Heading Control on Differential Drive Wheeled Mobile Robot with Odometry for Tracking Problem

Fahmizal
Department of Electrical Engineering and Informatics Vocational College, Universitas Gadjah Mada
Yogyakarta, Indonesia
fahmizal@ugm.ac.id

Angga Priyatmoko
Department of Electrical Engineering and Informatics Vocational College, Universitas Gadjah Mada
Yogyakarta, Indonesia
priyatmokoangga@gmail.com

Esa Apriaskar
Department of Electrical Engineering Universitas Negeri Semarang
Semarang, Indonesia
esa.apriaskar@mail.unnes.ac.id

Afrizal Mayub
Graduate School of Science Education University of Bengkulu
Bengkulu, Indonesia
afrizalmayub@unib.ac.id

***Abstract*—This paper describes the process of designing odometry and heading controls from a differential steering wheeled mobile robot (DSWMR). The odometry system aims to estimate the position relative to the initial position of the robot to estimate changes in position from time to time in the trajectory tracking process. The problem that often arises in the tracking problem is the heading error that can be caused by a slip on the robot wheel or an irregularity between the speed of the DC motor on the robot wheel. Heading errors in DSWR can be obtained with the help of a rotary encoder located on a DC motor. This work applied PID control to obtain the heading error close to 0 degrees on the odometry system for trajectory tracking. It works by controlling the rotating speed of each DC motor on the robot wheel. The results of the PID control parameters implemented on the DSWMR were obtained from the results of tuning experiments.**

Keywords—DSWMR, heading control, tracking problem

I. INTRODUCTION

The rapid development of science has had a considerable impact on human life to learn and develop the technology. One of them is robotics technology, where more and more robot innovations are created both in terms of saving energy sources and time efficiency. In modern industry, robots have taken over the positions of workers in factories. Both in manufacturing and process industries, robots tend to have become the main drivers of the industry. The main reason for using robots is because robots can be ideal workers with high levels of accuracy and efficiency, and more importantly, operating costs tend to be low for large-scale production [1].

The more agile development of the world of robotics makes researchers and engineers motivated to create a robot that can support the needs of the industrial world. Robots that can move following the predefined path accurately are one example of applications that are useful in industrial society [2]. It is intended that robots should move autonomously without the need for operators to increase the effectiveness of the industrial process. This autonomous driving requires the ability of a robot to solve a tracking problem [3]. That is one reason for the importance of researching the tracking problem in robots.

Many studies often used a mobile robot as an object in testing a way to complete a tracking problem. In [4], a two-wheeled mobile robot resolves this problem by using a proportional-integral (PI) controller and comparing it with a model-predictive controller. In that research, information on the dynamics model of the mobile robot used is something mandatory in supporting the controller's design. [5] also requires information about the dynamics model of a mobile robot to make the robot can follow a particular trajectory. That study uses the robust controller method that can provide better stability against the uncertainty which might occur. Has a different viewpoint, [6] involves image processing to detect lines to accomplish the trajectory tracking mission. The camera sensor here performs as a motion sensor used in odometry techniques. The use of cameras and the addition of fuzzy logic is proven to help the robot can follow the line and reach the coordinates of the expected destination.

In determining the system of movement, the robot can use two types of media; they are outside and inside. Examples of the use of external media are lines and walls [7]. However, the use of media from outside is less efficient. The robot should be able to determine the direction of movement from within the robot itself. The sensor that is commonly used to obtain position data from a robot is a rotary encoder. Data from this sensor will be included in the calculation of odometry to produce the relative position of the robot [8]. The odometry method has been widely used for the completion of a robot in the process of carrying out a tracking. A trajectory tracking is a method for determining the movement pattern of a robot from the starting point to the end point. There are two types of algorithms for trajectory tracking, namely off-line path trajectory and on-line trajectory [9].

