ISSN 2355-3286

Implementation of Trajectory Tracking on Mobile Robot Differential Drive

Imam Taufiqurrahman[1], Eko Sujarwanto[2], Andri Ulus Rahayu[3],
Mira Riski Aldiani[4], Sayyid Qhutub Abdul Hakim[5]
[1,3,4,5] Faculty of Engineering, Siliwangi University, Tasikmalaya, Indonesia
[2] Physics Education, Universitas Siliwangi, Tasikmalaya, Indonesia.
[1]imamtaufiqurrahman@unsil.ac.id

Accepted on October 02, 2025
Approved on December 28, 2025

***Abstract*—This study discusses the implementation of trajectory tracking on a differential drive mobile robot using the odometry method. The system was designed by utilizing rotary encoders to estimate the robot's position and a proportional controller to regulate movement toward the target point. The research employs an Arduino microcontroller as the processing unit, integrated with L298N motor drivers and DC motors to achieve autonomous navigation capabilities. Experiments were carried out on multi-point trajectories, namely three-way and four-way paths, under two different surface conditions: flat and textured fields. Each trajectory was analyzed through systematic measurement of Mean Absolute Trajectory Error (MATE) to quantify the accuracy of the tracking system. The proportional control constant (Kp) was determined through iterative testing, with Kp=4 identified as the optimal value that enables the robot to reach target coordinates while maintaining velocity stability. The results showed that the robot was able to follow the predetermined path with a relatively high level of accuracy, especially on flat surfaces, achieving a MATE of 0.0661 for three-point trajectories and 0.0561 for four-point trajectories. However, on textured paths, the accuracy decreased significantly, with MATE values increasing to 0.2065 and 0.2778 respectively, due to wheel slip and disturbances in encoder readings. PWM analysis revealed that textured surfaces required 15-23% higher power consumption and resulted in substantial signal fluctuations (±25-35 PWM units) compared to flat surfaces (±10-15 PWM units). The comparison between both conditions emphasizes that surface characteristics have a significant influence on the performance of odometry-based trajectory tracking. This research contributes to the understanding of environmental factors affecting autonomous mobile robot navigation and provides practical insights for implementing trajectory tracking systems in real-world applications where surface conditions may vary.**

Index Terms — Mobile robot, differential drive, odometry, trajectory tracking, proportional control

I. INTRODUCTION

Mobile robots are a type of autonomous system designed to move independently across floors or specific pathways through the use of wheel actuators and motion control algorithms [1]. Among various types of wheeled robots, the differential drive mobile robot is one of the most widely adopted configurations due to its simple mechanical structure, high maneuverability, and ease of control. This robot operates using two independently driven wheels located on the right and left sides, allowing it to move forward, backward, turn, and even rotate in place through variations in the rotational speed of each wheel. Such flexibility makes differential drive robots suitable for numerous applications in industry, logistics, exploration, and service robotics, where precise movement and adaptability are crucial requirements [2][3][4].

The capability of a mobile robot to follow a predetermined path with high precision is a fundamental aspect of autonomous navigation systems [5]. This ability, known as trajectory tracking, involves guiding the robot's movement along a defined route or toward a sequence of target coordinates while continuously minimizing position and orientation errors. Trajectory tracking plays a key role in many robotic applications, including automated material transport, warehouse management, and mapping or surveillance tasks, where accuracy and reliability of motion strongly influence overall performance [6][7].

To achieve accurate trajectory tracking, one essential element is the robot's position estimation method. A widely used technique for this purpose is odometry, which estimates the robot's relative position by calculating the displacement of its wheels using rotary encoder data [8]. Odometry provides a simple yet effective way to track motion in real time without relying on external sensors. The distance traveled by each wheel is calculated based on encoder pulses, which are then converted into translational and rotational movements of the robot through differential kinematic equations [9]. However, despite its practicality, odometry is sensitive to cumulative errors, particularly when wheel slip, surface irregularities, or mechanical misalignments occur during motion.

In order to compensate for these errors and maintain accurate motion toward the target trajectory, a suitable control strategy must be implemented. One of the most fundamental and widely used approaches is the Proportional (P) control method, which adjusts the robot's motor speed proportionally to the detected error

between the desired trajectory and the actual position [10]. Although relatively simple, proportional control can produce a stable and responsive movement when properly tuned, especially for low-speed navigation tasks and short-range path following [11][12]. The tuning of the proportional gain directly affects the system's responsiveness and stability—values that are too small cause sluggish motion, while excessively high values can induce oscillations and instability [13].

Environmental and surface conditions also play a critical role in determining the performance of odometry-based trajectory tracking [14]. When operating on flat, untextured surfaces, wheel movement tends to be uniform, resulting in precise encoder readings and accurate position estimation. Conversely, on rough or textured surfaces, wheel slip and vibration introduce disturbances that cause deviations in encoder feedback, leading to cumulative position and orientation errors [15][16]. These factors highlight the importance of evaluating robot performance under different physical conditions to understand how terrain characteristics influence tracking accuracy and control response.

Based on these considerations, this study aims to implement and evaluate an odometry-based trajectory tracking system on a differential drive mobile robot using proportional control. The research focuses on comparing the robot's movement accuracy across two surface conditions—flat (untextured) and rough (textured)—to analyze how terrain characteristics affect odometry readings and trajectory stability. Experiments were conducted using multi-point paths, specifically three-point and four-point trajectories, to observe variations in tracking accuracy and system response. Through these analyses, this study seeks to provide insights into the practical limitations of odometry in real-world environments and offer references for improving trajectory tracking performance in future mobile robot applications.

II. METHODOLOGY

In this study, a differential drive mobile robot was designed using an Arduino microcontroller as the main processing unit to implement an odometry-based trajectory tracking system. Figure 1 illustrates the overall research flowchart, presenting the systematic workflow starting from literature review, system design, component integration, and experimental testing. This flowchart provides a clear visualization of how the trajectory tracking system is developed, implemented, and validated throughout the study.

