JTEV (Jurnal Teknik Elektro dan Vokasional)
http://ejournal.unp.ac.id/index.php/jtev

Vol. 10 No. 1 (2024)
E-ISSN: 2302-3309
P-ISSN: 2746-6086

Utilization of PWM Signals in Automatic Parking Gate Speed Control Using PID System Modeling

Ario Gustama, Ridwan Satrio Hadikusuma, Rahmat Hidayat
[1,2,3]Universitas Singaperbangsa Karawang
ariogustama008@gmail.com, ridwansatrio2@gmail.com, rahmathidayatt377@gmail.com

Abstract
Utilization of signals (Pulse Width Modulation) on the l298n motor driver and PID controller (Proportional Integral Derivative) is often used in industries, especially those that require robots as their automation medium, so that a process in robot automation does not only design system modeling, but is also required to implement a control system. angular position on a DC motor using the PID model. The PID control system in this study utilizes speed control with a PWM signal on the l298n motor driver and is applied as a DC motor angular position control module which is used for the parking doorstop system. The use of trial error method is a suitable choice value to be applied in finding PID position control. This system utilizes the PWM input signal to maintain the speed and position to match the set point when given a load and the ability of the system to catch up with speed and position to reach the set point when the motor starts running is a very important factor because it is a measure of speed and position performance. because of its effectiveness, in the application of automatic doors produces a standard configuration of the PID controller with parameters Kp, Ki and Kd which can be determined by the factory characteristics resulting in an effective design. In the trial error method, the most effective value was obtained with Kp = 10, Ki, 1.9 and Kd = 0.8 which were in a steady state and there was no overshoot.

***Keywords**: PID control, Parking bar, PWM signal, DC Motor, Set point*

INTRODUCTION

The need for a parking space is an urgent need at this time along with the volume of vehicles, especially in densely populated urban areas. based on data from the Central Bureau of Statistics, the volume of motorized vehicles in 2020 is 115.023.039 and of course it will continue to increase every year [1]. If the arrangement of the parking area is not managed properly, worrying about vehicles causing congestion and parking discipline cannot be avoided. Automatic parking barrier systems that still use DC motors without using system modeling that are used continuously for a long time will usually result in a less than optimal motor drive response. This of course can make the DC motor hot and the performance of the doorstop will be increasingly ineffective [2]. Therefore, it is necessary to optimize the control of the parking barrier motor in order to get an optimal response.

Motor control is needed to get the movement of the motor speed or motor position to match the predetermined value in order to avoid the accumulation of vehicles when they want to enter or leave the parking area due to the slow pace of the automatic parking gate. To be able to fully control a DC motor, it is necessary to control the speed and direction of rotation by utilizing the PWM (Pulse Width Modulation) technique for speed control. The ability of the system to maintain speed and position to match the set point under load and the ability of the system to catch up with speed and position to reach the set point when the motor starts running are very important factors as measures of speed and position control performance. How to control the motor by providing a control signal. In general, motor control is designed using a PID controller [3-5]. Previously there was research related to "Designing an Automatic Parking Door System With Arduino and RFID" by Yohana Tri Utami and Yuri Rahmanto, in this study it was found that the doorstop could only open automatically after scanning for RFID, there was also research related to "Controlling Automatic Doorstop For Radio Frequency Based Public Facilities" by Galuh Satriawan. In this study, automatic doors use RFID to be able to open and close doors automatically slowly using radio frequencies on RFID [2, 6]. And for the first time, research was conducted on the use of PWM (Pulse Width Modulation) signals in automatic parking door speed control using PID system modeling.

The Proportional-Integral Derivative (PID) controller is the most popular controller today, because of its effectiveness, simple implementation and wide usage. The standard configuration of the PID controller has Kp, Ki and Kd parameters that can be determined so that the plant characteristics match the expected design criteria. The motor control system used in this study is the use of PWM signals on the PID control system using the Arduino UNO microcontroller. The speed of a DC motor can be controlled by varying its input voltage. A common technique for doing this is to use PWM. The higher the duty cycle, the greater the average voltage applied to the dc motor (High Speed) and the lower the duty cycle, the smaller the average voltage applied to the dc motor (Low Speed) [7-9].