In this study, we made a mobile robot with a differential drive driver that can reach the desired target position. Robots with this type are usually used in modern industries to move goods from one place to another. The mobile robot uses odometry technique by utilizing a rotary encoder to detect the movement of the robot. In contrast to research [4]–[6] which looks at the performance of the results of tracking from the position generated by the robot, this study focuses on evaluating the performance of heading control on a mobile robot. PID control is used with several variations to see differences in performance.

This paper is organized as follows. In Part II, the methodology in this study was proposed. Part III explains the design of DSWMR robot hardware. Section IV discusses experiments that show the effectiveness and benefits of the proposed method. Part V is the conclusion of this study.

II. METHODS

A. Kinematics of Differential Steering Mobile Robot

In this paper, the type of steering or control of the movement of the robot used is the differential steering type. The geometry and kinematic parameters of the differential steering shown in Fig. 1. The position and heading of the robot are explained by vector $q=\left[x_Q, y_Q, \varphi\right]^T$, with

978-1-7281-3090—3/19/$31.00 ©2019 IEEE

(x_Q, y_Q) indicating the location of point Q (the center of the body axis of the robot being moved) and φ is the heading of the robot in the framework of global coordinates (G_{lcs}). $\dot{\theta}_l$ and $\dot{\theta}_r$ denote the angular velocity of the left and right wheels. There are three variables to control (x_Q, y_Q, φ) with only two control inputs. Therefore, the system is under-actuated. The relationship between the linear velocity of the robot v and its angular velocity ω with the rate of change of position and heading are explained in (1).

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

Linear speed and heading robot angle are explained in (2) - (3),

$$v = \frac{r\left(\dot{\theta}_r + \dot{\theta}_l\right)}{2} \quad (2)$$

$$\omega = \frac{r\left(\dot{\theta}_r - \dot{\theta}_l\right)}{2b} \quad (3)$$

where r is the wheel radius, and $2b$ is the base length of the wheel, which is the distance between the wheels measured along the axis of the wheel rotation. By substituting (2) and (3) into (1), (4) – (6) are obtained.

$$\dot{x}_Q = \frac{r}{2}\left(\dot{\theta}_r \cos\varphi + \dot{\theta}_l \cos\varphi\right) \quad (4)$$

$$\dot{y}_Q = \frac{r}{2}\left(\dot{\theta}_r \sin\varphi + \dot{\theta}_l \sin\varphi\right) \quad (5)$$

$$\omega = \frac{r\left(\dot{\theta}_r - \dot{\theta}_l\right)}{2b} \quad (6)$$

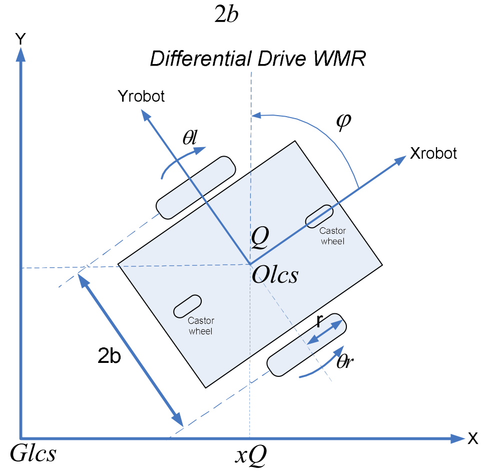


Fig. 1. Geometry and kinematic parameters of the differential steering

B. Odometry System in Differential Steering Mobile Robot

Odometry is the use of data from the movement of an actuator to estimate changes in coordinate position over time. Odometry is used to calculate the coordinates of location relative to the initial position [10], [11]. In the wheeled odometry system, the sensor used is a rotary encoder to detect the number of wheel turns. There are three main parameters in calculating the coordinates of the robot position, namely diameter of the freewheel $DW_{(i)}$, the number of encoder resolution ($resolusi_{enc}$), and the number of rotary encoder pulses produced ($pulse$). To calculate the circumference of the freewheel ($KW_{(i)}$), $DW_{(i)}$ is multiplied by π as described in (7). When using the axis of the cartesian motion, changes in the position x coordinates (x_{tempuh}) and y coordinates (y_{tempuh}) can be calculated using (8) and (9).