To evaluate the influence of environmental conditions on tracking performance, the experiments were conducted under two different surface conditions, namely untextured and textured surfaces. The untextured surface consists of a flat and smoothly sanded wooden board, providing uniform wheel–ground contact and minimal friction disturbance. In contrast, the textured surface is constructed from textured concrete wall material with irregular roughness, resulting in non-uniform friction and increased wheel–ground interaction variability. This textured condition introduces intermittent wheel slip and mechanical vibrations during robot motion, which directly affect encoder readings and odometry accuracy. The selection of these two surface types enables a controlled comparison of trajectory tracking performance under ideal and non-ideal traction conditions.

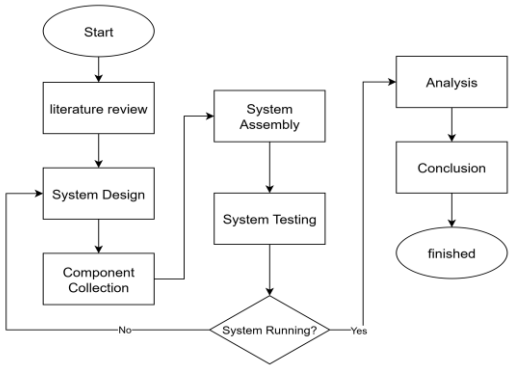

Fig. 1. System workflow

A. Literature Review

The research began with a literature study, system design, component collection, system assembly, and system testing. If the system testing met the desired targets, it proceeded to analysis, conclusion, and completion. However, if the system testing did not meet the expected targets, the process returned to system planning and retesting. This process was repeated until the target objectives were achieved, and then it was completed.

B. System Design

The wiring diagram in this study shows the integration of the main components of an Arduino-based mobile robot system. The hardware integration of the differential-drive mobile robot is shown in Figure 2, which presents the complete circuit diagram of the system. The diagram illustrates the connection between the Arduino microcontroller, motor driver, DC motors, and rotary encoders, forming a closed-loop feedback structure.

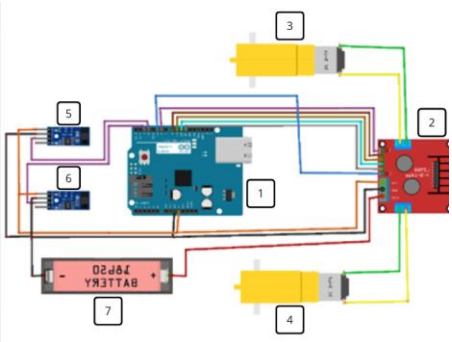

Fig. 2. Circuit diagram system

The Arduino microcontroller (1) functions as a control center that processes input data and regulates

control signal output. The L298N motor driver (2) is used to control the direction and speed of two DC motors, namely the right motor (3) and the left motor (4). To obtain real-time information on wheel position and speed, a right encoder sensor (5) and a left encoder sensor (6) are used, which are connected directly to the Arduino as closed-loop control system feedback. The entire circuit is powered by an 18650 battery (7) that serves as the main power source. With this configuration, the system is able to control the robot's movements more accurately by utilizing speed feedback from the encoder sensors. The relationship between the main components of the robot system is visualized in Figure 3, which provides the system block diagram. This figure highlights the data flow from the encoder sensors to the Arduino controller and the control signals sent to the motor driver for regulating the robot's movement.

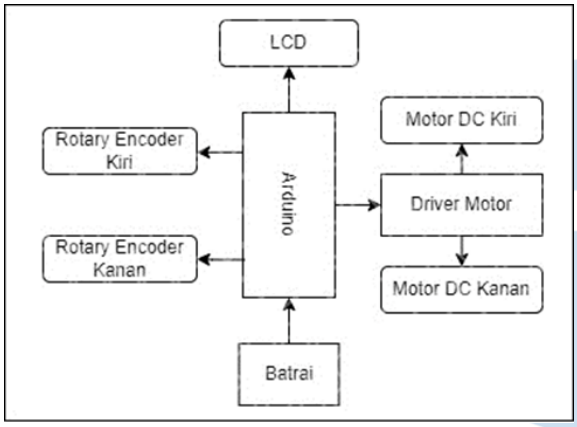

Fig. 3. Diagram system

The block diagram in this study shows the relationship between the main components used in the Arduino-based mobile robot system. Arduino functions as a control center that receives input data from the left and right rotary encoders, which are used to read the speed and number of wheel revolutions. The data is then processed by Arduino to regulate the control signals sent to the motor driver, which then controls the left DC motor and right DC motor as needed. In addition, the Arduino is also connected to an LCD to display important information such as speed or robot status in real-time, and receives power from a battery as its main energy source. This structure allows the robot system to work with a closed-loop control mechanism through the use of feedback from the encoder sensor to improve the accuracy of the robot's movements. The closed-loop control mechanism used in this study is illustrated in Figure 4. This diagram demonstrates how the setpoint, motor commands, encoder feedback, and position estimation interact continuously to ensure the robot follows the desired trajectory accurately using odometry-based feedback.

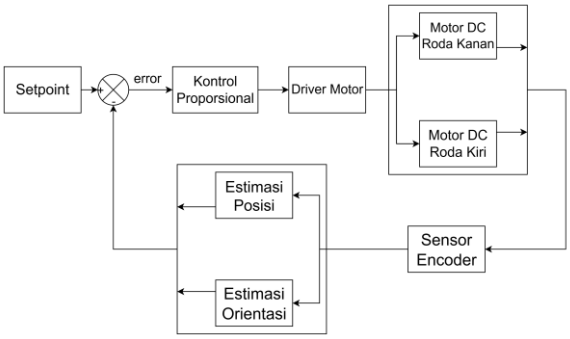

Fig. 4. Closed-loop system

Figure 4 illustrates the block diagram of a closed-loop control system for an odometry-based differential drive mobile robot. The system begins with a setpoint defined as the desired destination coordinates, which are compared with the estimated position and orientation of the robot obtained through odometry to generate a position error. This error is processed by a proportional controller to produce a control signal in the form of PWM commands, which are then sent to the motor driver. In this system, the speed control of the DC motors at the actuator level is implemented in an open-loop manner, where the PWM signals directly regulate the speed and direction of motor rotation without motor speed feedback. The motor driver controls the speed and rotation direction of the left and right DC motors. Motor motion is monitored using encoder sensors that generate pulse data to estimate the robot's position and orientation through odometry calculations. This estimated information is fed back into the control system at the position level, enabling real-time correction of position and orientation errors. Through this control mechanism, the system is able to improve the accuracy of the robot's movement toward the specified target coordinates, despite the use of open-loop motor speed control.

