



Trajectory Tracking of a Four-Wheel Omni-Directional Mobile Robot using PID Control Method

Fizar Maulana^{1*}, Nundang Busaeri², Imam Taufiqurrahman³

¹²³Department of Electrical Engineering, Faculty of Engineering, Universitas Siliwangi, Indonesia

Correspondence E-mail: fizar.maulanaokta9@gmail.com

ABSTRACT

Omni-Directional Mobile Robot (OMR) offers superior mobility that allows it to move in any direction without the need to change orientation. However, for precision applications such as robotic competitions or industrial logistics, the main challenge is the ability to accurately track the trajectory. This study proposes a simulation of a Proportional-Integral-Derivative (PID) control system for OMR with a four-wheel configuration. The research methodology includes inverse kinematics modeling to translate the robot's speed into the speed of individual wheels, the design of PID controllers to minimize position errors, and its implementation in a MATLAB simulation environment. The test scenario is carried out by providing a series of reference points (waypoints) that the robot must follow. The simulation results showed that the designed PID control system was able to direct the robot to follow a specified trajectory with a high degree of accuracy. The speed analysis of each wheel also shows a dynamic response that corresponds to kinematic calculations. This study proves the feasibility of implementing PID controllers for trajectory tracking applications on four-wheel OMRs and can be the basis for development on physical platforms.

ARTICLE INFO

Article History:

Submitted/Received 3 Sep 2025

First Revised 12 Sep 2025

Accepted 28 Sep 2025

First Available online 1 Oct 2025

Publication Date 1 Oct 2025

Keyword:

Omni-Directional Robot,
PID Control,
Trajectory Tracking,
Inverse Kinematics,
MATLAB simulation.

1. INTRODUCTION

The development of robotics technology, especially mobile robots, has had a significant impact on various aspects of life. One type that attracts attention is the Omni-Directional Mobile Robot (OMR) [1][2]. OMRs utilize special omni-wheels with additional rollers [3][4], enabling them to move in any direction without changing the robot's body orientation, making them holonomic robots [5]. This superior maneuverability is beneficial in tight or complex environments [6][7]. OMRs come in configurations like three-wheel (120°) or four-wheel (90° or 45°), each affecting its kinematic model [8], [9]. This study focuses on a four-wheel configuration with an X-shaped (45°) arrangement, which offers good mechanical stability. OMR applications are diverse, ranging from industrial automation [10][11], healthcare, exploration, to robotic competitions.

In industrial applications, OMRs are used for the transport of materials within factories autonomously, while in military applications, these robots are used for reconnaissance missions to reduce risks to personnel [12][3]. In these various applications, the fundamental problems faced are communication and, most importantly, the positioning of the robot. Precisely controlling movement becomes very difficult because it has to adjust the speed of four DC motors simultaneously. The robot's ability to calculate the current position relative to the previous position, or dead reckoning, requires considerable effort [5]. The determination of this position relies heavily on input from several sensors, one of which is the rotary encoder, which specifically aims to obtain the rotational speed of each wheel [14][1].

To overcome positioning problems and achieve accurate trajectory tracking, a control system is needed. Some commonly used control methods include odometry, artificial neural network (NN), fuzzy logic, and Proportional-Integral-Derivative (PID) controllers [15][16][17]. The odometry method, which estimates position changes over time using data from motion sensors such as rotary encoders, is often implemented as feedback to improve precision [8][9].

PID controllers, although conventional, are still very widely used due to their ability to produce precise and fast control. In robotic applications, PID controllers are ideal for counteracting the decrease in DC motor speed due to load, so that the rotary speed can be kept constant as desired [18][16]. PID controllers work by minimizing the error values present in the system and can provide a fast response, making them ideal for controlling DC motor actuators [19][20][21].

This research focuses on the design and simulation of a PID control system for an omni-directional mobile robot (OMR), aiming for precise movement toward predefined waypoints. The study's main contribution is the integration of an accurate inverse kinematics model for a four-wheel configuration with a PID-based closed-loop control scheme to minimize position errors. The simulated OMR uses a four-wheel layout arranged at 45-degree angles, ensuring balanced motion dynamics. All modeling and analysis, including kinematic computation and controller performance evaluation, were conducted using MATLAB. The scope is limited to system design and validation within a simulation environment, serving as a preliminary step toward implementation on a physical platform.