$$KW_{(i)} = DW_{(i)} \times \pi \quad (7)$$

$$x_{tempuh} = \frac{pulse_{(x)}}{resolusi_{enc}} KW_{(x)} \quad (8)$$

$$y_{tempuh} = \frac{pulse_{(y)}}{resolusi_{enc}} KW_{(y)} \quad (9)$$

To find out the direction of the robot heading (φ), the current position coordinates (x_{saat_ini}, y_{saat_ini}) relative to the movement of the initial position (x_{awal}, $y_{\_awal}$) as presented in Fig. 2. Then the robot heading can be obtained by (10). Whereas to find the length of distance from the current position to the initial state can be calculated by (11).

$$\varphi = \tan^{-1} \frac{y_{tempuh}}{x_{tempuh}} \quad (10)$$

$$Distance = \sqrt{(x_{saat\_ini} - x_{awal})^2 + (y_{saat\_ini} - y_{awal})^2} \quad (11)$$

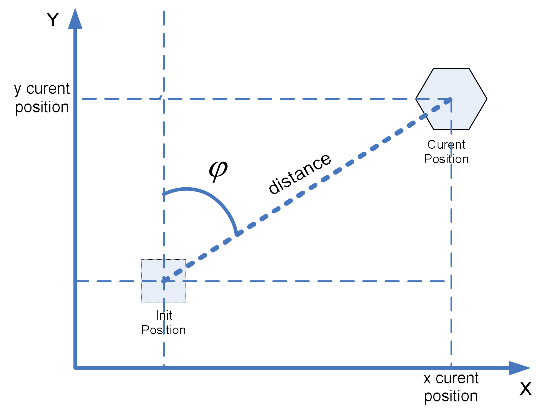


Fig. 2. Robot heading calculations on the odometry system

Fig. 3. presents changes in coordinates and heading on the odometry system over time. The change value consists of changes in the position of the robot's heading and coordinates to the robot's global coordinates, where φ is the direction of the robot heading, and the bearing target (β) is the angle between the robot's current position and the destination point. To determine the β value and robot heading error (α) can be calculated using (12) and (13). Furthermore, to find out the target distance can be calculated using (14).

$$\beta = 90° - \left[tan^{-1} \frac{y_{tujuan} - y_{saat\_ini}}{x_{tujuan} - x_{saat\_ini}} \right] \quad (12)$$

$$\alpha = \beta - \varphi \quad (13)$$

$$Target = \sqrt{(x_{tujuan} - x_{saat\_ini})^2 + (y_{tujuan} - y_{saat\_ini})^2} \quad (14)$$

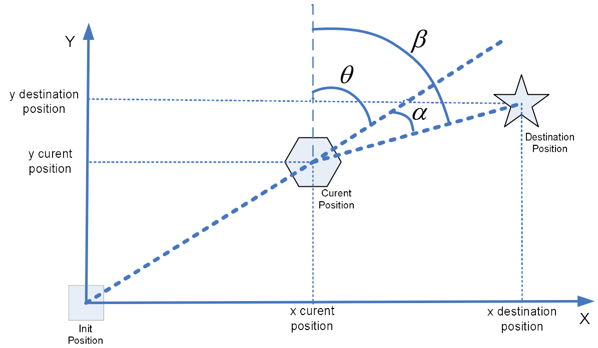


Fig. 3. Illustration of the position of the robot on the cartesian axis in the odometry system

C. Digital PID Control and Parameter Tuning

Digital PID control is another form of PID control programmed and run using a computer or microcontroller. To be able to implement digital PID on a computer or microcontroller, analog PID control design must be changed to digital form. The mathematical form of the PID control is in (15).