METHODS

Research Flow

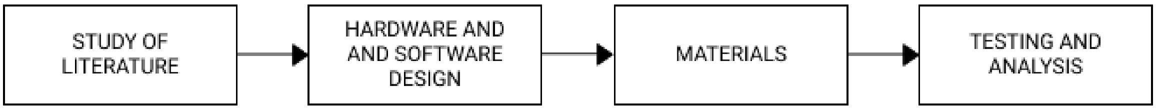


Figure 1. Block Diagram Flow Research

In the implementation of this final project, methods are used that aim to solve research problems. The methods that will be carried out in the research:

1. Literature Study
   Literature study in the form of collecting references from books and journals related to the basic PWM, PID, DC motor, and Arduino microcontroller signals, as well as conducting various searches for references and other relevant sources as a reference in making journals.
2. Design
   Designing hardware and software, in the form of a prototype by considering the needs and problems. design in the form of programs that will be arranged to drive the hardware.
3. Tools and materials
   Write down the components needed in the composition of the Journal and start designing and modeling the system that will be tested in the research.
4. Testing and Analysis
   Performing overall system testing both hardware and software by operating and communicating the software in each section, then running the hardware according to software commands and data from sensors.

System Planning

The design of the system in a block diagram is to make it easier to analyze the circuit as a whole. The block diagram is broadly shown in Figure 2.

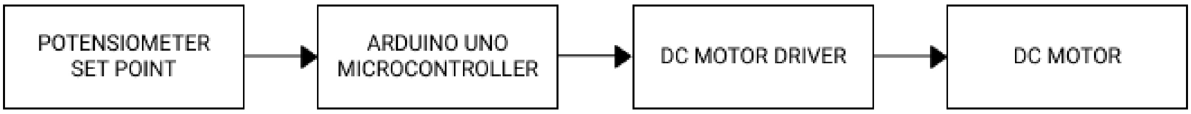


Figure 2. System Planning

In Figure 2. the potentiometer functions as an input for the set point in the form of an analog value derived from the potentiometer adc value. The signal from the potentiometer will enter and be processed by the microcontroller, in the microcontroller the value is processed into output logic and then sent to the motor driver which is to drive the DC motor. The rotation of the DC Motor shaft generates waves on the encoder which is connected to the DC Motor shaft.

Electronics Design

The schematic of the electronic design of the system is done using fritzing software. Fritzing itself is software that is generally used in conducting electronic design designs which are commonly used in research design. In designing the hardware, it is better to do as many trials as possible to get an effective cable wiring [10, 11].

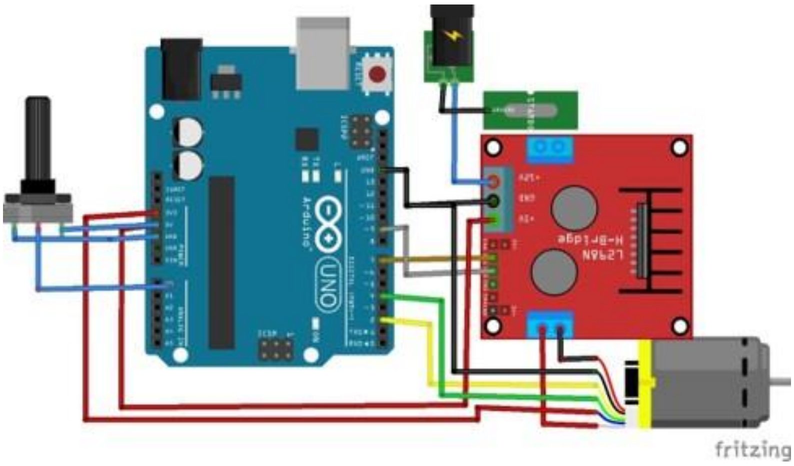
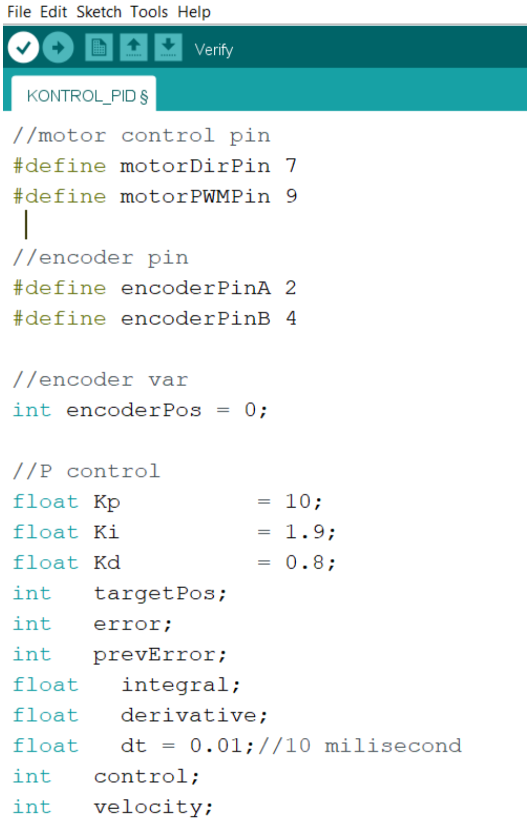