2. METHODS

The steps that must be taken to achieve the goals of this research can be seen in **Figure 1**. below.

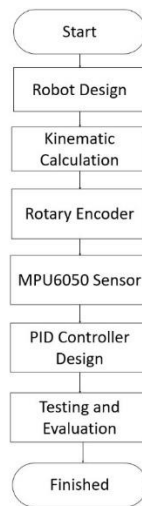


Figure 1. Research Flowchart

The research flowchart in Figure 1 illustrates the systematic stages of the robot design and development process. The research begins with defining the mechanical parameters of the mobile robot, such as wheel radius and wheelbase distance, to determine the kinematic relationships. Kinematic calculations are then performed to establish the mathematical model governing the robot's motion and trajectory control.

Next, the rotary encoder module is integrated to measure wheel rotation, generating pulse signals for displacement and speed feedback, which are essential for estimating position and velocity. The MPU6050 sensor is incorporated to provide complementary data on angular acceleration and orientation, enhancing motion estimation accuracy and stability analysis.

Subsequently, a PID (Proportional–Integral–Derivative) controller is designed and tuned to regulate the robot's motion dynamics. The controller processes feedback from the encoder and sensor data to minimize error between desired and actual trajectories, ensuring smooth movement. Finally, the system undergoes testing and evaluation to validate performance, responsiveness, and robustness.

Figure 2 shows a robot wheel that can move in any direction. An omni wheel is a wheel that is tucked between the wheels. This single wheel has a small wheel that functions to move in all directions. **Figure 2** shows the Omni wheel with 14 rollers and a width of 82 mm.



Figure 2. Omni-Wheel

Figure 3 is a schematic of a four-wheeled omni wheel. The linear speed of each wheel (V_i) is the result of the superposition of two components of motion, namely rotational motion and robotic translational motion.

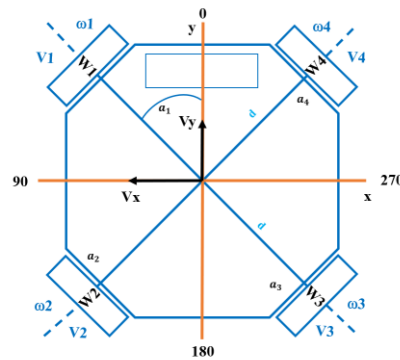


Figure 3. Schematic of the arrangement of the omni wheel with four wheels

The contribution of translational motion is obtained by projecting the velocity vector in the direction of the rotation of the wheel. Meanwhile, the contribution of rotational motion depends on the angular velocity $[x, y]$ of the robot θ and the distance from the center of the robot to the wheel R . This relationship is formulated with equation (1).

$$V_i = -\sin(a_i)x + \cos(a_i)y + R\theta \tag{1}$$

Furthermore, the linear velocity of a wheel V_i has a direct relationship with its angular velocity ω_i through the parameter r of the radius of the wheel, as shown in equation (2).

$$V_i = r\omega_i \tag{2}$$

By substituting equation (1) into equation (2), a reverse kinematics model for one wheel is obtained, as presented in equation (3).

$$\omega_i = \frac{1}{r}(-\sin(a_i)x + \cos(a_i)y + R\theta) \tag{3}$$

Equation (3) is arranged for each wheel with equation (4).

$$\begin{aligned} \omega_1 &= \frac{1}{r}(-\sin(a_1)x + \cos(a_1)y + R\theta) \\ \omega_2 &= \frac{1}{r}(-\sin(a_2)x + \cos(a_2)y + R\theta) \\ \omega_3 &= \frac{1}{r}(-\sin(a_3)x + \cos(a_3)y + R\theta) \\ \omega_4 &= \frac{1}{r}(-\sin(a_4)x + \cos(a_4)y + R\theta) \end{aligned} \tag{4}$$

The specifications of the Omni robot are shown in **Table 1**.

Table 1. Specification of Omni Robot

No.	Symbol	Information
1	x and y	The coordinates of the robot's position from the global map reference.