C. Kinematics of Differential Drive Mobile Robot

The differential drive mobile robot operates using two independently controlled wheels positioned on opposite sides of the robot chassis. The kinematic model describes the relationship between wheel velocities and the robot's overall motion in the global coordinate frame.

The robot's configuration can be represented by the state vector $\mathbf{q} = [x_Q, y_Q, \varphi]^T$, where (x_Q, y_Q) denotes the position of point Q the midpoint of the wheel axis and θ represents the robot's orientation (heading angle) with respect to the global reference frame. The kinematic model of the differential-drive mobile robot is supported by the illustration shown in Figure 5. This figure provides a geometric representation of the robot's coordinate frame, wheel configuration, and orientation angle, which form the foundation for the differential-drive kinematic equations.

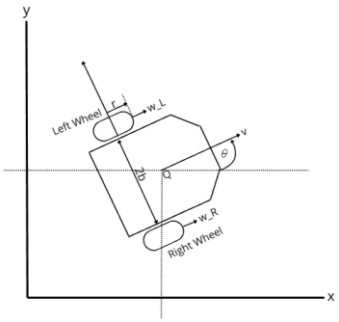

Fig. 5. Kinematic model of a differential drive mobile robot

The fundamental kinematic equations relating the robot's linear velocity (v) and angular velocity (ω) to its positional and angular changes are expressed as:.

$$\dot{x}_Q = v\cos(\theta) \quad (1)$$

$$\dot{y}_Q = v\sin(\theta) \quad (2)$$

$$\dot{\theta} = \omega \quad (3)$$

Where the linear and angular velocities are determined by the individual wheel velocities according to:

$$v = \frac{r(\omega R + \omega L)}{2} \quad (4)$$

$$\omega = \frac{r(\omega\_R - \omega\_L)}{(2b)} \quad (5)$$

In these equations, r represents the wheel radius, ω_R and ω_L are the angular velocities of the right and left wheels respectively, and 2b is the wheelbase—the distance between the two drive wheels measured along the axle.

By substituting equations (4) and (5) into the kinematic model, the complete differential drive kinematics can be expressed as:

$$\dot{x}_Q = \left(\frac{r}{2}\right)(\omega_R + \omega_L)\cos(\theta) \quad (6)$$

$$\dot{y}_Q = \left(\frac{r}{2}\right)(\omega_R + \omega_L)\sin(\theta) \quad (7)$$

$$\theta = \frac{r(\omega\_R - \omega\_L)}{(2b)} \quad (8)$$

These equations demonstrate that the robot's motion is controlled through differential wheel velocities: equal velocities produce straight-line motion, while velocity differences generate rotation. This under-actuated system has three degrees of freedom (x, y, θ) controlled by only two independent inputs (ω_R, ω_L), which necessitates sophisticated control strategies for trajectory tracking.

D. Odometry-Based Position Estimation

Odometry is a technique for estimating the robot's position and orientation by measuring the displacement of its wheels over time. This method utilizes rotary encoder sensors mounted on each wheel to count discrete pulses corresponding to incremental wheel rotation. The accumulated pulse data is then converted into distance traveled and orientation changes relative to an initial reference position.

The foundation of odometry lies in converting encoder pulses into physical distances. The wheel circumference is first calculated using equation (9), where r is the wheel radius. The encoder resolution defines the relationship between pulses and distance as shown in equation (10). This conversion factor enables the calculation of distance traveled by each wheel based on the number of pulses detected, as expressed in equations (11) and (12).

$$wheel\_circumference = 2\pi r \quad (9)$$

$$pulse\_per\_mm = \frac{encoder\_resolution}{wheel\_circumference} \quad (10)$$

$$distance\_right = \frac{pulse\_encoderRight}{pulse\_per\_mm} \quad (11)$$

$$distance\_left = \frac{pulse\_encL}{pulse\_per\_mm} \quad (12)$$

Once the individual wheel distances are obtained, the robot's forward displacement and change in orientation can be computed. Equation (13) calculates the average distance traveled by taking the mean of both wheel displacements, providing the linear displacement along the robot's heading direction. Meanwhile, equation (14) determines the angular displacement (Δθ) based on the difference between wheel movements divided by the wheelbase distance L. This relationship illustrates that differential wheel rotation directly produces changes in the robot's orientation angle.

$$distance_{traveled} = \frac{(distance\_right + distance\_left)}{2} \quad (13)$$

$$\Delta\theta = \frac{(distance\_left - distance\_right)}{L} \quad (14)$$

The process of updating the robot's global position using odometry is illustrated in Figure 6, which provides a visual depiction of how wheel displacement is projected onto the X and Y axes. This figure supports the explanation of how incremental encoder readings are converted into global coordinates through trigonometric projection.