Figure 3. Electronics Design

Arduino UNO microcontroller is used to process the input and output signals given to each component. The analog input signal comes from the potentiometer to move up or down from the automatic doorstop, the potentiometer is connected to pin A0 on the Arduino UNO microcontroller and uses a 5V voltage. Furthermore, the input signal is processed on the Arduino UNO and the L298N Motor Driver, the L298N motor driver component uses a voltage of 12V for the DC motor and 5V for the L298N motor driver itself. INT1 and INT2 on the motor driver are connected directly to Arduino UNO on PINs 9 and 7 which are PWM PINs. After the input signal is processed, it will produce output in the form of up and down movement at the parking gate, the use of an encoder motor to drive the doorstop, on the motor encoder there are 6 wiring cables, 2 PINs for VCC and ground on the driving motor, 2 PINs for VCC and ground on the Encoder, and 2 PINs as phase A.

Software Design

Software design is a program design on the main Arduino UNO microcontroller board which functions to process DC motor encoder data, process adc values and process for PID controllers. Programming is done using Arduino IDE to write program code.




```
//motor control pin
#define motorDirPin 7
#define motorPWMPin 9

//encoder pin
#define encoderPinA 2
#define encoderPinB 4

//encoder var
int encoderPos = 0;

//P control
float Kp          = 10;
float Ki          = 1.9;
float Kd          = 0.8;
int   targetPos;
int   error;
int   prevError;
float   integral;
float   derivative;
float   dt = 0.01;//10 milisecond
int   control;
int   velocity;
```

Figure 4. Source Code

Using the Arduino IDE (Integrated Development Environment) application because this application is easy to create basic logic functions and easy to understand because it uses C language, in addition to Arduino IDE software to enter programs into Arduino, it takes a USB driver, and the Arduino board itself to program that has been made to run [12].

RESULT AND DISCUSSION

PID (Proportional–Integral–Derivative controller) is a controller to determine the precision of an instrumentation system with the characteristics of the presence of feedback on the system [13]. The PID controller will give action to the output based on the size of the error obtained. The output results will be the actuator to regulate the motor speed and position control on the motor in the work process of the parking doorstop system, the accuator is also called the Set Point. Error is the difference between the Set Point and the resulting output. The PID block diagram can be seen in the image below:

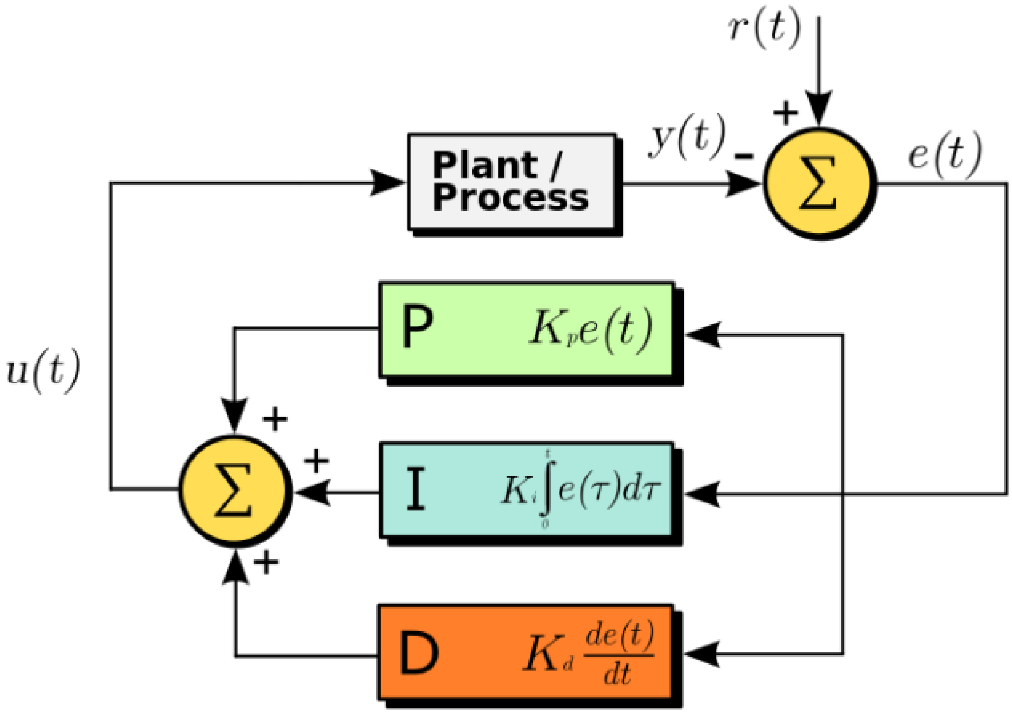


Figure 5. PID Block System

The PID controller can be modeled using the following formula:

$$mv(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right)$$

with: mv(t) as output from PID controller or Manipulated Variable, Kp as Proportional constant, Ti as Integral constant, Td as Detivative constant and e(t) as error (difference between set point and output/result)

Table 1. PID Control Effect

Respon	Rise Time	Overshoot	Settling Time	Steady State Error
Proporsional	Lower	Increase	Small Change	Lower
Integral	Lower	Increase	Increase	Eliminate
Derivatif	Small Change	Lower	Lower	Small Change

Trial Data Retrieval-1

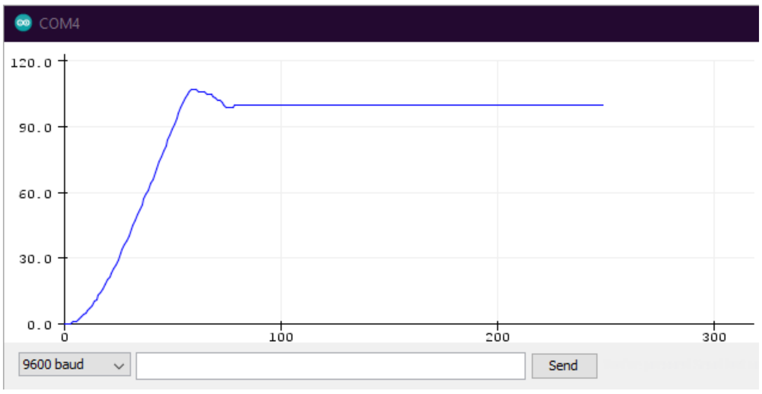


Figure 6. Trial Data Retrieval-1

The experimental data above is the result of a trial from the trial error method with control values at PID Kp=50, Ki=0, and Kd=0. The graphic results from experiment 1 are not optimal, do not reach steady state and there is still an overshoot