2	a_1, a_2, a_3, a_4	The angle that is shaped between the wheel and the axis y robot.
3	$W1, W2, W3, W4$	Symbols for the wheel in each position.
4	d [meter]	Distance (meters) between the wheel and the center of the wheel.
5	v_1, v_2, v_3, v_4 [m/s]	Linear speed of the robot's wheels.
6	$\omega_1, \omega_2, \omega_3, \omega_4$ [rad/s]	Wheel angle speed.
7	V_x	Speed against the robot's x-axis.
8	V_y	Speed against the robot's y axis.

To represent the system as a whole, equations (4) are arranged into the form of equation matrices (5).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(a_1)x + \cos(a_1)y + R \\ -\sin(a_2)x + \cos(a_2)y + R \\ -\sin(a_3)x + \cos(a_3)y + R \\ -\sin(a_4)x + \cos(a_4)y + R \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (5)$$

By substituting the specific angular values ($a_1, a_2, a_3, dan a_4$) of the equation (5), the robot configuration used in this study obtained the equation (6).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(45^\circ)x + \cos(45^\circ)y + R \\ -\sin(135^\circ)x + \cos(135^\circ)y + R \\ -\sin(225^\circ)x + \cos(225^\circ)y + R \\ -\sin(315^\circ)x + \cos(315^\circ)y + R \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (6)$$

Equation (6) is at the heart of the motion control algorithm, which allows precise calculation of the rotational speed of the motor to achieve an accurate goal. For computational implementations, the angular value in degrees is converted into radian units in the equation (7).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin\left(\frac{\pi}{4}\right)x + \cos\left(\frac{\pi}{4}\right)y + R \\ -\sin\left(\frac{3\pi}{4}\right)x + \cos\left(\frac{3\pi}{4}\right)y + R \\ -\sin\left(\frac{5\pi}{4}\right)x + \cos\left(\frac{5\pi}{4}\right)y + R \\ -\sin\left(\frac{7\pi}{4}\right)x + \cos\left(\frac{7\pi}{4}\right)y + R \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (7)$$

The PID controller integrates three control actions proportional, integral, and derivative which mathematically work together to reduce or eliminate errors within the control system. Equation (8) represents the PID control output signal, expressed as a Pulse Width Modulation (PWM) signal, which is used to regulate the motor's rotational speed.

$$u = Kp(e)t + Ki \int_0^t e(t)dt + Kd \frac{de(t)}{dt} \quad (8)$$

In this system, u represents the control signal, while Kp , Ki , and Kd denote the proportional, integral, and derivative constants, respectively. The term $e(t)$ refers to the error signal, which is obtained by comparing the desired reference value with the actual sensor readings. This approach utilizes encoder data to estimate the system's position parameters x , y , and θ allowing the controller to correct deviations and compensate for positional errors that may occur during operation.

The next step is to determine how to organize the production of the robot. Research flow chart in **Figure 4.** shows the steps involved in the manufacture of the robot.

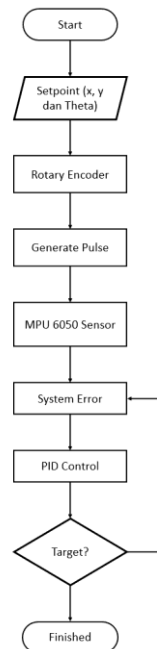


Figure 4. Study Flowchart

The set points at the x , y , and θ coordinates are the starting points of the flowchart from Figure 4. The pulse given is then made using sensor readings and the position to be traversed, so that it matches the trajectory of the drive. There is a compatibility ratio with a particular trajectory. A system error is a reading sign that has a set point. Enter the system fault into the PID controller to adjust the generator to match the desired target and make the error zero.

3. RESULTS AND DISCUSSION

In designing this robot, the next step is to create a 3D design that serves as an initial visualization before the actual design is made, when all the required dimensions have been determined. The robot's design drawings aim to know the parameters to make the programming process easier.

Figure 5. is a Robot design designed to know the size of the frame. From Figure 5, the wheelbase distance and base radius of the OMR robot can be determined.



Figure 5. 3D design of mobile robots

After knowing the image, the next step is to determine the image of the stage series. The specifications of the robot frame can be seen from the **Table 2**.