$$u(t) = K_p\left[e(t) + \frac{1}{\tau_i}\int e(t)dt + \tau_d \frac{d}{dt}e(t)\right] \tag{15}$$

With $Ki = \frac{1}{\tau_i}$ and $Kd = \tau_d$, then the integral and differential forms can be written in discrete form as in (16) - (17).

$$\int_0^t e(t)\, dt \approx \mathrm{T}\sum_0^{\mathrm{k}} \mathrm{e(k)} \tag{16}$$

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{T} \tag{17}$$

Thus, it obtains the form of a discrete PID control, as stated in (18).

$$u_{(k)} = K_p e_k + K_i T\sum_0^k e_k + \frac{1}{T}K_d(e_k - e_{k-1}) \tag{18}$$

where Kp is a proportional constant, Ki is an integral constant, Kd is a differential constant, e_k is an error value, and the e_{k-1} is the value of the previous error and Ts is the sampling time.

Tuning experiment is a process carried out to obtain optimal control results through an experiment. The essence of tuning experiment is to determine the value of three parameters contained in the PID control, namely proportional constant (Kp), integral constant (Ki) and differential constant (Kd). Several methods can help in tuning PID control parameters with an experiment such as the Ziegler-Nichols method, the Cohen-Coon method, and the empirical method. In this study, the PID control was tuned using empirical methods based on the effect of each control on the system. The steps for determining the parameters Kp, Ki, and Kd is adopted from [12], as follows:

a. The first step is to use proportional control first, ignore the integral constant and the derivative by giving them zero values
b. Add the maximum proportional constant until the state is stable but the robot is still oscillating.
c. To reduce oscillation, add a differential constant by dividing the proportional constant, and observe the state of the robot system to be stable and more responsive.
d. If the robot system has been stable, integral control can be optional to improve the performance of the robot. However, giving improper integral value can make the robot system unstable.
e. The value of sampling time also affects the calculation of PID, especially when using integral and differential controls.
f. Re-check system performance to get satisfactory results.

III. System Design

A. Design of Robot Mechanics and Electronics

The design of the robot mechanics in this study was designed using two wheels that had a rotary encoder as an odometry process counter. Fig. 4 displays the overall appearance of the robot design, while Fig. 5 presents the display of the base and bottom of the robot.

Fig. 4. Overall robot design

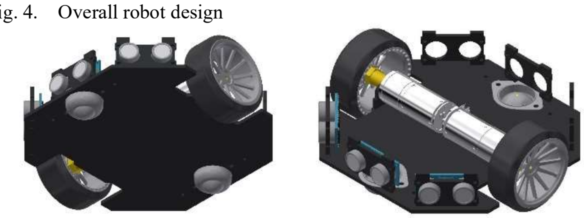


Fig. 5. The mechanical design of the robot.

The robot's electronic system uses an Arduino Mega, which is placed on the top of the robot. The wheel rotary encoder sensor is used to monitor direction and number of DC motor rotation of the robot. The electronic system diagram block configuration of the robot as a whole is presented in Fig. 6. Furthermore, the results of the printed circuit board (PCB) design of this robot are shown in Fig. 7.

