Journal of Electrical Technology UMY (JET-UMY), Vol. 7, No. 2, December 2023
ISSN 2550-1186 e-ISSN 2580-6823

Implementation of a PI-Controlled Closed-Loop System for Mecanum Wheel Coordination in an Omni-Directional Robot

Maulana Hakim Susatyo[1], Muhamad Yusvin Mustar[*1, 2]

[1]Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, Bantul 55183, Indonesia
[2]Centre of Artificial Intelligence and Robotics Studies, Innovation and Research Centre, Universitas Muhammadiyah Yogyakarta, Bantul 55183, Indonesia
*Corresponding author, e-mail: yusvin@umy.ac.id

Abstract – *The MR CRUSH robot is designed for participation in the 2023 Indonesian Thematic Robot Contest (KRTMI). In the motor part of the MR CRUSH robot, each wheel is designed to rotate independently and using a mecanum wheel system. Mecanum wheels are a type of wheel that has the ability to move in all directions by combining rotational speed and direction. To maximize this capability, the force generated on each mecanum wheel must be the same as set point. One of the influences of the amount of force produced is the mecanum wheel rotation speed. This research aims to implement a closed-loop control system with a Proportional-Integral (PI) controller. By implementing this PI controller, it ensures that the rotation speed of the dc motor corresponds to the set point value given, so that the four dc motors that drive the mecanum wheel will have the same rotation speed. The parameters for the PI controller used in this research are Kp = 0.3 and Ki = 0.08. Based on the results, the closed-loop control system with the designed PI controller produces constant motor rotation speed at the set point with 0 steady-state error.*

Keywords*: PI Controller, Mecanum Wheels, DC Motor*

I. Introduction

The MR CRUSH robot is designed for participation in the 2023 Indonesian Thematic Robot Contest (KRTMI). At the regional level of the KRTMI competition, the robot has the objective of picking up coins one by one on a rack totaling 10 coins and then moving around carrying the coins to be placed at certain locations within 3 minutes. To make the robot's movement more effective, the MR CRUSH robot has four wheels designed to rotate independently and using a mecanum wheel system [1].

The mecanum wheel system implemented on the MR CRUSH robot allows the robot to move in all directions without turning, making the robot's movement effective. The combined results of the rotational speed and direction of each mecanum wheel will determine the direction movement of the robot. To maximize this capability, the force generated on each mecanum wheel must be the same as set point. One of the influences of the amount of force produced is the mecanum wheel rotation speed [2].

The control system used on the previous MR CRUSH robot used an open-loop control system that had no feedback input. When each motors that drive the mecanum wheel are given the same Pulse Width Modulation (PWM) value, the rotation speed output produced by the dc motors is not the same as each other. Therefore, it takes a lot of experimentation to get the exact PWM value for each dc motors to produce the same rotation speed.

This research proposes the application of a closed loop control system with a Proportional-Integral (PI) controller. By implementing this PI controller, it ensures that the rotation speed of the dc motor corresponds to the set point value given, so that each dc motors that drive the mecanum wheel will have the same rotation speed.

II. Methods

II.1. *Hardware Design*

The block diagram of the hardware system on the MR CRUSH robot is shown in Figure 1. This system has a configuration including input, output and process. In the input section there is an encoder

Manuscript received October 2023, revised December 2023

sensor and wireless receiver. In the output section there is a VNH2SP30 motor driver and a JGB37-520 dc motor. In the processing section, RobotDyn Mega2560 Pro is used. Meanwhile, the voltage source used comes from a 4 cell battery with a voltage of 14V which is then regulated to 12V, 5V and 3.3V respectively. The encoder sensor and wireless receiver are connected to a voltage of 3.3V, the microcontroller is connected to a voltage of 5V. The motor driver requires a voltage of 5V as a voltage source for the electronics, while 12V is a voltage source for the dc motor.