Trial Data Retrieval-2

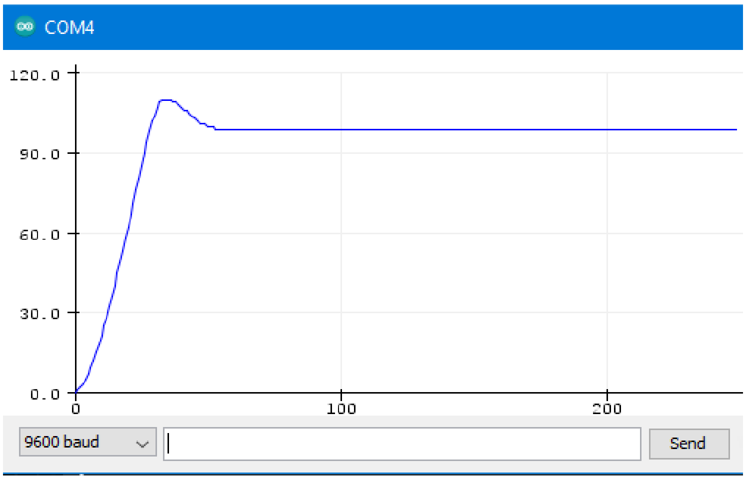


Figure 7. Trial Data Retrieval-2

The experimental data above is the result of a trial from the trial error method with control values at PID Kp=10, Ki=0, and Kd=0. The graphic results from experiment 2 are not optimal, do not reach steady state and there is still an overshoot.

Trial Data Retrieval-3

The experimental data above is the result of a trial from the trial error method with control values at PID Kp=10, Ki=1, and Kd=1. The graph results from experiment 3 are almost close to steady state but not optimal.

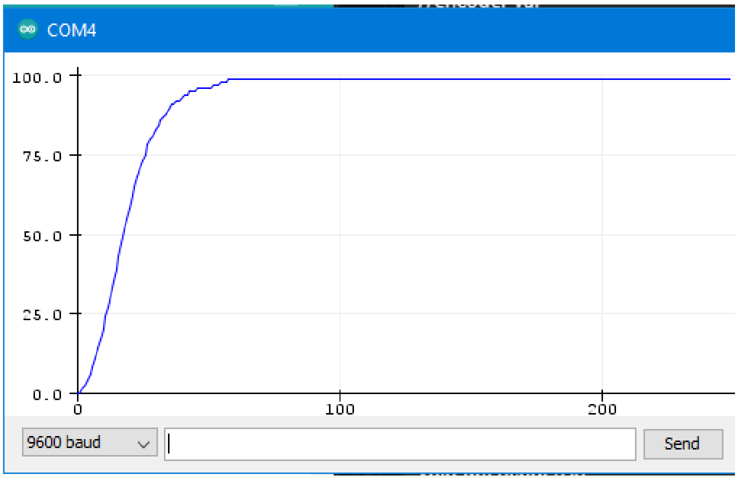


Figure 8. Trial Data Retrieval-3

Trial Data Retrieval-4

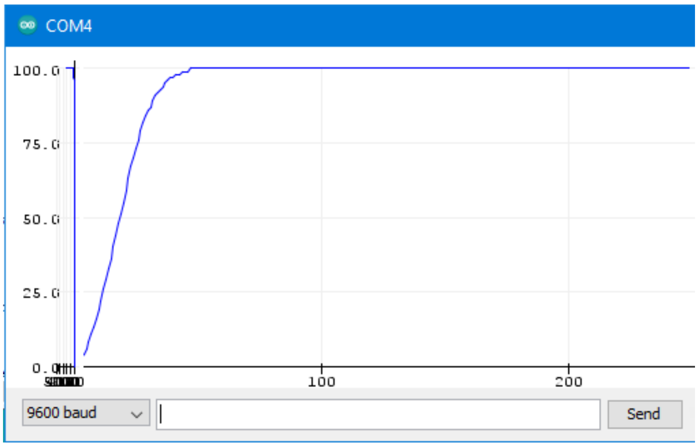


Figure 9. Trial Data Retrieval-4

The experimental data above are the results of trials from the trial error method with control values at PID Kp = 10, Ki = 1.9, and Kd = 0.8. The graph results from experiment 4 have been maximized and have reached steady state and there is no overshoot.

