PID-Based Motion Control of a Differential Drive Robot: A Simulation Study in CoppeliaSim

Fachruddin Ari Setiawan[1], Ghali Marzani[2], Daffa Arkana Sheva Budianto[3], Muhamad Julian Hadinata[4]
[1,3,4]Study Program of Computer Control Engineering, Engineering Department, Madiun State of Polytechnic, Madiun, 63133, Indonesia
[2]Study Program of Software Engineering, Engineering Department, Madiun State of Polytechnic, Madiun, 63133, Indonesia
[1]fachruddin_ari@pnm.ac.id

Abstract
Motion control is a critical aspect of mobile robot operation, particularly for differential drive robots that require precise regulation of linear and angular velocities. This paper presents the implementation of a PID-based motion control system for a differential drive mobile robot in a CoppeliaSim simulation environment. The proposed control architecture employs two independent PID controllers operating in a closed-loop configuration to regulate the robot's linear (surge) and angular (yaw) velocities. The robot kinematic model is used to transform the controller outputs into wheel velocity commands. Several simulation scenarios with varying velocity references were conducted to evaluate the performance of the proposed approach. Controller performance was assessed using standard metrics, including transient response characteristics and Root Mean Square Error (RMSE). The simulation results demonstrate that the PID controller achieves stable tracking of the desired velocity references with low tracking errors. The obtained RMSE values of 0.01339 for linear velocity and 0.01496 for angular velocity indicate reliable motion control performance in the simulated environment. The controller performance is further characterized by its steady-state accuracy and transient response behavior during setpoint changes. These findings confirm that PID-based control remains an effective and practical solution for low-level motion control of differential drive robots and provides a solid baseline for further research and experimental validation.

Keyword: *Differential drive robot, PID control, motion control, CoppeliaSim simulation.*

I. INTRODUCTION

Differential Drive Robot (DDR) is one of the configurations of mobile robots that is often applied in the industrial field, autonomous services, and robotics research because the mechanical structure and mathematical model are quite simple but capable of controlling linear and angular speeds. Controlling linear speed (surge) and angular speed (yaw) is an important task to be done so that the robot can move according to the given command. This control method usually uses a feedback system that provides the current state to obtain the difference (error) between the current condition and the desired condition, and aims to minimize the error in real-time.

One of the frequently used control methods is the Proportional-Integral-Derivative (PID) control method because it has a simple structure, wide application, and the ability to provide a stable response in many types of mechanical and electronic systems. PID controllers are generally often applied to mobile robots to control wheel speed, position, and orientation either directly or indirectly by becoming a more complex navigation system. For example, PID controllers have been used to control the differential drive of mobile robots and have shown that PID successfully stabilizes motor speed and improves the quality of the resulting map when integrated with SLAM (Simultaneous Localization and Mapping), as well as ensuring the system response is in accordance with the given setpoint (Fahmizal *et al.*, 2024).

A crucial initial step in robot control research is simulation, as it provides a controlled environment for testing an algorithm before applying it to a real robot. One frequently used simulation platform is CoppeliaSim (formerly V-REP), an application that provides numerous robot models (even allowing you to create custom robot models) and realistic simulation scenarios that are ideal for comprehensively exploring robot dynamics and the performance of a control method.

Although the PID method has many advantages, this method also has limitations, including: (1) sensitivity to parameter tuning, PID is highly dependent on tuning its control parameters in order to provide stable results and optimal response; (2) simple linear model, the PID structure is a linear control that will be less suitable for non-linear systems; (3) cannot adapt to dynamic environments, if the system is given significant external disturbances, then the classical PID cannot adjust its own parameters (Ghazal *et al*., 2025). Therefore, this study will use an environment that does not have external disturbances so that the PID can be optimal in controlling mobile robots.