Table 2. Specification of Omni Robot Frame

No.	Information	Size (mm)
1	Wheelbase to midpoint of the body	150
2	Length Diameter	300
3	Diameter Roda	82

Table 2. shows the frame specifications of the omni robot. It further describes the flow of the omni robot control system. The process starts from a reference that determines the target trajectory and this data is processed through inverse kinematics to calculate the required wheel angular speed. Furthermore, the PID Controller adjusts the speed of each motor according to the reference speed and feedback from the rotary encoder can be seen in **Figure 6**.

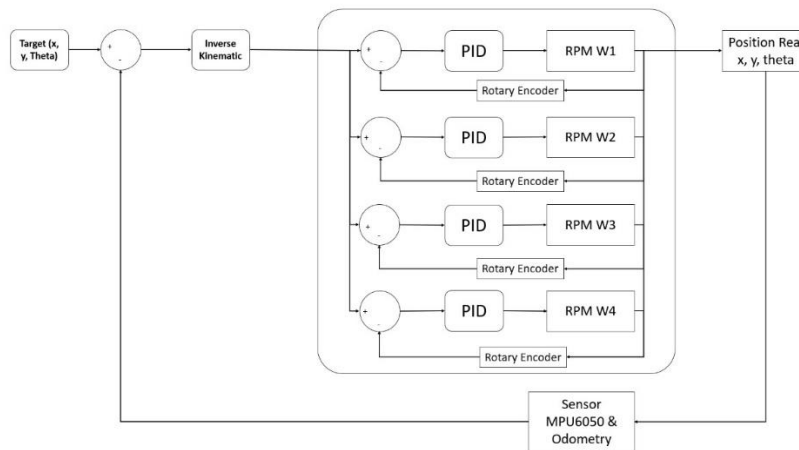


Figure 6. Diagram control

A mathematical model of a four-wheeled omni-wheel mobile robot is simulated using Simulink's MATLAB with the physical parameters shown in Table 2. These parameter values are adapted from previous research that applied PID control to omni-wheel robot systems with similar configurations to support the accuracy of track movement.

The PID controller is designed to regulate the robot's translation and rotation speed to be able to follow a predetermined reference trajectory. The simulation was carried out without

considering external interference, as the research was focused on simulated system performance analysis. **Figure 7, Figure 8, and Figure 9** show the Simulink model of the omni-wheel robot, wheel speed calculations through inverse kinematics, as well as translational and rotational movement results that demonstrate the effectiveness of PID control in maintaining track stability and precision.