Table 2. Trial Data Results

Test	Value			Results	Status
	Kp	Ki	Kd		
1	50	0	0	Failed	Not reaching Steady State and Overshoot
2	10	0	0	Failed	Not reaching Steady State and Overshoot
3	10	1	1	Failed	Almost reached Steady State
4	10	1.9	0.8	Passed	Reach Steady State and Maximum

In the implementation of the PID control system on the parking gate using the trial error method, which is to get the desired value or set point. In the above result we used a set point of 100 we tried and finally got the Kp value. Ki, and Kd are maximum and reach steady state, namely at Kp = 10, Ki = 1.9, and Kd = 0.8. We can see in the experimental results at the values above in Figure 9, the response curve obtained is very stable with a short rise time and without any oscillations and overshoots. As in the picture below is the code used in the program.

```
targetPos = analogRead(A0)/10; //ADC max 1023
error    = targetPos - encoderPos;
integral += error * dt;
derivative = (error - prevError)/dt;
control = (Kp*error) + (Ki*integral) + (Kd*derivative);

velocity = min(max(control, -255), 255);
if(velocity >= 0)
{
    digitalWrite(motorDirPin, CW);
    analogWrite(motorPWMPin, velocity);
}
else
{
    digitalWrite(motorDirPin, CCW);
    analogWrite(motorPWMPin, 255+velocity);
}
//Serial.println(encoderPos);
prevError = error;
delay(dt*1000);
```

Figure 10. Source Code PID

Where in the above formula when implementing the experimental tool we use a potentiometer as the target post used.

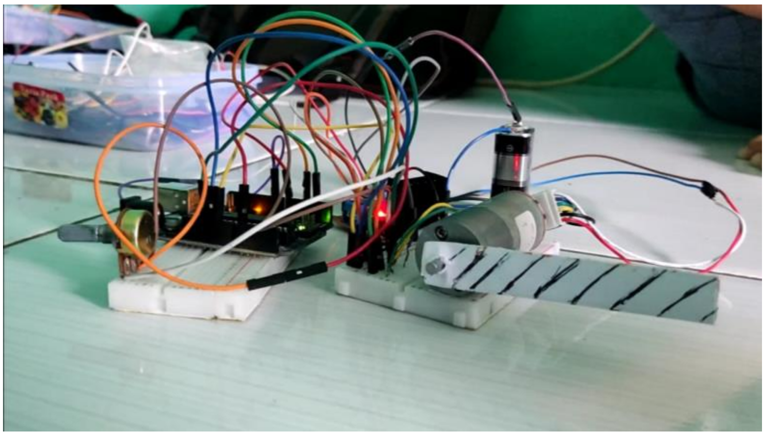


Figure 11. Hardware Design

The design of the tool was carried out using Arduino UNO as a microcontroller with and a potentiometer as input or driving the parking doorstop and a DC motor for the accuator or output on the parking doorstop.

CONCLUSION

From the discussion in the conclusions of this study it can be concluded that:

1. Implementation of the PID controller on the parking gate system can be done by utilizing the PWM signal on the motor driver but still using the input voltage in the 6V-12V range.
2. The use of the trial error method succeeded in getting the most effective value when utilizing the PWM signal pin on the motor driver with Kp = 10, Ki = 1.9, and Kd = 0.8 to get the best steady state and without overshoot so that the automatic parking gate latch can open and close effectively (not too fast nor too slow)
3. The use of the right method can maximize the potential of this PID control by utilizing the PWM signal on the motor driver so that the DC motor movement is in accordance with the desired design.

REFERENCES

[1] Kantor Kepolisian Republik Indonesia, "Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis tahun 1987-2013," *Badan Pusat Statistik*. p. 2014, 2013, [Online]. Available: http://www.bps.go.id/linkTabelStatis/view/id/1413.

[2] Y. T. Utami and Y. Rahmanto, "Rancang Bangun Sistem Pintu Parkir Otomatis Berbasis Arduino dan RFID," *JTST*, vol. 02, no. 02, pp. 25–35, 2021.