Based on the background and limitations that have been explained, this study aims to: (1) apply PID to control linear velocity (surge) and angular velocity (yaw) on a differential drive robot in the CoppeliaSim simulation environment; (2) analyze the performance of PID control in maintaining the desired speed reference using gain parameters determined through trial and error methods; (3) evaluate the system response through performance metrics such as steady-state error, transient responses, and stability to setpoint changes in the simulation; and (4) provide an experimental basis for simulation that can be used as a foundation for further research, such as parameter optimization, adaptive control, or implementation in physical robots.

II. THEORY

A. Differential Drive Mobile Robot Kinematics

Differential drive wheeled mobile robots are one of the most common robot configurations used in mobile robotics research and applications due to their simple mechanical structure and ability to generate translational and rotational motion independently. This robot uses two main drive wheels mounted parallel to one axis, with the speed of each wheel controlled separately to generate linear velocity (surge) and angular velocity (yaw). The differential drive kinematic model is non-holonomic, so the robot's movement is limited by the orientation and configuration of the wheels.

The relationship between wheel angular velocities and the robot linear and angular velocities is defined by the kinematic model given in equation (1) and (2).

$$v = \frac{r}{2}(\omega_L + \omega_R) \tag{1}$$

$$\omega = \frac{r}{L}(\omega_L - \omega_R) \tag{2}$$

where r represents the wheel radius, L is the distance between the wheels, and ω_L and ω_R are the angular velocities of the left and right wheels, respectively. This model is widely used as a basis for designing motion control systems in differential drive robots, both for speed control, trajectory tracking, and autonomous navigation.

Various studies have adopted this kinematic model in designing mobile robot control systems. A study by Setiawan (2022) applied a differential drive model to mobile robots in the context of target tracking and formation control in a pursuit–evasion game scenario, utilizing the Modified Extremum Seeking Control (MESC) approach. The results of this study indicate that the differential drive kinematic model remains relevant and flexible for combination with various control strategies, including optimization-based adaptive control (Setiawan *et al*., 2022).

Although the control approaches used are different, these studies confirm that understanding differential drive kinematics is an important foundation in the development and evaluation of robot motion control algorithms. In this study, the same kinematic model is used as the basis for implementing PID control to regulate the linear and angular velocities of the robot in the CoppeliaSim simulation environment.

B. Proportional-Integral-Derivative (PID) Controller

The PID controller is one of the most widely used closed-loop control techniques in engineering systems due to its simplicity of implementation, ability to minimize steady-state error, and tunable transient response via gain parameters. In general, the PID control law can be written as in equation (3).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{3}$$

where $e(t)$ is the error between the reference and the actual value, K_p,K_i,K_d are the proportional, integral, and derivative gains, respectively. The proportional component provides a proportional response to the current error; the integral addresses the steady-state error; while the derivative helps dampen overshoot and adds stability to sudden error changes.

PID is widely used in robotics to control motor speed, position, and orientation of robots because of its ability to stabilize relatively simple systems without requiring a complete dynamic model. PID implementations can be at the motor level or directly at the robot motion level, such as controlling the linear and angular velocities of the robot (Haq, 2017).

C. PID Control in Differential Drive Robots

PID applications in differential drive robots are usually performed on wheel speed control loops or translational and yaw speed control loops separately. In some studies, PID is used to control the speed of DC motors that drive the left and right wheels independently so that the robot's linear and angular speeds can be maintained according to the desired reference values. Optimization of PID parameters can be done through experimental methods such as trial and error, Ziegler–Nichols, or other optimization techniques to obtain the best performance in terms of settling time, overshoot, and steady-state error.

Several studies also show the integration of PID with additional systems such as Simultaneous Localization and Mapping (SLAM) to improve the stability of robot motion in real environments, where PID control plays an important role in maintaining consistent robot speed so that the performance of the SLAM system is not affected by significant speed fluctuations. (Fahmizal *et al*., 2024).