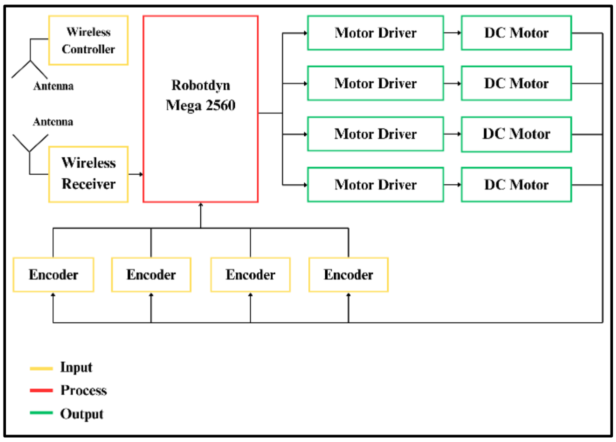


Fig. 1. MR CRUSH Robot Hardware System Block Diagram

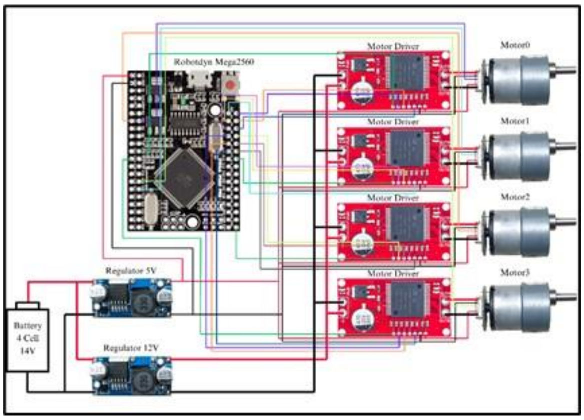


Fig. 2. MR CRUSH Robot Hardware System Schematic

A schematic of the motor driver and encoder sensor system circuit is shown in Figure 2. The motor driver system is designed using 4 VNH2SP30 motor drivers where each motor driver controls one dc motor. As a result, each dc motor can have its rotation speed controlled separately via the PWM value. The top view of the robot is shown in the Figure 3. The configuration of the motor driver pin can be seen in Table 1. The sensor used to detect the rotation speed of a DC motor is an encoder sensor. The encoder sensor is located at the bottom of the robot and is integrated with the dc motor. The configuration of the encoder sensor pin can be seen in Table 2. The bottom view of the robot is shown in the Figure 4.

Table 1. Motor driver pin configuration

Motor	Pin Function			
	PWM Pin	EN Pin (Enable)	INA Pin (Motor Driver Direction)	INB Pin (Motor Driver Direction)
Motor 0	10	A4	A2	A0
Motor 1	12	A1	A3	A5
Motor 2	46	A10	A8	A6
Motor 3	13	A11	A9	A7

Table 2. Encoder sensor pin configuration

Encoder	**Pin Function**
	Encoder A
Encoder0	2
Encoder1	20
Encoder2	3
Encoder3	18

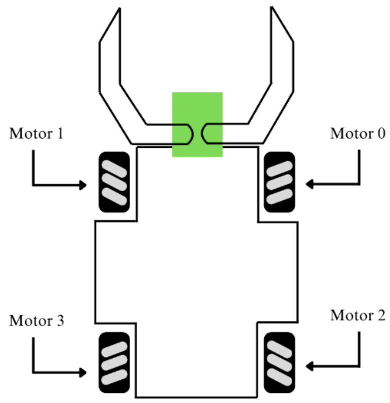


Fig. 3. Top View of the MR CRUSH Robot

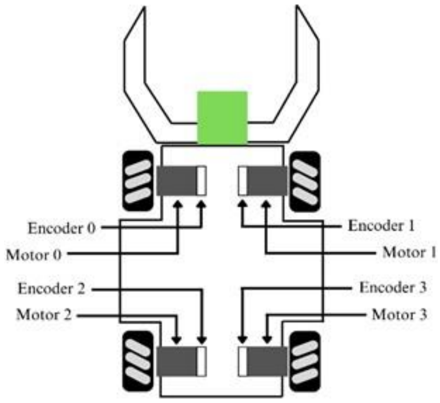


Fig. 4. Bottom View of the MR CRUSH Robot

II.2. *Software Design*

The control system block diagram is shown in Figure 5. This control system is categorized as a closed-loop control system. This control system has a configuration including input, processor (microcontroller) and output. The input device is an encoder sensor, the processing device is an ATmega2560 microcontroller and the output device is a VNH2SP30 motor driver and a JGB37-520 dc motor. The type of controller used is a PI controller. The set point is the input initial reference value which will be compared with the value from the feedback to produce a difference value called error. The error results obtained will be processed by the PI algorithm and produce a control signal in the form of PWM. This PWM signal will then be sent to the motor driver so that it can function to regulate the speed of the dc motor. The resulting output is the dc motor rotation speed per minute (RPM). The control system flowchart in software design is shown in Figure 6.