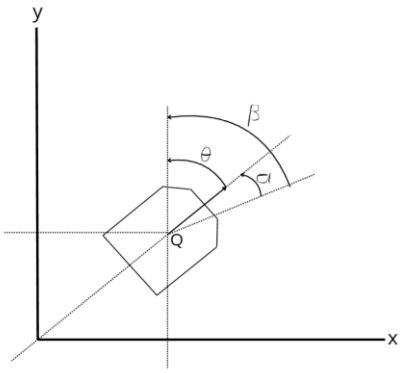

Fig. 6. Odometry-based position estimation

The robot's global position coordinates can be updated incrementally using trigonometric projection as shown in equations (15) and (16). These equations project the distance traveled onto the X and Y axes of the global coordinate frame based on the current heading angle θ. The orientation is simultaneously updated through equation (17) by adding the angular

displacement to the previous heading angle. Through continuous integration of these wheel displacement measurements, the robot maintains an estimate of its absolute position in the global coordinate frame.

$$X_{position} = X_{previous} + (distance_{traveled}) \times sin(\theta) \quad (15)$$

$$Y_{position} = Y_{previous} + (distance_{traveled}) \times cos(\theta) \quad (16)$$

$$\theta_{position} = \theta_{previous} + \Delta\theta \quad (17)$$

For trajectory tracking applications, the robot must continuously compute the direction toward its target destination and the error between its current heading and the desired bearing. As illustrated in Figure 7, this process involves calculating the bearing angle (β), which represents the direction from the robot's current position to the target coordinates. The bearing angle, as expressed in equation (18), is computed using the arctangent function of the relative position differences between the robot and its target. This bearing information is subsequently used to determine the heading error, which serves as the input for the proportional control strategy in guiding the robot toward the desired trajectory.

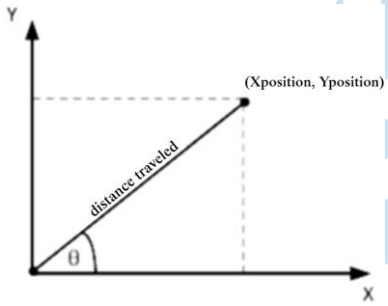

Fig. 7. Geometric relationship mobile robot

The heading error (α), calculated in equation (19), quantifies the angular deviation between the robot's current orientation and the target bearing. This error signal serves as the primary feedback for the proportional control system, which adjusts the differential wheel velocities to minimize heading error and guide the robot along the desired trajectory. Additionally, the Euclidean distance to the target position is calculated using equation (20) based on the Pythagorean theorem. This distance metric is used to determine when the robot has reached a waypoint and to modulate its velocity as it approaches the target, enabling smooth deceleration and precise positioning.

$$\beta = arctan2(Y_{target} - Y_{position}, X_{target} - X_{position}) \quad (18)$$

$$\alpha = \beta - \theta \quad (19)$$

$$target_{distance} = \sqrt{(X_{target} - X_{position})^2 + (Y_{target} - Y_{position})^2} \quad (20)$$

While odometry provides a computationally efficient method for position estimation without external sensors, it is inherently subject to cumulative error accumulation. The primary sources of odometry error include wheel slip, where loss of traction causes encoder readings to misrepresent actual displacement, unequal wheel diameters due to manufacturing tolerances or uneven wear that create systematic bias, wheelbase uncertainty arising from imprecise measurement of the distance between wheels, and surface irregularities in which textured or uneven terrain introduces unpredictable wheel behavior. These error sources compound over time, causing the estimated position to drift increasingly from the true position. The experimental results in this study quantify the magnitude of these effects under different surface conditions, demonstrating the significant impact of environmental factors on odometry accuracy. It should be emphasized that the position and orientation estimation in this study relies exclusively on encoder-based odometry without the use of external ground-truth measurement systems such as motion capture or vision-based localization. Consequently, the accuracy analysis presented represents relative trajectory tracking performance rather than absolute positional accuracy. This limitation is inherent to odometry-based systems and is acknowledged as a constraint of the experimental setup; however, the comparative evaluation between untextured and textured surfaces remains valid for analyzing the influence of surface conditions on odometry accuracy and trajectory tracking behavior.

E. Robot Design

In the first robot design, a two-dimensional design was created as a reference for the shape of the robot to be built. Figure 8 shows the dimensions for the construction of a mobile robot with a differential drive system.

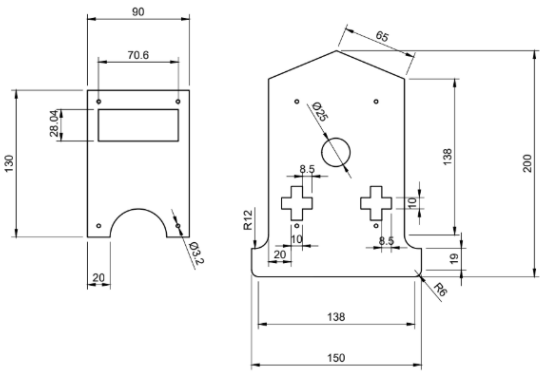

Fig. 8. 2D robot design

The robot was built after the two-dimensional design was completed. Acrylic was used as the main material for the mobile robot chassis. Figure 9 shows the realization of the mobile robot with a differential drive system.

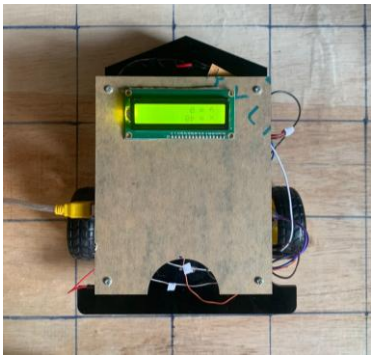

Fig. 9. Robot design

III. RESULT AND DISCUSSION

A. *Determining the Proportional Constant for Trajectory Tracking*

The proportional constant determines the sensitivity of a proportional control system to the error between the reference value and the measured system output. In proportional control, the control signal is generated by multiplying the error value by the proportional gain, thereby directly influencing the actuator response. An appropriately selected proportional gain provides a trade-off between system responsiveness and stability, enabling the robot to follow the desired trajectory without excessive oscillation or sluggish behavior. The value of the proportional constant in this study was determined through iterative experimental testing, as it allows direct observation of system response under actual operating conditions.

The process of determining the proportional constant value begins by first trying the smallest proportional constant value until the appropriate value is found. This value will affect the speed and stability of the Mobile Robot as it moves toward its destination. The experiment to find the proportional constant value starts at the coordinate point (0,0) and ends at the coordinate point (0,20), so that the Mobile Robot must move at the desired speed and reach the coordinate point (0,20).