D. Recent Developments and Variations in PID Applications

In recent literature, PID approaches continue to develop to overcome the limitations of classical control in mobile robots. One example is Hybrid Control, which combines PID with other methods such as fuzzy logic to improve control response by adaptively adjusting PID parameters based on varying error conditions. Such studies demonstrate improved stability and response speed compared to pure PID (Sutisna *et al*., 2016).

Furthermore, there is research on Neural Network-Assisted PID, integrating PID with artificial neural networks to adjust parameters in real time, particularly in trajectory tracking in differential drive robots, demonstrating an adaptive control approach to address nonlinear dynamics (Ly *et al*., 2024).

Comparative Control Studies also exist, comparing PID with other control methods such as backstepping or sliding mode control in the context of trajectory tracking, demonstrating that PID can provide competitive performance when combined with appropriate kinematic control design (Yigit & Sezgin, 2023).

E. Performance Metrics in Motion Control

Evaluating controller performance in differential drive robots is a crucial aspect for assessing the effectiveness of control algorithms in maintaining motion stability and accuracy. In motion control systems, performance metrics are generally used to measure the system's response to setpoint changes and disturbances, as well as the controller's ability to minimize error. Some performance metrics commonly used in mobile robot research include steady-state error, transient response, robustness, and statistical error measures such as Root Mean Square Error (RMSE) (Siegwart *et al*., 2017; Corke, 2017).

Steady-state error is used to evaluate the difference between a reference value and the actual value when the system has reached a stable state. This value indicates the controller's ability to maintain system output at the setpoint over a long period of time. Meanwhile, transient response characteristics such as rise time, settling time, and overshoot are used to describe the system's speed and stability in response to changes in the linear or angular velocity reference.

In addition to these metrics, Root Mean Square Error (RMSE) is a widely used error measure in motion control and trajectory tracking evaluations due to its ability to represent the overall magnitude of error over a given time period. RMSE is calculated based on the root mean square error between the reference signal and the actual output signal, thus giving a greater penalty to errors with high amplitude. Mathematically, RMSE is formulated as in equation (4).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - x_i^{ref}\right)^2} \quad (4)$$

where x_i represents the actual system value (e.g., the robot's linear or angular velocity), x_i^{ref} is the desired reference value, and N is the number of observation samples.

In the context of motion control in differential drive robots, RMSE is often used to evaluate the tracking accuracy of linear velocity (surge) and angular velocity (yaw). A smaller RMSE value indicates that the control system is able to consistently follow the reference with lower error throughout the simulation. Therefore, RMSE is an effective metric for comparing PID controller performance under various test conditions, such as setpoint changes or control parameter variations (Li *et al*., 2019; Zhang *et al*., 2020).

Using RMSE alongside other performance metrics provides a more comprehensive evaluation of control system performance. In this study, RMSE is used as one of the main indicators to assess the quality of robot velocity control in CoppeliaSim simulations, ensuring quantitative and objective analysis results.

III. METHOD

A. Research Framework

This study uses an experiment-based simulation approach to evaluate the motion control performance of a differential drive robot. The research framework generally consists of several main stages, namely: (1) kinematic modeling of the differential drive robot; (2) design of a PID control system for linear velocity and angular velocity; (3) implementation of the control system in the CoppeliaSim simulation environment; (4) testing the system on several reference velocity scenarios; (5) evaluation of the controller performance using predetermined performance metrics. This approach was chosen because it allows for a controlled analysis of the control system behavior before being applied to a physical robot.

B. Differential Drive Robot Model

The robot used in this study is a wheeled mobile robot with a differential drive configuration, consisting of two main drive wheels and one balancing wheel. The robot's kinematic model is used to relate the angular velocities of the left and right wheels to the robot's linear and angular velocities. The robot's linear velocity v and angular velocity ω are expressed in equations (1) & (2). This model is used as a basis for designing the robot's speed control system.

C. PID Controller Design

Robot motion control is performed by adjusting the linear velocity (surge) and angular velocity (yaw) using two separately designed PID controllers. Each PID controller receives an error signal derived from the difference between the reference value and the actual robot velocity. The general PID control law is formulated in equation (3).