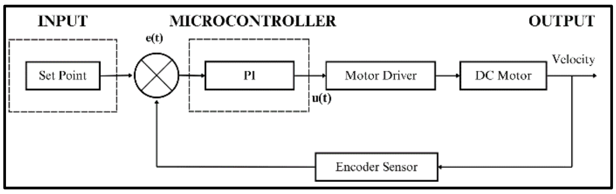


Fig. 5. Control System Block Diagram

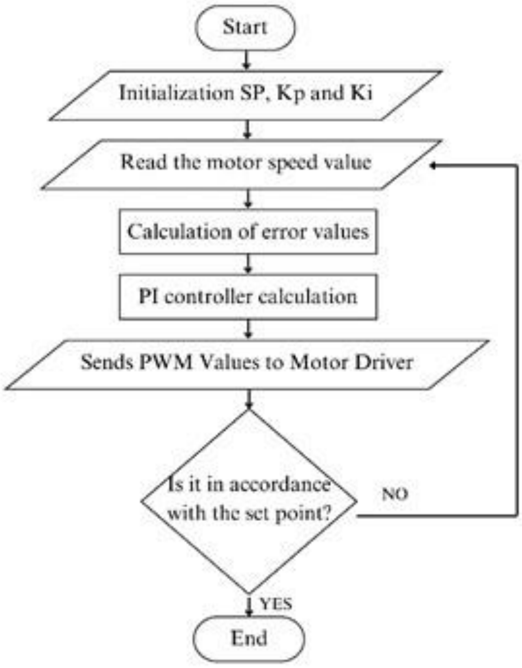


Fig. 6. Control System Flowchart

II.3. *RPM Counter*

The encoder sensor reads the rotation speed of the motor by providing feedback in the form of high and low pulse signals to the microcontroller. Then the signal will be calculated to get a value in the form of motor revolutions per minutes (RPM). To get the value of revolutions per minute (RPM), the following equation is used.

$$RPM = \frac{n}{n_r \times interval} \times 60 \tag{1}$$

Where n is the number of pulse signals detected by the encoder sensor, nr is the number of pulse signals in one turn, interval is the time interval used in the program and 60 is the second to minute conversion constant. This equation is implemented in the program as follows.

```
int interval = 50;
currentMillis = millis();
 if(currentMillis - previousMillis >
interval) {
   previousMillis = currentMillis;
   rpm = (float)((n/(112*0.05))*60);
 }
```

II.4. Proportional-Integral Controller

Proportional-Integral is a control algorithm in a closed-loop control system which functions to determine the level of response and precision of a system. The PI controller is a type of controller that is formed from combining two controllers, namely a proportional controller and an integral controller. These two controllers have different control characteristics. The PI controller reads the feedback value from the sensor, then performs calculations to get the desired set point value by adding the P and I controls. In designing the PI controller, it is necessary to tune the P and I parameters to get an ideal and precise response in a system control [3]–[5]. The PI controller equation is as follows.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt \qquad (2)$$

where u(t) is a control signal, Kp is the proportional control parameter value, Ki is the integral control parameter value, e(t) is the error or difference between the reference value and the feedback value. Proportional control deals with the error between the reference value and the feedback value. The integral control deals with the sum of all errors. This equation is implemented in the program as follows.

```
int interval = 50;
currentMillis = millis();
 if(currentMillis - previousMillis >
interval) {
   previousMillis = currentMillis;
   error = sp - rpm;
   sum_error = sum_error + error;
   motorSpeed = ((kp*error)+(ki*
sum_error));
   analogWrite(PWM, motorSpeed);
 }
```

II.5. Tuning Proportional-Integral Parameter

Tuning is the process of determining the values of proportional and integral control variables to achieve an ideal and precise response in a control system. Tuning is done by trying several Proportional and Integral parameter (Kp, Ki) values to observe system response differences with each parameter value change.