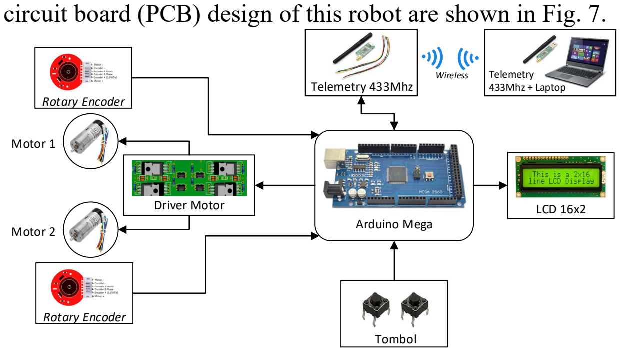


Fig. 6. Schematic diagram of omni robot electronic devices

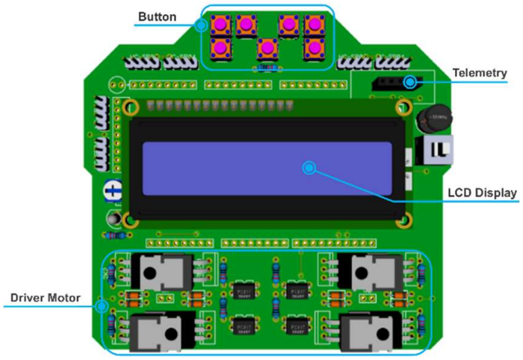


Fig. 7. PCB design of the robot

B. PID Control Design

When the robot is running, slip problems that occur on the robot wheels appear, causing the direction of the robot to be an incorrect or heading error. If the value of the heading error is less than zero, then the robot turns left and vice versa if the value of the heading error is positive then the robot turns right as in the illustration in Fig. 8.

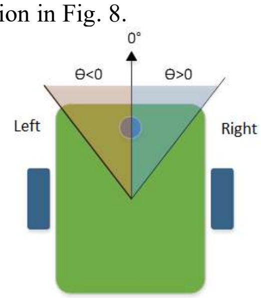


Fig. 8. Illustration of robot *heading*.

The difference in the value of heading robot with setpoint value is an *error* as in (19). Setpoint (SP) is a desired value or reference, present value (PV) is the value of sensor readings or inputs that are fed back by sensors to the system. Every time the program looping, the rotary encoder updates the *error* data and the previous error stored on another variable named *last_error*. The *error* and *last_error* are then processed with digital PID controls as in (18). After the PID value is obtained, it will be added or subtracted by the PWM Base value as presented in Fig. 9.

$$error = SP - PV \tag{19}$$

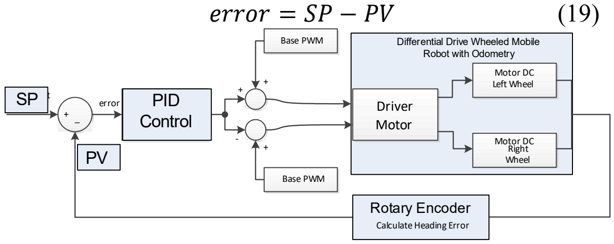


Fig. 9. PID control block diagram

IV. Results and Discussions

This work conducted the test by giving nine target points of odometry coordinates (0.40), (20.39), (38.60), (39.81), (58.80), (60.40), (39.0) and (20.0) for the robot in units of centimeter distance as shown in Fig. 10 uses graphical user interface (GUI) from Processing IDE.

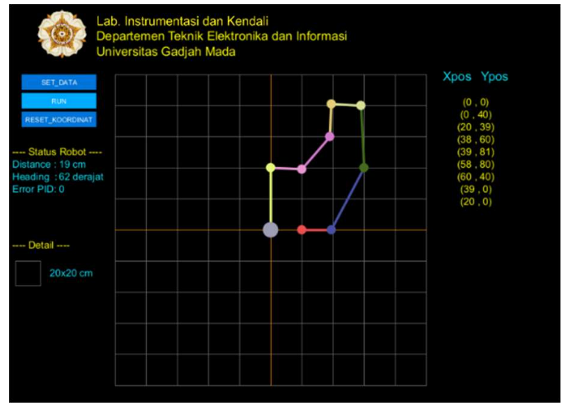


Fig. 10. The odometry testing.

The bearing (β) target value can be obtained from the reading of the rotary encoder found on the left and right wheels of the robot concerning (12-14). The error position value on the y-axis is denoted by e_y, and on the x-axis, it is denoted by e_x as presented in Fig. 11.