In this study, two PID controllers were used: the first PID controller was used to control the robot's linear velocity and the second PID controller was used to control the robot's angular velocity. The outputs of both controllers were then converted into left and right wheel speed signals based on the robot's kinematic model.

Parameters were determined through a trial-and-error simulation experiment-based tuning process until a stable system response was achieved that met the desired performance criteria.

D. Simulation Environment

The simulation was conducted using CoppeliaSim, which provides a dynamic and realistic simulation environment for testing mobile robot systems. A differential drive robot model was built or selected from an available library, then configured with physical parameters such as wheel radius and wheel spacing.

A PID controller was implemented using a script connected to the robot's wheel actuators. Speed sensors were used to obtain the robot's actual linear and angular velocities, which were then used as feedback in the closed-loop control system.

E. Test Scenarios

To evaluate the performance of the PID controller, several test scenarios were designed, including: (1) linear velocity response testing with gradual setpoint changes; (2) angular velocity response testing with changes in the yaw reference value; (3) linear and angular velocity combination testing to simulate robot maneuvers. Each scenario was run for a certain time interval and the actual and reference velocity data were recorded for analysis purposes.

F. Performance Evaluation

The control system performance evaluation was conducted using several performance metrics, namely: (1) steady-state error; (2) transient response characteristics; and (3) Root Mean Square Error (RMSE).

RMSE is used to measure the accuracy of tracking the robot's linear and angular velocities during the simulation. The RMSE values were calculated using the difference between the reference velocity and the measured actual velocity obtained from the simulation at each sampling time. The RMSE formulation is given in equation (4), where N represents the total number of sampled data points. Simulation data is processed to calculate the RMSE value, which is then used as the main indicator in assessing the quality of PID control in the differential drive robot.

IV. RESULTS AND DISCUSSION

A. Simulation Setup and Data Acquisition

The simulation was conducted in CoppeliaSim using a differential drive mobile robot model. The robot parameters and PID gains were selected based on preliminary tuning experiments and are summarized in Table 1.

Table 1. Simulation Parameters

Parameter	Value
Wheel radius	0.1 m
Distance of both wheel	0.3 m
Time Sampling	0.1 s
Simulation Duration	70 s

The selected simulation parameters were chosen to represent a typical small-scale differential drive mobile robot commonly used in laboratory environments. The PID control structure was selected due to its simplicity, robustness, and widespread use as a baseline low-level controller in mobile robotics.

The robot that has been made and used in Coppeliasim is shown in Figure 1.

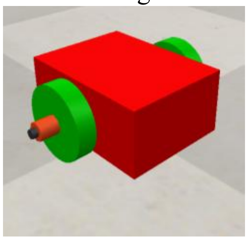

Figure 1. Differential Drive Robot in CoppeliaSim

B. Linear Velocity (Surge) Control Results

This test was conducted by controlling the robot's surge speed using PID with the parameters K_P, K_I, K_D sequentially valued at 8, 0.8, and 0. The PID gains were determined through iterative tuning in simulation to achieve a compromise between fast response and minimal overshoot. The robot's surge speed was given various setpoint values to see the response of the controlled speed.

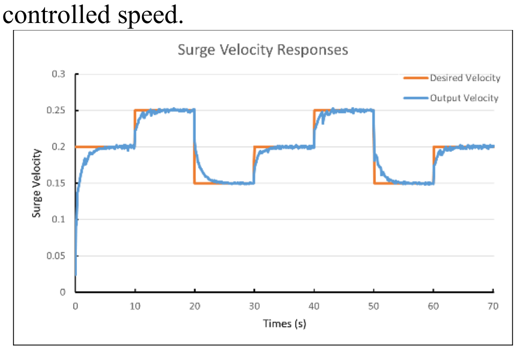

Figure 2. Surge Velocity Responses

Figure 2 illustrates the response of the robot's linear velocity to the given reference signal. The PID controller is able to track the desired velocity with minimal steady-state error. A small transient overshoot can be observed during step changes in the reference, indicating the influence of the proportional gains.