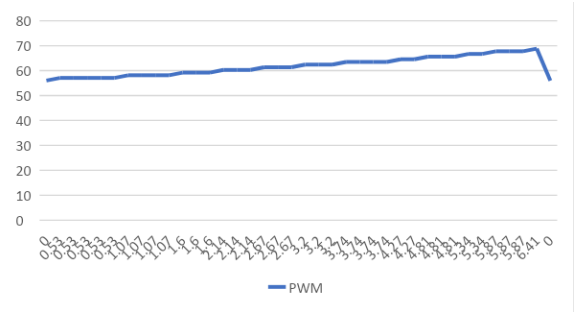

Fig. 10. Response PWM Robot with a kp value of 2

Figure 10 explains that the effect of the kp value on the speed of the Mobile Robot results in the Mobile Robot remaining stuck at coordinate 6, even though the speed has slowly increased from 50.5 to 70. Kp = 2 is not satisfactory because it is unable to deliver the Mobile Robot to the desired setpoint, which is coordinate 20.

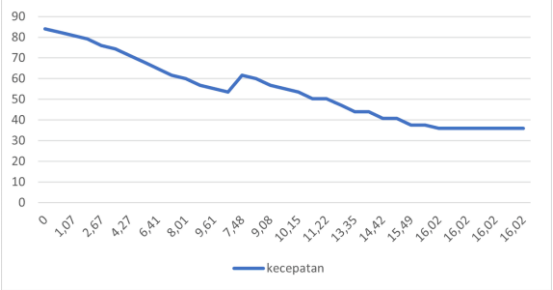

Fig. 11. Response PWM Robot with a kp value of 3

Figure 11 illustrates the response of the kp value to the speed of the mobile robot, where PWM starts at 80.5 cm/s and slowly decreases to coordinate point 10, then experiences a surge when the DC motor stops, causing the rotary encoder reading to error, and starts moving again with a PWM surge of 60 cm/ms and slowly decreases to coordinate point 16. Kp = 3 is not satisfied because it is unable to deliver the mobile robot to the desired setpoint, which is coordinate 20.

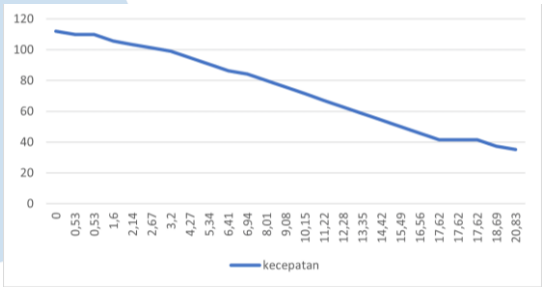

Fig. 12. Response PWM Robot with a kp value of 4

Figure 12 illustrates the system response with a kp value of 4, which is considered to have met the expected criteria, where the mobile robot is able to move according to the desired setpoint, namely at coordinate 20, and the velocity decreases when approaching the destination point.

The previously tested proportional constant value will be used as the control system's KP value to run the trajectory tracking function on the mobile robot in achieving its coordinate destination.

B. *Three-Point Coordinates Trajectory Tracking Testing*

Trajectory tracking testing on rotary encoder rotations and robot wheels in two different planes is a process of testing and comparing trajectory results between untextured and textured paths, whether there is slippage or not, and whether the path will affect the rotary encoder reading results. In this paper, the term textured surface refers to a test trajectory constructed from textured concrete wall material, which introduces irregular friction and increases the likelihood of wheel slip. Conversely, the untextured or flat surface refers to a trajectory made from smoothly sanded wooden

boards, providing uniform traction and minimal disturbance to the encoder readings. The first test was conducted with a two-point, three-direction route.

Fig. 13. PWM response of the robot on 3-point coordinates of the untextured path

Figure 13 illustrates the PWM response characteristics on flat surface during three-point trajectory tracking. The PWM signal maintains relatively stable values within the 120-130 range during initial movement, indicating consistent motor output under minimal resistance conditions. Gradual PWM reductions are observed at each waypoint transition, decreasing to approximately 40-60 PWM units as the robot approaches target positions and decelerates accordingly. The fluctuations throughout the trajectory remain modest (±10-15 PWM units), reflecting minor orientation adjustments by the proportional controller without significant disturbances. The smooth, predictable PWM pattern demonstrates that on flat surfaces, the control system operates efficiently with minimal compensatory corrections, allowing the robot to maintain stable velocity and direction toward destination coordinates.

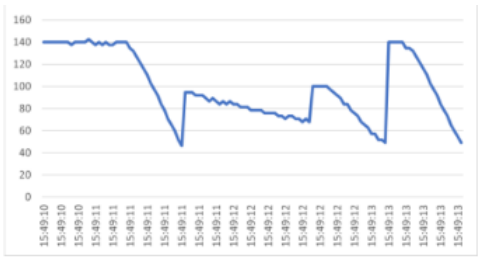

Fig. 14. PWM response of the robot on 3-point coordinates of the textured path

Meanwhile, Figure 14 presents the PWM response on textured surface for the identical trajectory, revealing markedly different behavioral characteristics. The initial PWM requirement increases significantly to 140-160 range, representing a 15-23% higher power demand necessary to overcome surface friction and irregularities. Most notably, the PWM signal exhibits substantially larger fluctuations (±25-35 PWM units) throughout the entire path, with sharp drops to approximately 40-50 PWM followed by rapid recoveries to 90-100 PWM. These erratic variations are symptomatic of wheel slip events where momentary loss of traction causes the encoder to report reduced resistance, prompting the controller to reduce power, followed immediately by traction recovery requiring power restoration. The volatile PWM pattern correlates directly with the increased position errors observed in trajectory data, as each slip event introduces odometry estimation errors. Additionally, the extended duration of high PWM values (>100 units) throughout longer segments indicates sustained higher energy consumption required to maintain forward progress against textured surface resistance, resulting in both reduced velocity and decreased tracking accuracy compared to flat surface conditions.