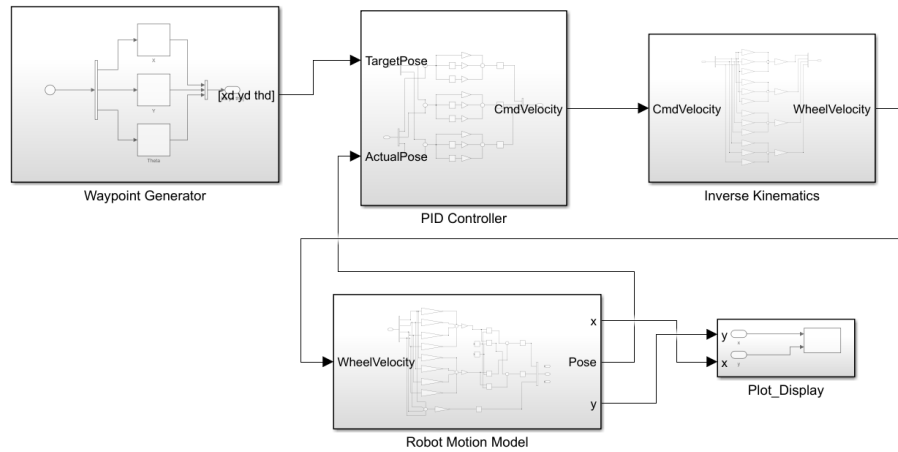


Figure 7. Block simulink model trajectory tracking robot omni-wheel

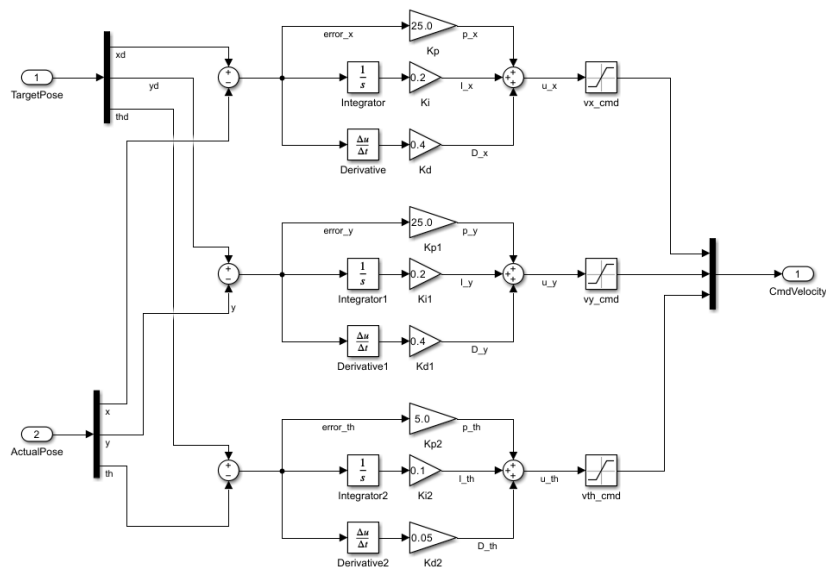


Figure 8. Block simulink control PID robot omni wheel

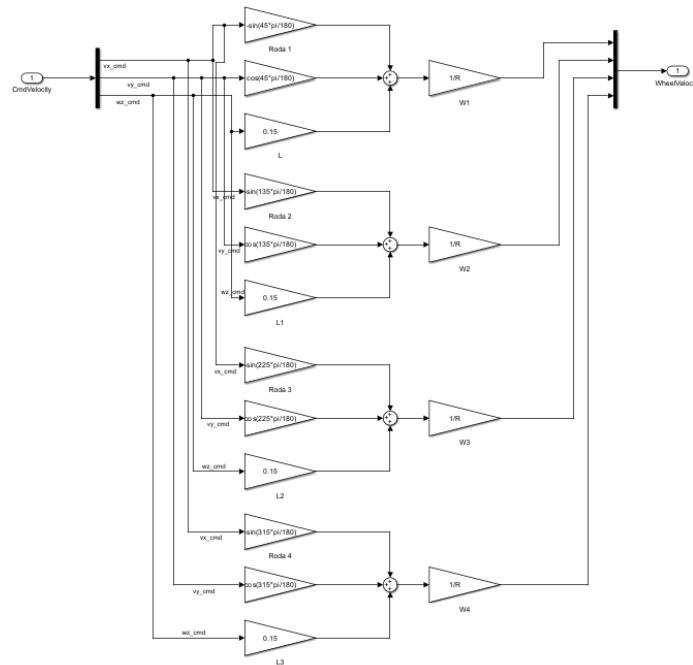


Figure 9. Block simulink inverse kinematic omni-wheel robot

The results of the program simulation using a square (or rectangle) trajectory, with multiple waypoints defined for the Omni Mobile Robot (OMR), are shown in **Figure 10**. The test is carried out by providing reference points that the robot must reach in order. The directional points are expressed in coordinates (x, y, θ) as $[0, 0, 0; 3.0, 0.0, \pi/4; 3.0, 2.0, \pi/2; 0.0, 2.0, \pi/2]$ in time units $[0; 3; 6; 9; 12; 15]$. Figure 13 shows the results of the kinematics of the robot's movement and control response. The initial condition is represented by the first point $[0, 0, 0]$. The second point $[3.0, 0.0, \pi/4]$ is reached in about 3 seconds, indicating the robot's smooth movement along the x-axis before turning in the y-direction. The subsequent waypoints on $[3.0, 2.0, \pi/2]$ and $[0.0, 2.0, \pi/2]$ depict the robot successfully following the planned square trajectory, although slight deviation may occur due to PID parameter tuning and kinematic interaction effects.