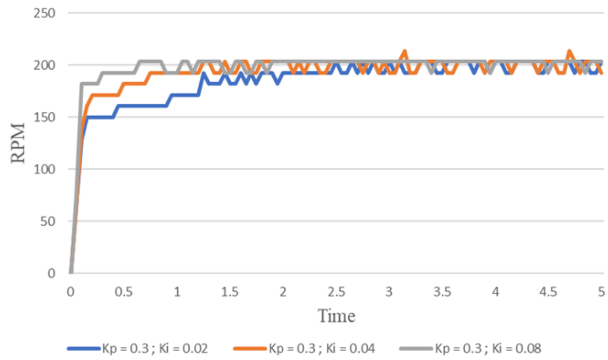


Fig. 7. Tuning System Response Graph

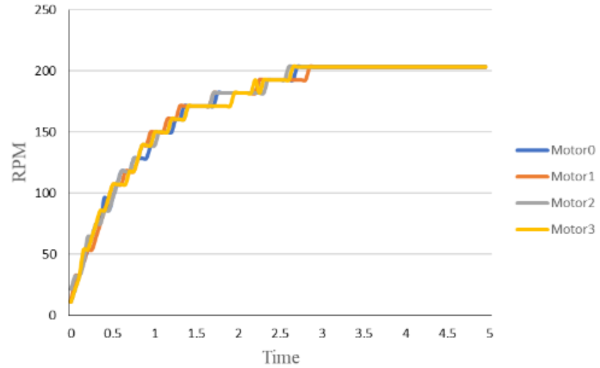
Based on the system response graph shown in Figure 7, the best PI control parameters (Kp, Ki) are Kp = 0.3 and Ki = 0.08. The system response reached the set point with 0 steady-state error. There is no overshoot, and the rise time has a value of 0.1 seconds. Therefore, the parameter values used for this research are Kp = 0.3 and Ki = 0.08.

III. Result and Discussion

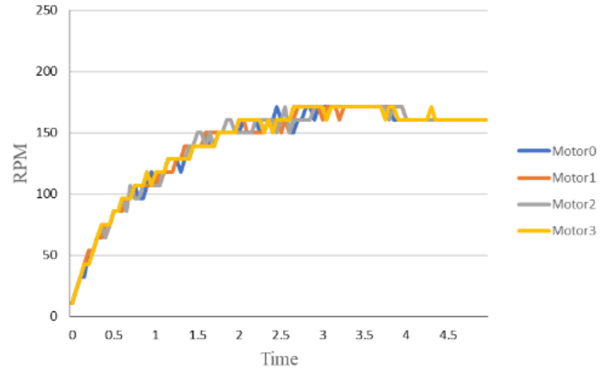
This section describes a comparison of an open loop-control system compared to a closed-loop control system using a PI controller. This test was carried out on two different surfaces, namely plywood and carpet. The results of the open-loop control system are shown in Figure 8, and the results of the closed-loop control system using the PI controller are shown in Figure 9. The set point value used is 200 RPM and the PI control parameters used are Kp = 0.3 and Ki = 008 with interval time 50 ms. The difference that can be seen by comparing the two control systems is seen in the system response in reaching the set point value. In the test results on the carpet surface, the open-loop control system was unable to reach the set point with a steady-state error of 29 RPM.

Meanwhile, in a closed-loop control system with a PI controller, the system is able to reach the set point value with a steady-state error of 0 RPM. This happens because in an open-loop control system there is no correction if the output speed of the dc motor does not reach the set point value. Meanwhile, a closed-loop control system can easily reach the set point value. Therefore, the system response produced by a closed-loop system with a PI controller is better in reaching the set point value. In an open-loop control system, the rise time obtained is 2 seconds. Meanwhile, in a closed-loop control system with a PI controller, the time required to reach the set point is faster than an open-loop system with the rise time value obtained is 0,4 seconds.

The results of the design and application of the mecanum wheel motor speed control system design with PI controller on Mr. Crush Robot are shown in Figure 10, at the Indonesian Thematic Robot Contest (KRTMI) in 2023.