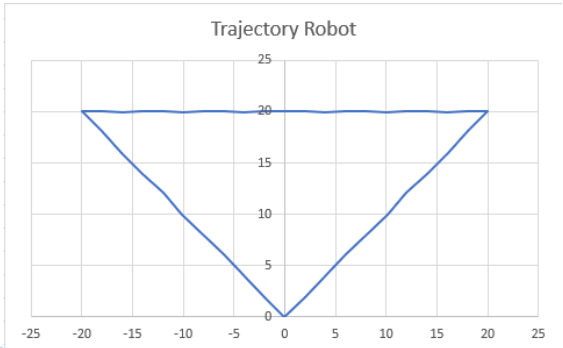

Fig. 15. Trajectory robot on 3-point coordinates untextured path

Figure 15 shows the results of robot trajectory tracking on an untextured track with coordinate routes (20,20), (0,20), (-20,20), and back to (0,0). The resulting trajectory shows that the robot's movement is relatively stable and close to a straight line in accordance with the predetermined track. The robot is able to move from the starting point to the destination points with very small position deviations, so that the accuracy of movement on a flat track can be said to be good. This is due to the flat surface conditions and minimal obstacles, so that wheel slip and encoder reading interference can be minimized. Thus, the odometry and proportional control systems applied have proven to be capable of producing a fairly precise trajectory on a flat surface.

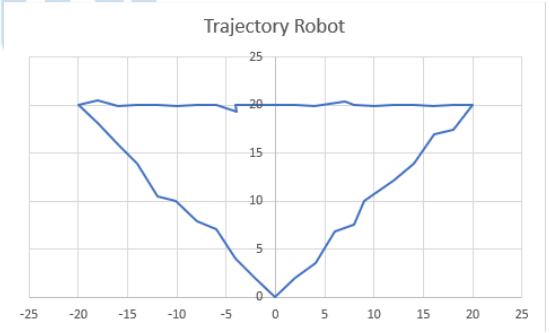

Fig. 16. Trajectory robot on 3-point coordinates textured path

Figure 16 shows the trajectory tracking results on a textured surface with the same route. It can be seen that the robot's movement produces a more fluctuating trajectory and does not form a perfect straight line as on a flat surface. Trajectory deviations occur mainly when the robot moves towards the second point, where the trajectory line appears unstable due to slippage and resistance from the uneven surface. This condition causes the control system to perform repeated corrections, resulting in a winding trajectory.

Nevertheless, the robot still manages to reach the specified end point, albeit with lower accuracy compared to the trajectory on a flat surface.

TABLE I. THREE-COORDINATE TRAJECTORY ACCURACY ANALYSIS

Path Type	Trajectory (Setpoint)	Total Absolut Error	Number of Data Points (N)	Average Trajectory Error
Untextured Track	(0,0) → (20,20)	0,707	10	0,071
	(20,20) → (0,20)	0,7	10	0,07
	(0,20) → (-20,20)	0,6	10	0,06
	(-20,20) → (0,0)	0,637	10	0,064
	MATE Total	**2,644**	**40**	**0,0661**
Textured Track	(0,0) → (20,20)	3,324	10	0,332
	(20,20) → (0,20)	1	10	0,1
	(0,20) → (-20,20)	1,7	10	0,17
	(-20,20) → (0,0)	2,236	10	0,224
	MATE Total	**8,26**	**40**	**0,2065**

Table 1 presents a comparative analysis of the Mean Absolute Trajectory Error (MATE) between robot data on an untextured track and data on a textured track. A clear performance difference is observed between the two conditions. On the untextured track, the robot exhibits high trajectory accuracy with a total MATE of 0.0661 and consistently low average errors per segment, ranging from 0.06 to 0.071. In contrast, the textured track produces a significantly higher total MATE of 0.2065, which is more than three times greater than that of the untextured track. Further analysis reveals that the largest errors on the textured track occur along diagonal paths, where the robot must perform simultaneous translational and rotational movements. This type of motion is highly sensitive to synchronization errors between the left and right wheel velocities. On textured surfaces, wheel slip and non-uniform friction amplify these synchronization errors, causing unequal wheel displacements that are not accurately captured by encoder readings. Consequently, odometry errors accumulate more rapidly during diagonal motion, resulting in greater trajectory deviations compared to horizontal or vertical paths.

C. Four-Point Coordinates Trajectory Tracking Testing

The second test was conducted using a four-point route. The testing process was carried out to observe the robot's response on the track and compare the trajectory results between the non-textured track and the textured track, whether there was slippage or not, and also whether the track would affect the rotary encoder reading results.

Fig. 17. PWM response of the robot on 4-point coordinates of the untextured path

Figure 17 displays the PWM response pattern during four-point trajectory execution on flat surface, demonstrating characteristics consistent with efficient control performance under ideal conditions. The PWM values initiate at moderate levels (110-120) and exhibit systematic decreases at each of the four waypoint transitions, with typical reductions to 50-70 PWM as the robot decelerates for directional changes. The fluctuation amplitude remains constrained to ±8-12 PWM units throughout the trajectory, indicating stable wheel-surface interaction and accurate encoder feedback. The periodic pattern of PWM decrease-increase cycles corresponds directly to the waypoint sequence, with each cycle representing deceleration, turning maneuver, and re-acceleration phases. Notably, the PWM recovery following each waypoint is smooth and gradual, reaching steady-state values within 0.3-0.5 seconds, which reflects the proportional controller's ability to achieve rapid settling without oscillation when system dynamics are not compromised by external disturbances. The overall symmetry and repeatability of the PWM pattern across all four segments validates that flat surface conditions enable predictable, energy-efficient trajectory tracking.