[3] H. H. Vo *et al.*, "Sliding Mode PWM-Direct Torque Controlled Induction Motor Drive with Kalman Filtration of Estimated Load," doi: 10.25073/jaec.202154.342.

[4] S. Nakamori, "Arduino-based PID Control of Temperature in Closed Space by Arduino-based PID Control of Temperature in Closed Space by Pulse Width Modulation of AC Voltage Pulse Width Modulation of AC Voltage Arduino-based PID Control of Temperature in Closed Space by Pulse Width Modulation of AC Voltage," 2021. [Online]. Available: https://digitalcommons.fairfield.edu/ijcase.

[5] A. MA'ARIF, R. ISTIARNO, and S. SUNARDI, "Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 2, p. 374, Apr. 2021, doi: 10.26760/elkomika.v9i2.374.

[6] B. A. I. Hendy Prasetyo, Tarmukan, "Implementasi Pengendali Posisi Sudut Motor DC Berbasis PID Dengan

Interface Mikrokontroler dan Matlab Pada Laboratorium Sistem Kendali Digital," *J. ELKOLIND*, 2017.

[7] R. Muhardian, "Kendali Kecepatan Motor DC Dengan Kontroller PID dan Antarmuka Visual Basic," *JTEV (JURNAL Tek. ELEKTRO DAN VOKASIONAL)*, vol. 06, 2020, [Online]. Available: http://ejournal.unp.ac.id/index.php/jtev/index.

[8] M. Mahmud, S. M. A. Motakabber, A. H. M. Zahirul Alam, and A. N. Nordin, "Utilizing of Flower Pollination Algorithm for Brushless DC Motor Speed Controller," Dec. 2020, doi: 10.1109/ETCCE51779.2020.9350866.

[9] K. Vanchinathan and K. R. Valluvan, "A Metaheuristic Optimization Approach for Tuning of Fractional-Order PID Controller for Speed Control of Sensorless BLDC Motor," *J. Circuits, Syst. Comput.*, vol. 27, no. 8, Jul. 2018, doi: 10.1142/S0218126618501232.

[10] M. Mahmud, S. M. A. Motakabber, A. H. M. Z. Alam, and A. N. Nordin, "Control BLDC motor speed using PID controller," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 477–481, 2020, doi: 10.14569/ijacsa.2020.0110359.

[11] S. Kasmungin, "Perancangan Sistem Kontrol Kecepatan Motor DC Dengan PID Labview," *J. Manutech*, 2010.

[12] M. Irhas, S. Asyiqah, A. Ilham, and I. Artikel, "Penggunaan Kontrol PID Dengan Berbagai Metode Untuk Analisis Pengaturan Kecepatan Motor DC," *J. Fis. dan Ter.*, vol. 7, no. 1, pp. 78–86, 2020, [Online]. Available: http://journal.uin-alauddin.ac.id/index.php/jft.

[13] M. Ruswandi Djalal, T. Energi, J. Teknik Mesin, and P. Negeri Ujung Pandang Jalan Perintis Kemerdekaan, "Penalaan Optimal Kendali Motor DC Berbasis ANT Colony Optimization," *J. Teknol. Univ. Muhammadiyah Jakarta*, vol. 12, no. 1, 2020, doi: 10.24853/jurtek.12.1.49-56.

Author Biography

Ario Gustama, born in Bogor, 29 August 1999. Bachelor of Engineering in Electrical Engineering Study Program, Concentration in Control and Computer Engineering, Singaperbangsa Karawang University 2018 - now.

Ridwan Satrio Hadikusuma, born in Bogor, July 1, 2000. Bachelor of Engineering in Electrical Engineering Study Program, Concentration in Telecommunication Engineering, Singaperbangsa Karawang University 2018 – now.

Rahmat Hidayat, born in Indramayu, March 19, 1989. Completed his Bachelor's degree (S1) at the University and then continued his Postgraduate Education (S2) at. Since he has been a permanent lecturer in Electrical Engineering at Singaperbangsa Karawang University and has been a lecturer in the Electrical Engineering Study Program since.