For the surge velocity response, the rise time was approximately 1.3 s, measured from 10% to 90% of the reference velocity. The settling time was approximately 2.5 s, indicating a fast and stable response with minimal oscillation. The transient response parameters were obtained by visual inspection of the simulation plots using standard control system definitions for rise time and settling time.

The tracking accuracy of the linear velocity was evaluated using the Root Mean Square Error (RMSE). The RMSE values were calculated using the difference between the reference surge velocity and the measured actual surge velocity obtained from the simulation at each sampling time. The obtained RMSE value of 0.01339 indicates that the PID controller maintains a consistent tracking performance throughout the simulation period.

C. Angular Velocity (Yaw) Control Results

This test was conducted by controlling the robot's yaw speed using PID with the parameters KP, KI, KD sequentially valued at 0.5, 0.5, and 0.2. The robot's yaw speed was given various setpoint values to see the response of the controlled yaw speed. For its implementation, the robot's surge speed was controlled at a certain setpoint value, namely at 0.2.

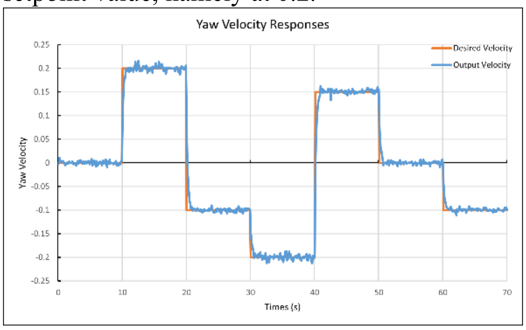

Figure 3. Yaw Velocity Responses

Figure 3 shows the angular velocity response of the robot under PID control. The controller demonstrates stable tracking behavior with acceptable transient characteristics and no sustained oscillations.

The yaw velocity controller exhibited a faster transient response, with a rise time of approximately 0.8 s and a settling time of about 1.8 s. These results indicate good responsiveness and stability of the angular velocity control.

The RMSE of the angular velocity tracking is calculated to quantify the control accuracy. RMSE value obtained from the yaw speed control test is 0.01496. The relatively low RMSE value confirms the effectiveness of the PID controller in regulating the yaw motion of the differential drive robot.

D. Combined Motion Control Performance

When both linear and angular velocity controllers are activated simultaneously, the robot exhibits stable motion behavior. The interaction between the two control loops does not introduce significant performance degradation, indicating proper decoupling at the velocity control level.

This scenario is executed by activating both the surge and yaw controllers, and assigning them setpoints that vary over time. The PID controller parameters are selected using the same values as those in the previous two scenarios.

The test results of the PID control combination for surge and yaw speed are shown in Figure 4-6.