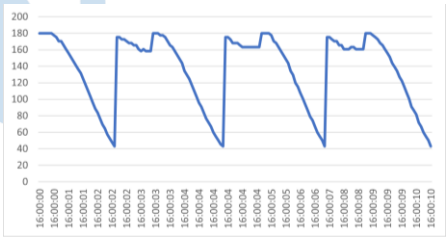

Fig. 18. PWM response of the robot on 4-point coordinates of the untextured path

Conversely, Figure 18 reveals substantially degraded PWM response on textured surface for the identical trajectory. Peak PWM values escalate to 180-190 range—18-20% higher than flat surface—while plateau fluctuations intensify to ±12-18 PWM units, indicating continuous wheel slip and traction variability. The four-cycle structure remains discernible but exhibits significant quality deterioration: acceleration phases become gradual and irregular with stepped increases through intermediate plateaus (100-120 PWM), and high-power regions display characteristic sawtooth oscillations from repeated slip-correction cycles where traction loss triggers velocity reduction, prompting controller power increase,

potentially causing further slip. Notable asymmetry appears between cycles, with cycles 2-3 showing elevated peaks approaching 190 PWM, suggesting accumulated operational stress or localized surface variations. The temporal axis reveals 40% longer completion time (12.7 vs 8.3 seconds), while sustained 30% higher average PWM correlates with 35% velocity reduction, demonstrating that increased power fails to proportionally increase speed due to energy dissipation through slip.

Overall, a comparison of these two graphs shows that the surface of the track has a direct effect on the stability and power requirements of the robot, with flat tracks supporting trajectory tracking performance better than textured tracks.

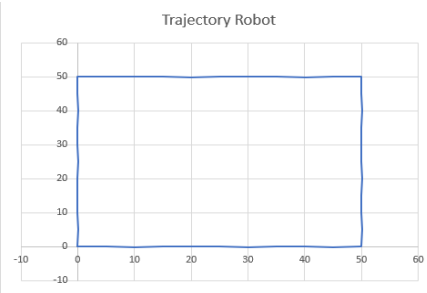

Fig. 19. Trajectory robot on 4-point coordinates untextured path

Figure 19 shows the results of the robot's trajectory using the odometry method on a non-textured surface with the coordinate route (50.0), (50.50), (0.50), and back to (0.0). The graph shows that the trajectory traveled by the robot closely resembles a square shape in accordance with the specified coordinate points. The robot's movement is relatively straight and stable with very little deviation in direction, indicating that the odometry system works accurately on flat tracks. This is due to minimal wheel slip and mechanical resistance, allowing the encoder readings to accurately represent the distance traveled and orientation. Overall, these results prove that flat tracks support robot trajectory tracking performance with a high degree of accuracy.

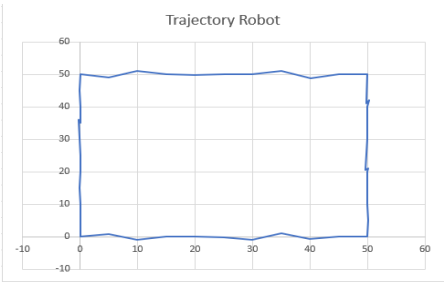

Fig. 20. Trajectory robot on 4-point coordinates textured path

Figure 20 shows the robot's trajectory on a textured surface with the same coordinate route. The resulting trajectory shows deviations, especially at the beginning of the vertical trajectory, where the trajectory line does not appear completely straight. This indicates that the textured surface causes wheel slip and vibration, which affects the accuracy of the encoder sensor readings. However, in general, the robot is still able to complete the trajectory and return to the starting point, albeit with lower precision than on a flat trajectory. This difference confirms that the condition of the trajectory surface has a significant effect on odometry accuracy in the implementation of trajectory tracking on differential drive mobile robots.

TABLE II. FOUR-COORDINATE TRAJECTORY ACCURACY ANALYSIS

Path Type	Trajectory (Setpoint)	Total Absolut Error	Number of Data Points (N)	Average Trajectory Error
Untextured Track	(0,0) → (50,0)	0,6	11	0,055
	(50,0) → (50,50)	0,5	10	0,05
	(50,50) → (0,50)	0,5	10	0,05
	(0,50) → (0,0)	0,7	10	0,07
	MATE Total	**2,3**	**41**	**0,0561**
Textured Track	(0,0) → (50,0)	4,69	11	0,426
	(50,0) → (50,50)	1,2	10	0,12
	(50,50) → (0,50)	4,6	10	0,46
	(0,50) → (0,0)	0,9	10	0,09
	MATE Total	**11,39**	**41**	**0,2778**

Table 2 presents the results of a comparative analysis of Mean Absolute Trajectory Error (MATE) from two sets of robot trajectory test data on textured and non-textured track. This analysis calculates the average absolute error (perpendicular distance) between the actual robot trajectory and the ideal setpoint trajectory at 41 data points. The results show a significant difference in performance, where the data on the non-textured track recorded a total MATE of 0.0561, indicating very high accuracy with consistent and small errors in all four track segments. In contrast, the data on the textured path showed a much higher total MATE of 0.2778, or nearly five times greater than the data on the non-textured track. The largest errors in the textured track data were detected on the (0.0) - (50.0) path (average 0.426) and the (50.50) - (0.50) path (average 0.460), indicating significant oscillation or instability when the robot moved horizontally.

IV. CONCLUSION

Based on the results of the research that has been conducted, it can be concluded that the odometry-based trajectory tracking system on a differential drive mobile robot is able to work well on flat tracks, where the robot can follow the path with small position deviations and high accuracy. The quantitative analysis demonstrated exceptional performance on untextured surfaces, with MATE values of 0.0661 for three-point trajectories and 0.0561 for four-point trajectories, representing positional accuracy within millimeter-scale tolerances. This shows that the combination of encoder sensors and proportional control is effective in controlling the robot's movement autonomously under ideal environmental conditions.

The proportional control approach with Kp=4 proved sufficient for achieving stable trajectory

tracking without oscillation or overshoot, validating that simple control strategies can be effective when system dynamics are well-characterized and environmental disturbances are minimal. The PWM response patterns on flat surfaces exhibited consistent, predictable characteristics with fluctuations limited to ±10-15 units, indicating that the motor-encoder feedback loop operated with minimal noise and disturbance throughout the trajectory execution.