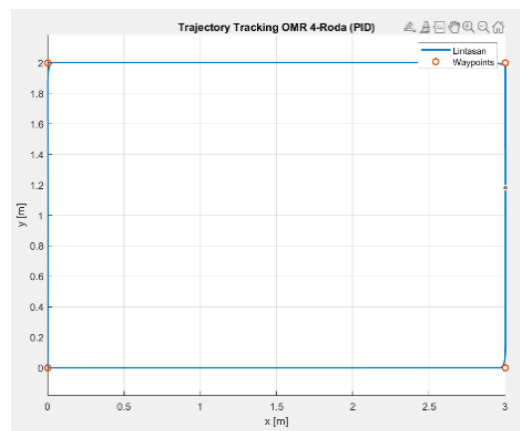


Figure 10. Trajectory Tracking on a square trajectory

Figure 11 continues the experiment from Figure 8 by changing the shape of the trajectory to a triangular path, while maintaining the same time interval $[0; 3; 6; 9; 12; 15]$. The waypoint points are set at position (x, y, θ) $[0.0, 0.0, 0.0; 3.0, 0.0, \pi/4; 1.5, 2.0, \pi/2]$. The robot starts at $[0,0,0]$, moves along the x-axis, then turns upwards towards the y direction to complete the

triangular trajectory. The simulation results showed that the PID controller remained stable and performed consistent track tracking across various path shapes, demonstrating good control system adaptability.

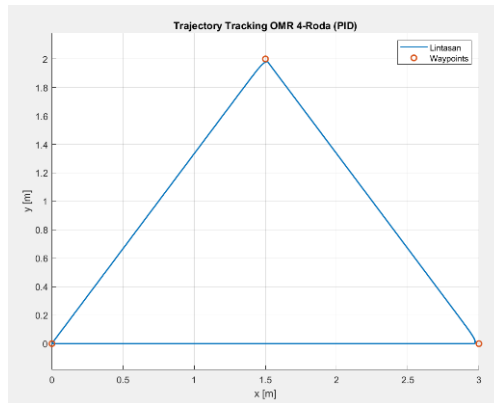


Figure 11. Trajectory Tracking on a square trajectory

Figure 12 and **Figure 13** show a comparison of the angular speed response of the omni wheel on square and triangular trajectories. Each image shows the tracking performance between the actual wheel angle speed ($\omega_1 - \omega_4$, full line) and its reference value ($\omega_{1,ref} - \omega_{4,ref}$, dotted line). The results show that the PID controller is able to create each wheel in a way that is consistent with the stability and synchronization of the system. In a square trajectory, the change in speed is sharper due to the greater weight, whereas in a triangular trajectory, the change in speed is smoother. This shows that the control algorithm can effectively adapt to different types of trajectories.