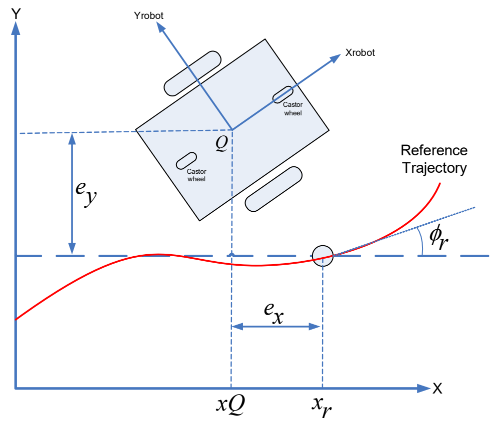


Fig. 11. The illustration of robot *heading error*.

The purpose of the PID control in this system is driving the robot to be able to move straight. It can be noticed from the value of the robot *heading error*, which is 0°. Before using PID, the test was carried out by giving a PWM value of 80 on the right motor and left motor. Fig. 12. shows heading data before the robot is tested using a PID controller.

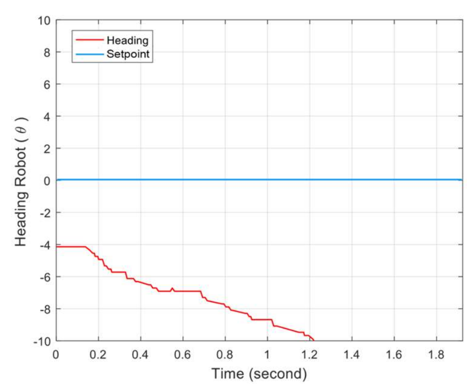
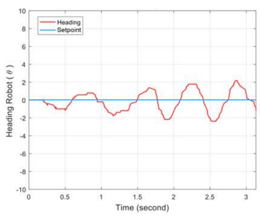
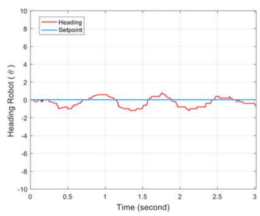
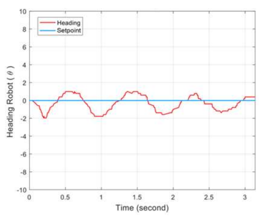
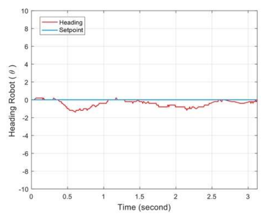
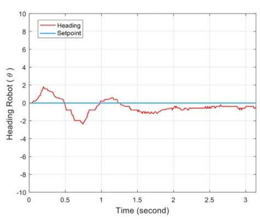
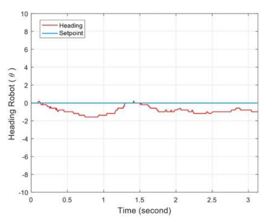


Fig. 12. *Heading* robot data without PID control.

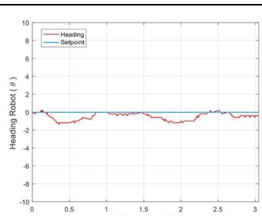
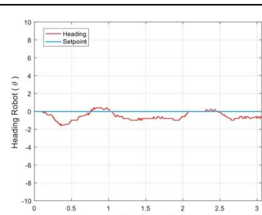
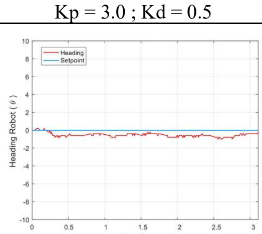
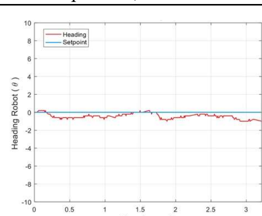
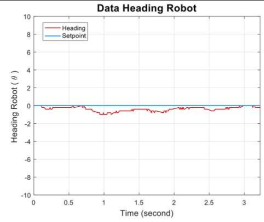
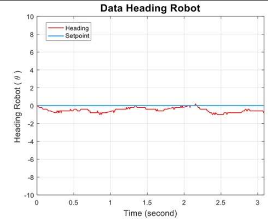
The graph above shows that the value of the robot heading is always negative during the test. It means that the robot tends to turn left. Table III addresses the results of the heading robot value after adding the proportional component that forms the PID controller. The experiment is carried out by running a robot that adjusts the straight track reference as far as 100 cm.