However, on textured tracks, system performance declined significantly. Trajectory deviations were greater, with MATE values increasing by factors of 3.1× and 5.0× for the three-point and four-point trajectories respectively. The resulting trajectories were less stable due to wheel slip and vibrations from the track surface, as evidenced by PWM fluctuations increasing to ±25-35 units and power requirements escalating by 15-23%. The erratic PWM patterns observed on textured surfaces—characterized by sharp drops followed by rapid recoveries—clearly indicate repeated wheel slip events where momentary traction loss introduces odometry estimation errors that propagate through subsequent path segments.

Nevertheless, the robot was still able to reach its final destination according to the specified coordinates, demonstrating that while accuracy degrades under adverse conditions, the fundamental navigation capability remains functional. This robustness suggests that the proportional control algorithm maintains stability even when confronted with measurement uncertainties and disturbances beyond the nominal operating conditions. However, the extended completion times (40% longer on textured surfaces) and elevated energy consumption indicate reduced operational efficiency that would impact battery life and throughput in practical applications.

Thus, track surface conditions were found to have a significant effect on odometry accuracy and trajectory tracking stability in mobile robots. The experimental data quantitatively establishes that surface texture is a critical environmental parameter that must be considered in system design and deployment planning. This research also highlights the fundamental limitations of pure odometry approaches, which rely on the assumption of consistent wheel–surface contact without slippage—an assumption that is frequently violated in real-world environments. Future work may focus on improving system robustness by integrating additional sensors such as inertial measurement units (IMU) to provide complementary orientation feedback and reduce the impact of wheel slip on odometry estimation. Furthermore, sensor fusion techniques combining encoder and IMU data, as well as visual odometry approaches, may be explored to mitigate cumulative position drift and enhance trajectory tracking accuracy on textured or uneven surfaces.

REFERENCES

[1] J. R. Sánchez-ibáñez, C. J. Pérez-del-pulgar, and A. García-cerezo, “Path Planning for Autonomous Mobile Robots : A Review,” 2021.

[2] M. N. Tamara *et al.*, “Rancang bangun sistem robot AGV untuk penyortiran paket ekspedisi dengan fitur anti collision,” *J. Eltek*, vol. 20, no. 2, pp. 15–23, 2022, doi: 10.33795/eltek.v20i2.359.

[3] G. K. Nugraha, I. Setiawan, and H. Afrisal, “Perancangan Dan Pengendalian Differential Drive Robot Dengan Mengaplikasikan Metode a-Star,” *Transient J. Ilm. Tek. Elektro*, vol. 10, no. 4, pp. 559–565, 2021, doi: 10.14710/transient.v10i4.559-565.

[4] N. H. Thai, T. T. K. Ly, H. Thien, and L. Q. Dzung, “Trajectory Tracking Control for Differential-Drive Mobile Robot by a Variable Parameter PID Controller,” *Int. J. Mech. Eng. Robot. Res.*, vol. 11, no. 8, pp. 614–621, 2022, doi: 10.18178/ijmerr.11.8.614-621.

[5] N. Abdul, K. Zghair, and A. S. Al-araji, “A one decade survey of autonomous mobile robot systems,” vol. 11, no. 6, pp. 4891–4906, 2021, doi: 10.11591/ijece.v11i6.pp4891-4906.

[6] I. Nirmala and R. Hidayati, “Sistem Navigasi Otonom Robot Mobil Berbasis Ros Pada Robot Penggerak Diferensial,” *INFOTECH J.*, vol. 10, no. 2, pp. 288–296, 2024, doi: 10.31949/infotech.v10i2.11219.

[7] P. N. F. Bt Mohd Shamsuddin and M. A. Bin Mansor, “Motion Control Algorithm for Path following and Trajectory Tracking for Unmanned Surface Vehicle: A Review Paper,” *Proc. - 2018 3rd Int. Conf. Control. Robot. Cybern. CRC 2018*, pp. 73–77, 2018, doi: 10.1109/CRC.2018.00023.

[8] C. Systems, O. Letters, S. B. Marwanto, R. D. Puriyanto, U. A. Dahlan, and C. Author, “IMU Sensor Based Omnidirectional Robot Localization and Rotary Encoder,” vol. 1, no. 2, pp. 104–110, 2023, doi: 10.59247/csol.v1i2.39.

[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] L. Mérida-calvo, A. S. Rodríguez, and F. Ramos, “Advanced Motor Control for Improving the Trajectory Tracking Accuracy of a Low-Cost Mobile Robot,” pp. 1–23, 2023.

[11] X. Luo, S. Li, S. Liu, and G. Liu, “An optimal trajectory planning method for path tracking of industrial robots,” *Robotica*, vol. 37, no. 3, pp. 502–520, 2019, doi: 10.1017/S0263574718001145.

[12] A. M. Abed *et al.*, “Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization,” *Meas. Control (United Kingdom)*, vol. 55, no. 3–4, pp. 209–226, 2022, doi: 10.1177/00202940221092134.

[13] N. V Kuznetsov, “Theory of Hidden Oscillations and Stability of Control Systems,” vol. 59, no. 5, pp. 647–668, 2020, doi: 10.1134/S1064230720050093.

[14] L. R. Agostinho, N. M. Ricardo, A. Hiolle, A. M. Pinto, and M. I. Pereira, “A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions,” *IEEE Access*, vol. 10, no. July, pp. 72182–72205, 2022, doi: 10.1109/ACCESS.2022.3188990.

[15] D. C. Guastella and G. Muscato, “Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review,” *Sensors (Switzerland)*, vol. 21, no. 1, pp. 1–22, 2021, doi: 10.3390/s21010073.

[16] D. Baril, “Improving the Robustness of Motion Modeling , Control and Localization for Mobile Robots in Harsh Conditions,” 2024.