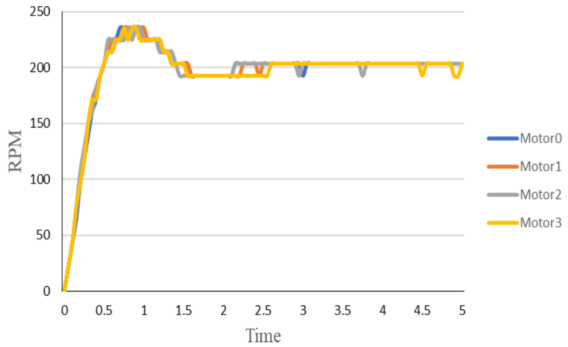


(a) plywood surfaces

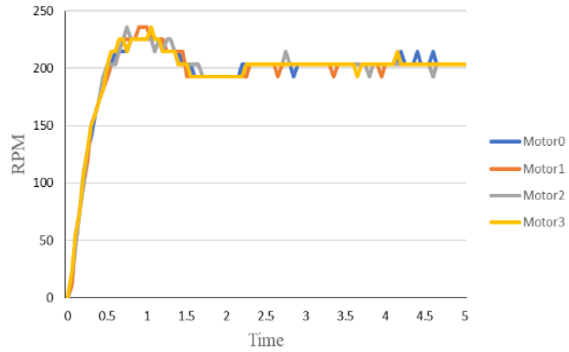


(b) carpet surfaces

Fig. 8. Response Results of The Open-Loop Control System



(a) plywood surfaces



(b) carpet surfaces

Fig. 9. Response Results of The Closed-Loop Control System Using a PI Controller

(a) MR. Crush Robot

(b) KRTMI Competition

Fig. 10. Application of mecanum wheel motor speed control system design with PI controller on Mr. Crush Robot

IV. Conclusion

This research proposes the application of a closed loop control system with a Proportional-Integral (PI) controller. Based on the result, The system response by the closed-loop control system with PI controller is better in achieving the set point value than the open-loop control system in tests on both surfaces. In the test results on the carpet surface, the open-loop control system was unable to reach the set point with a steady-state error of 29 RPM. Meanwhile, in a closed-loop control system with a PI controller, the system is able to reach the set point value with a steady-state error of 0 RPM. The rise time of a closed-loop control system with a PI controller is faster than that of an open-loop control system. In an open-loop control system, the rise time obtained is 2 seconds. Meanwhile, in a closed-loop control system with a PI controller, the time required to reach the set point is faster than an open-loop system with the rise time value obtained is 0,4 seconds. The PI control parameters used in this research are Kp = 0.3 and Ki = 0.08.

Acknowledgements

This work was supported by Universitas Muhammadiyah Yogyakarta through Center of Artificial Intelligence and Robotic Studies, Research and Innovation Institute of Universitas Muhammadiyah Yogyakarta.

References

[1] B. Kusumoputro *et al.*, “Buku Pedoman Kontes Robot Indonesia (KRI) Tahun 2023,” Balai Pengembangan Talenta Indonesia Pusat Prestasi Nasional Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia, hal. 1–150, 2023.

[2] N. Thongpance dan P. Chotikunnan, “Design and Construction of Electric Wheelchair with Mecanum Wheel,” *J. Robot. Control*, vol. 4, no. 1, hal. 71–82, 2023, doi: 10.18196/jrc.v4i1.17095.

[3] R. Hansza dan S. I. Haryudo, “Rancang Bangun Kontrol Motor DC dengan PID Menggunakan Perintah Suara dan Monitoring Berbasis Internet of Things (IOT),” *J. Tek. Elektro*, vol. 9, no. 2, hal. 477–485, 2020.

[4] “The PID Controller & Theory Explained,” 2023. https://www.ni.com/en/shop/labview/pid-theory-explained.html (diakses 1 Februari 2024).

[5] A. Ma’arif, “Embedded Control System of DC Motor Using Microcontroller Arduino and PID Algorithm,” *IT J. Res. Dev.*, vol. 6, no. 1, hal. 30–42, 2021, doi: 10.25299/itjrd.2021.vol6(1).6125.

Authors’ information

Maulana Hakim Susatyo received B.Eng degree from Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia, in 2024. His research interests are in robotics and control engineering.

Muhamad Yusvin Mustar Received Diploma degree Electrical Engineer from Universitas Haluoleo, Kendari in 2009, B.Sc. degree from Department of Electrical Engineering Universitas Muhammadiyah Yogyakarta in 2011, M.Eng. degree from Department of Electrical Engineering and Informatics Technology, Universitas Gadjah Mada, Yogyakarta, Indonesia in 2014. Muhamad Yusvin Mustar, M.Eng. is a Lecturer in Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia. His research interests are in human-robot interaction, robotics and electronics engineering.