TABLE III
THE RESULTS OF THE PROPORTIONAL CONTROL TEST

 Kp = 6.0	 Kp = 5.0
 Kp = 4.0	 Kp = 3.0
 Kp = 2.5	 Kp = 2.0

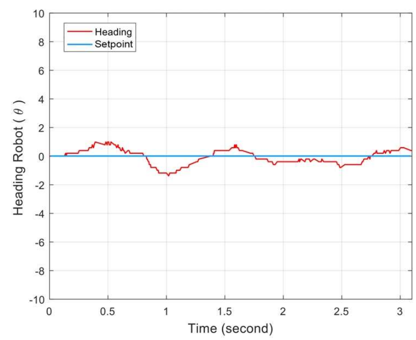
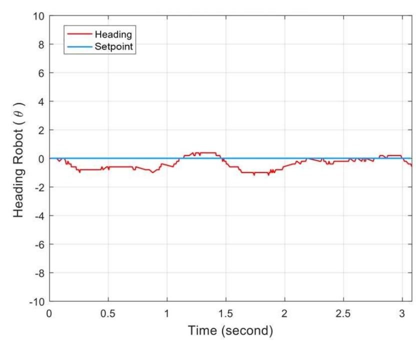
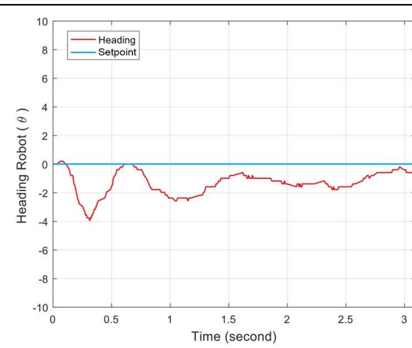
From the results of the proportional control test, the value of Kp 3.0 produces a heading value that is close to the setpoint (0o). There are still oscillations, but this is better than the other Kp values. To improve system response for the better, derivative control is necessary. The results of the proportional and derivative control tests are shown in Table IV.

TABEL IV
THE RESULTS OF THE PROPORTIONAL AND DERIVATIVE CONTROL TEST

 Kp = 3.0 ; Kd = 0.5	 Kp = 3.0 ; Kd = 1.0
 Kp = 3.0 ; Kd = 1.5	 Kp = 3.0 ; Kd = 2.0
 Kp = 3.0 ; Kd = 2.5	 Kp = 3.0 ; Kd = 3.0

The data shown in table IV explains that the addition of Kd makes the oscillations smaller, but still not able to get a system output that is close to the expected setpoint. Therefore, Integral control is applied to resolve the problem. The results of testing the PID control are displayed in Table V.

TABEL V
THE RESULTS OF THE PROPORTIONAL, INTEGRAL AND DERIVATIVE CONTROL TEST

PID Paramater	**System response**
Kp (3.0) Kd (2.5) Ki (0.0001)	
Kp (3.0) Kd (2.5) Ki (0.0003)	
Kp (3.0 Kd (2.5) Ki (0.0005)	

From the results of the PID control test, it can be inferred that the bigger the value of Kp produces a bigger oscillation. Choosing the smaller gain for Kp will be safer to the navigation system of the robot. Based on the experiment, Kp=3.0, Kd=2.5 and Ki=0.0003 is the most optimal option of trial-error tuning experiment. Even it seems that the system still has an error steady state, but it is small enough to be neglected. It does not look having bad performance through direct eye observation as depicted in Fig.13.