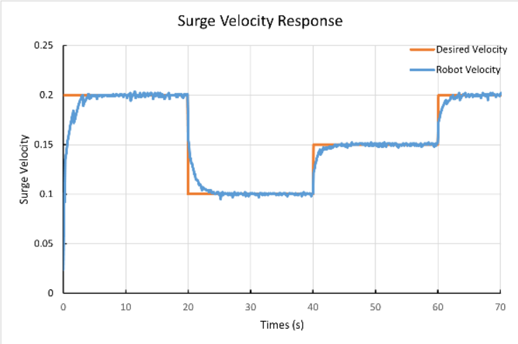

Figure 4. Surge Velocity Responses of Combined Motion Control Scenario

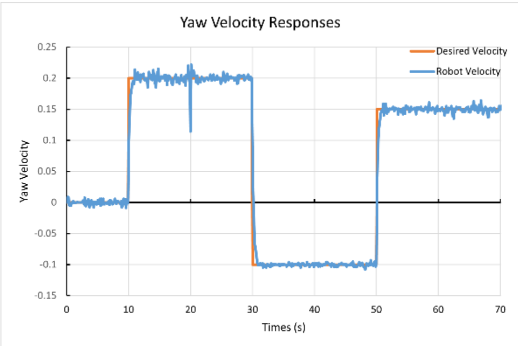

Figure 5. Yaw Velocity Responses of Combined Motion Control Scenario

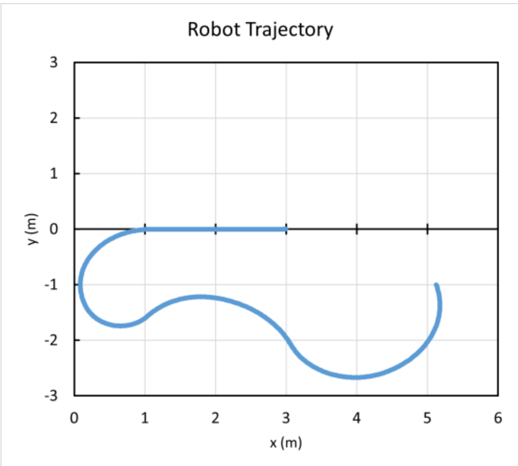

Figure 6. Robot Trajectory of Combined Motion Control Scenario

Figure 4 illustrates the linear velocity response of the robot compared with the reference signal. This figure is used to evaluate the tracking accuracy and transient response of the surge velocity controller. From the figure, it can be seen that the robot's speed can be controlled to follow the setpoint or desired value given.

Meanwhile, Figure 5 shows that the yaw speed can be controlled according to the desired value. From the figure, it can be seen that there has been a spike in the value at the 20th second of the simulation. This can occur because at that second, the surge speed changes from a value of 0.2 to a value of 0.1, thus affecting the yaw speed response. However, with the PID controller used, it can be seen that the spike in value only occurs momentarily and the value returns to follow the given setpoint.

The steady-state error indicates the controller’s ability to maintain the desired velocity once the transient response has settled, while the transient response reflects how quickly and smoothly the system reacts to changes in the reference signal.

Figure 6 shows the trajectory of the controlled robot. It is indicated by the robot's location on the x-axis and y-axis. The robot's initial position is at (3,0) and moves towards the negative x-axis, then turns left, then right, and then left again. This action corresponds to a positive yaw velocity value, then changes to negative, and returns to positive. This indicates that the yaw velocity value is in accordance with the trajectory created by the controlled robot.

Overall, Figures 4–6 demonstrate that the proposed PID controller is capable of maintaining stable velocity tracking for both linear and angular motions. Figure 4 shows accurate surge velocity tracking with minimal steady-state error, while Figures 5 and 6 confirm consistent yaw control and acceptable transient behavior. These results collectively indicate that the control system performs reliably under the tested simulation scenarios.

It should be mentioned that all of the results displayed were from a simulation environment. Wheel slip, sensor noise, actuator saturation, and external disturbances were not specifically modelled. Future experimental validation should take these elements into account as they may have an impact on controller performance in practical applications. Future study should incorporate sensor noise models, test the controller under disturbances, and validate the method on a real robot platform.

V. CONCLUSION

This study has demonstrated the implementation and evaluation of a PID-based motion control system for a differential drive mobile robot in a CoppeliaSim simulation environment. The research was motivated by the need for a simple yet effective low-level control strategy capable of regulating both linear (surge) and angular (yaw) velocities of mobile robots with acceptable accuracy and stability. By employing separate PID controllers for linear and angular velocity, the proposed control scheme was able to achieve stable closed-loop performance under various reference velocity scenarios.

