat Pada Robot Lengan 3 DOF,” in *Seminar Nasional Teknologi dan Riset Terapan*, 2019, no. September, pp. 30–37.

[9] A. Ningsih and C. Puspita, “Kendali PID Training Kit ELABO TS 3400 Menggunakan Sensor Posisi,” *Techné J. Ilm. Elektrotek.*, vol. 15, no. 1, pp. 9–16, 2016.

[10] W. S. Pambudi, T. Suheta, J. Muhammad, J. T. Elektro, I. Teknologi, and A. Tama, “Aplikasi Sistem Kontrol Sudut Dengan Metode Pid Pada Arm Flip Folding Machine Menggunakan Lego Mindstorm Ev3 Aplikasi Sistem Kontrol Sudut Dengan Metode Pid Pada Arm Flip Folding Machine Menggunakan Lego Mindstorm Ev3,” in *Seminar Nasional Sains dan Teknologi Terapan*, 2017, no. October.

[11] B. Joshi and R. Shrestha, “Modeling , Simulation and Implementation of Brushed DC Motor Speed Control Using Optical Incremental Encoder Feedback Modeling , Simulation and Implementation of Brushed DC Motor Speed Control Using Optical Incremental Encoder Feedback,” in *Proceedings of IOE Graduate Conference*, 2015, no. October 2014.

[12] S. J. Hammoodi, K. S. Flayyih, and A. R. Hamad, “Design and implementation speed control system of DC Motor based on PID control and Matlab Simulink,” *IJPEDS*, vol. 11, no. 1, pp. 127–134, 2020, doi: 10.11591/ijpeds.v11.i1.pp127-134.

[13] I. A. El-, “Design of discrete-time PID controller,” in *CEIT14*, 2014, pp. 110–115.

[14] J. Chotai, “Modelling and Position control of Brushed DC motor,” 2017.

[15] D. I. Saputra, C. Yusuf, and Z. Fakhri, “Optimasi Kendali Pada Sistem Node Nirkabel Berbasis Algoritma Fuzzy Logic Controller - PI,” in *SEMNASTERA (Seminar Nasional Teknologi dan Riset Terapan)*, 2020, pp. 49–57.