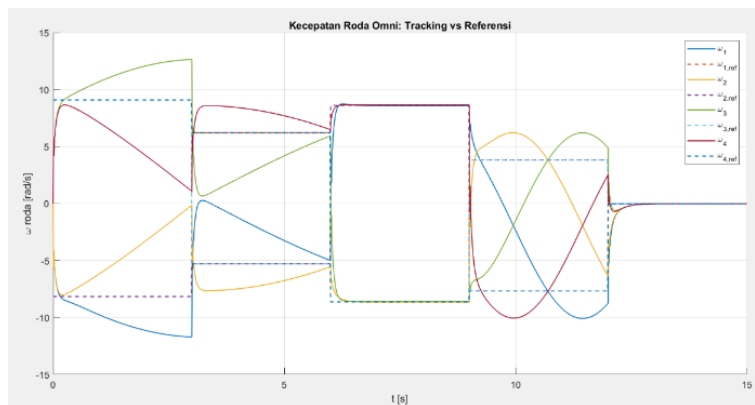


Figure 12. Wheel angular speed response on a square track

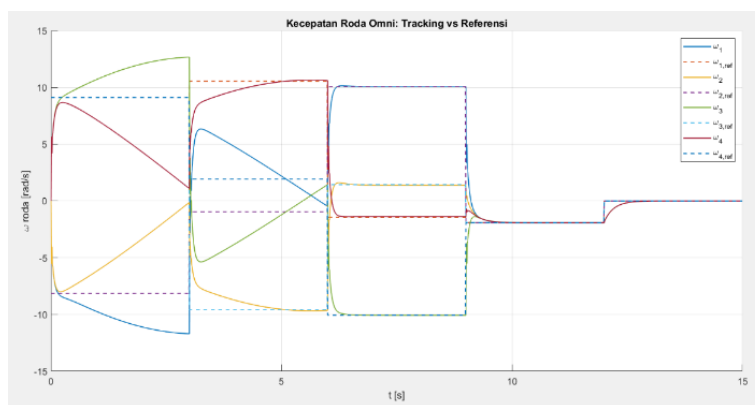


Figure 13. Wheel angular speed response on triangular trajectories

4. CONCLUSION

This study successfully demonstrated the simulation of a four-wheeled omni-directional mobile robot (OMR) controlled using a Proportional–Integral–Derivative (PID) method. The research encompassed several stages, including mechanical design, component selection, electrical system configuration, inverse kinematics modeling, and PID controller implementation for position control. Simulation results conducted over 15 seconds with various waypoints showed that the designed PID controller effectively guided the robot to follow both square and triangular trajectories with high accuracy and stability. The OMR was able to reach each waypoint precisely while maintaining smooth rotational motion, indicating that the PID control strategy successfully minimized position errors and ensured consistent trajectory tracking. These findings confirm the feasibility of applying PID control to four-wheeled omni-directional mobile robots and provide a strong foundation for future implementation on physical platforms where environmental disturbances and sensor noise can be further analyzed.

5. ACKNOWLEDGMENT

This study was funded by Universitas Siliwangi. The research resources and facilities provided by the Electrical Engineering department were instrumental in this study.

6. AUTHORS' NOTE

The author declares that there is no conflict of interest related to the publication of this article. The author confirms that the paper is free of plagiarism.

7. REFERENCES

- [1] H. Purnata, S. Ramadan, M. A. Hidayat, and I. Maulana, "PID Control Schematic Design for Omni-directional wheel mobile robot Cilacap State of Polytechnic," *J. Jartel J. Jar. Telekomun.*, vol. 12, no. 2, pp. 89–94, 2022, doi: 10.33795/jartel.v12i2.322.
- [2] P. Anggraeni, N. J. Ramadhan, and M. Huda, *Trajectory Tracking Control of Three Wheeled Omnidirectional Mobile Robot Using Proportional Controller*, vol. 2023. Atlantis Press International BV, 2024. doi: 10.2991/978-94-6463-364-1_51.
- [3] M. A. Shamseldin, "Real-Time Inverse Dynamic Deep Neural Network Tracking Control for Delta Robot Based on a COVID-19 Optimization," *J. Robot. Control*, vol. 4, no. 5, pp. 643–649, 2023, doi: 10.18196/jrc.v4i5.18865.
- [4] T. T. Pham, M. T. Le, and C. N. Nguyen, "Omnidirectional Mobile Robot Trajectory Tracking Control with Diversity of Inputs," *Int. J. Mech. Eng. Robot. Res.*, vol. 10, no. 11, pp. 639–644, 2021, doi: 10.18178/ijmerr.10.11.639-644.
- [5] K. Watanabe, Y. Shiraishi, S. G. Tzafestas, J. Tang, and T. Fukuda, "Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots," *J. Intell. Robot. Syst. Theory Appl.*, vol. 22, no. 3–4, pp. 315–330, 1998, doi: 10.1023/a:1008048307352.
- [6] Imam Taufiqurrahman, Andri Ulus Rahayu, and Permana, "Desain Kendali Kecepatan Motor DC Pada Mobile Robot Mecanum Wheels," vol. 05, no. 2, pp. 85–91, 2024.
- [7] Supriadi, A. Wajiansyah, M. Zainuddin, and A. B. W. Putra, "Optimization of Proportional Integral Derivative Controller for Omni Robot Wheel Drive by Using Integrator Wind-up Reduction Based on Arduino Nano," *J. Robot. Control*, vol. 5, no. 6, pp. 1690–1701, 2024, doi: 10.18196/jrc.v5i6.21807.
- [8] R. L. Putra, H. K. Safitri, and I. Siradjuddin, "Pemodelan dan Analisis Sistem Kontrol Kinematik," *J. ELKOLIND*, vol. 2, no. 1, pp. 80–87, 2015.
- [9] D. U. Rijalusalam and I. Iswanto, "Implementation kinematics modeling and odometry of four omni wheel mobile robot on the trajectory planning and motion control based microcontroller," *J. Robot. Control*, vol. 2, no. 5, pp. 448–455, 2021, doi:

- 10.18196/jrc.25121.
- [10] F. Fahmizal, D. U. Rijalussalam, M. Budiyanto, and A. Mayub, "Trajectory Tracking pada Robot Omni dengan Metode Odometry," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 8, no. 1, p. 35, 2019, doi: 10.22146/jnteti.v8i1.488.
- [11] G. W. Kurniawan, N. Setyawan, and E. A. Hakim, "PID Trajectory Tracking Control 4 Omni-Wheel Robot," *SinarFe7*, vol. 2, no. 1, pp. 345–350, 2019.
- [12] H. M. Alwan, V. A. Nikolaevic, S. F. Hasan, and K. O. Vladmerovna, "Kinematic and Dynamic Modeling Based on Trajectory Tracking Control of Mobile Robot with Mecanum Wheels," *Int. J. Technol.*, vol. 15, no. 5, pp. 1473–1486, 2024, doi: 10.14716/ijtech.v15i5.6908.
- [13] J. Kurniawan, M. Syafaat, K. Kasiyanto, D. Widiatmoko, R. Maulana, and Z. N. Putra, "Omni wheel robot movement exploration using a control system for military surveillance with integrated sensor," *TEKNOSAINS J. Sains, Teknol. dan Inform.*, vol. 12, no. 1, pp. 110–121, 2025, doi: 10.37373/tekno.v12i1.1233.
- [14] M. Khairudin, R. Refalda, S. Yatmono, H. S. Pramono, A. K. Triatmaja, and A. Shah, "The mobile robot control in obstacle avoidance using fuzzy logic controller," *Indones. J. Sci. Technol.*, vol. 5, no. 3, pp. 334–351, 2020, doi: 10.17509/ijost.v5i3.24889.
- [15] A. K. Kazi, Maria Andleeb Siddiqui, M. A. Zaki, and A. S. Khan, "Fuzzy Logic and Neural network-based self-tuned PID controller of Quadcopter," *Pakistan J. Eng. Technol. Sci.*, vol. 12, no. 1, pp. 49–63, 2024, doi: 10.22555/pjets.v12i1.1085.
- [16] M. Alfiyan and R. D. Puriyanto, "Mecanum 4 Omni Wheel Directional Robot Design System Using PID Method," *J. Fuzzy Syst. Control*, vol. 1, no. 1, pp. 6–13, 2023, doi: 10.59247/jfsc.v1i1.27.
- [17] N. H. Thai, T. T. K. Ly, and L. Q. Dzung, "Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller," *Bull. Electr. Eng. Informatics*, vol. 11, no. 4, pp. 1902–1910, 2022, doi: 10.11591/eei.v11i4.3712.
- [18] T. Zhao, P. Qin, and Y. Zhong, "Trajectory Tracking Control Method for Omnidirectional Mobile Robot Based on Self-Organizing Fuzzy Neural Network and Preview Strategy," *Entropy*, vol. 25, no. 2, 2023, doi: 10.3390/e25020248.
- [19] D. Ounnas, D. Guiza, Y. Soufi, R. Dhaouadi, and A. Bouden, "Design and Implementation of a Digital PID Controller for DC-DC Buck Converter," *Proc. - 2019 1st Int. Conf. Sustain. Renew. Energy Syst. Appl. ICSRESA 2019*, vol. 2, no. 2, pp. 1–20, 2019, doi: 10.1109/ICSRESA49121.2019.9182430.
- [20] M. Matli, A. Albayrak, and R. Bayir, "Development of a Mecanum-Wheeled Mobile Robot for Dynamic- and Static-Obstacle Avoidance Based on Laser Range Sensor," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 20, no. 3, pp. 188–200, 2020, doi: 10.5391/IJFIS.2020.20.3.188.
- [21] N. Holis, M. S. Zuhrie, E. Endryansyah, and N. Nurhayati, "Sistem Positioning Robot Four-Omnidirectional Wheels Menggunakan MPU6050 dan Encoder Berbasis PD-Fuzzy Controller," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 2, pp. 164–171, 2023, doi: 10.37905/jjee.v5i2.20138.