Based on the simulation results, the PID controller successfully tracked the desired linear and angular velocity references with minimal steady-state error and stable transient responses. Quantitative evaluation using the Root Mean Square Error (RMSE) metric showed that the tracking error for linear velocity was 0.01339, while the angular velocity tracking error was 0.01496. In addition to low RMSE values, the controller demonstrated small steady-state errors and acceptable transient responses, confirming stable velocity regulation. These values indicate that the controller maintained consistent performance throughout the simulation period and was able to suppress excessive oscillations during setpoint changes. The combined motion tests further confirmed that the interaction between linear and angular control loops did not introduce significant performance degradation, demonstrating adequate decoupling at the velocity control level.

The findings of this study confirm that, despite its simplicity, the PID controller remains a reliable and effective solution for motion control of differential drive robots in simulation environments. The results are consistent with existing literature, which shows that properly tuned PID controllers can provide satisfactory tracking performance for mobile robot applications, particularly as low-level controllers. However, it should be noted that this study was conducted entirely in a simulation environment without considering external disturbances, sensor noise, or actuator nonlinearities that are commonly present in real-world robotic systems.

In conclusion, the proposed PID-based motion control approach provides a solid baseline for differential drive robot control in CoppeliaSim and can serve as a foundation for further research. Future work may extend this study by implementing adaptive or optimization-based PID tuning methods, incorporating external disturbances, and validating the control strategy on a physical robot platform to assess its robustness and practical applicability.

REFERENCE

Corke, P. (2017). Robotics, Vision and Control: Fundamental Algorithms in MATLAB. 2nd edn. Cham: Springer.

Fahmizal, F., Pratikno, M. S., Isnianto, H. N., Mayub, A., Maghfiroh, H., & Anugrah, P. (2024). Control and Navigation of Differential Drive Mobile Robot with PID and Hector SLAM: Simulation and Implementation. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, 10(3), pp. 594–607. doi: 10.26555/jiteki.v10i3.29428.

Ghazal, Z., Al-Bustami, A., Gaaloul, K. & Kwon, J., (2025). Systematic Evaluation of Initial States and Exploration-Exploitation Strategies in PID Auto-Tuning: A Framework-Driven Approach Applied on Mobile Robots. arXiv preprint arXiv:2505.03159.

Haq, R., (2017). Kendali posisi mobile robot menggunakan sistem proportional integral derivative (PID) dengan metode odometry. Inovasi Fisika Indonesia, 6(3).

Li, X., Wang, J. and Zhao, Y. (2019) 'PID-based motion control of differential drive mobile robots under non-linear constraints', Journal of Intelligent & Robotic Systems, 95(2), pp. 423–437.

Ly, T.T.K., Thai, N.H. & Phong, L.T., (2024). Design of neural network-PID controller for trajectory tracking of differential drive mobile robot. Vietnam Journal of Science and Technology, 62(2), pp.374-386.

Setiawan, F.A., Agustinah, T. and Fuad, M., (2022). Modified Extremum Seeking Control for Target Tracking and Formation Control in Pursuit-Evasion Game. JAREE (Journal on Advanced Research in Electrical Engineering), 6(2).

Siegwart, R., Nourbakhsh, I.R. and Scaramuzza, D. (2017). Introduction to Autonomous Mobile Robots. 2nd edn. Cambridge, MA: MIT Press.

Sutisna, U., Siregar, W.D. and Nurhadiyono, S., (2016). PENERAPAN KENDALI HYBRID LOGIKA FUZZY-PID UNTUK MENINGKATKAN PERFORMA NAVIGASI ROBOT BERODA WALL

FOLLOWER. Techno (Jurnal Fakultas Teknik, Universitas Muhammadiyah Purwokerto), 17(2), pp.79-87.

Yigit, S. & Sezgin, A., (2023). Trajectory tracking via backstepping controller with PID or SMC for mobile robots. Sakarya University Journal of Science, 27(1), pp.120-134.

Zhang, Y., Li, S. and Chen, W. (2020) ‘Analysis and implementation of PID control for mobile robot velocity tracking’, IEEE Access, 8, pp. 119827–119838.