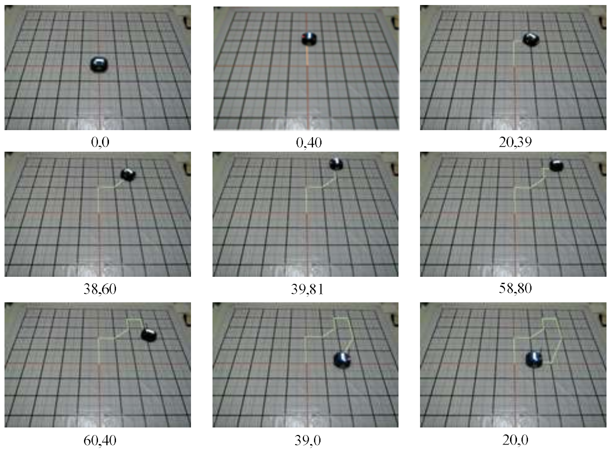


Fig. 13. Experimental result of robot *heading*.

V. Conclusion

This paper presents the performance evaluation of heading control on differential drive wheeled mobile robot with odometry for tracking problem. The test conducted in this paper used several target points to see the overall performance of the system. The results obtained are that PID controller is qualified to conduct heading control for the robot. The PID control parameters obtained from the tuning experiment are Kp = 3.0, Ki = 0.0003 and Kd = 2.5. The test results of this PID control have also published on the YouTube video channel at the address [13].

Acknowledgment

We firmly thank Department of Electrical and Informatics Engineering, Vocational College, Universitas Gadjah Mada and Department of Electrical Engineering, Universitas Negeri Semarang, for encouraging this work.

References

[1] B. Curuklu, G. Dodig-Crnkovic, and B. Akan, "Towards industrial robots with human-like moral responsibilities," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010, pp. 85–86.

[2] F. Kaiser, S. Islam, W. Imran, K. H. Khan, and K. M. A. Islam, "Line follower robot: Fabrication and accuracy measurement by data acquisition," in *2014 International Conference on Electrical Engineering and Information & Communication Technology*, 2014, pp. 1–6.

[3] R. V. Soans, Ranjith, A. Hegde, C. Singh, and A. Kumar, "Object tracking robot using adaptive color thresholding," in *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, 2017, pp. 790–793.

[4] M. Manel and B. Faouzi, "Trajectory tracking of two wheeled mobile robot," in *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, 2015, pp. 1–6.

[5] J. Osusky and J. Ciganek, "Trajectory tracking robust control for two wheels robot," in *2018 Cybernetics & Informatics (K&I)*, 2018, pp. 1–4.

[6] N. T. Truc, E.-H. Sun, Y.-M. Kim, and Y.-T. Kim, "Navigation method using fuzzy line tracking for the transportation robot," in *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, 2014, pp. 819–822.

[7] F. Fahmizal, M. Arrofiq, and A. Mayub, "Identifikasi Pemodelan Matematis Robot Wall Following," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 1, Mar. 2018.

[8] K. Ohno, T. Tsubouchi, B. Shigematsu, S. Maeyama, and S. Yuta, "Outdoor navigation of a mobile robot between buildings based on DGPS and odometry data fusion," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, pp. 1978–1984.

[9] J.-P. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot motion planning and control*, Berlin: Springer, 1998, pp. 1–53.

[10] Kok Seng Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proceedings of International Conference on Robotics and Automation*, 1997, vol. 4, pp. 2783–2788.

[11] J. Borenstein and Liqiang Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 869–880, 1996.

[12] K. Ogata, *Modern Control Engineering*, 5th ed. New Jersey: Prentice Hall, 2010.

[13]Video implementation [Online]. Available on : https://youtu.be/1vndGHznzyU