Control and Navigation of Differential Drive Mobile Robot with PID and Hector SLAM: Simulation and Implementation

Fahmizal[1], Matthew Sebastian Pratikno[1], Hidayat Nur Isnianto[1], Afrizal Mayub[2], Hari Maghfiroh[3], Pinto Anugrah[4,5]
[1]Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Indonesia
[2]Graduate School of Science Education, Universitas Bengkulu, Bengkulu, Indonesia
[3]Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia
[4]Department of Electrical Engineering, Universitas Andalas, Padang, Indonesia
[5]School of Electrical Engineering and Computer Science, University of Queensland, Brisbane, Australia

ARTICLE INFO

Article history:

Received July 12, 2024
Revised August 30, 2024
Published September 17, 2024

Keywords:

Mobile robot;
SLAM;
Path planning;
Navigation;
ROS

ABSTRACT

Navigation technology is essential in fields like transportation and logistics, where precise mapping and localization are critical. Simultaneous Localization and Mapping (SLAM) technologies, such as Hector SLAM, enable robots to map environments by detecting and predicting object locations using sensors like LiDAR. Unlike other SLAM methods, Hector SLAM operates without odometry, relying solely on LiDAR data to produce accurate maps. This study investigates the application of Hector SLAM in a differential drive mobile robot controlled via the Robot Operating System (ROS), with PID control managing the motor speeds. The research contribution is the integration of Hector SLAM with PID control to enhance mapping accuracy in environments without odometry data. The method involves testing the robot's mapping performance in an indoor environment, focusing on the impact of varying linear and angular velocities on the quality of the generated maps. The PID control was tuned to ensure stable speed values for the robot's differential drive motors. Results show that Hector SLAM, when combined with well-tuned PID control, generates highly accurate maps that closely match the actual environment dimensions, with minimal errors. Specifically, the mapping error was found to be within 0.10 meters, validating the effectiveness of this approach in non-odometric systems. In conclusion, the study demonstrates that Hector SLAM, supported by PID-controlled motor stability, is an effective solution for mapping in differential drive mobile robots, particularly in scenarios where odometry is unavailable.

Corresponding Author:

Hari Maghfiroh, Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia
Email: hari.maghfiroh@staff.uns.ac.id

1. INTRODUCTION

Navigation technology is a crucial aspect of modern technological development, playing a vital role in various fields such as transportation, healthcare, and logistics. The rapid advancement of information and communication technology has led to a growing demand for devices that are not only effective and efficient but also possess a high level of intelligence, significantly aiding human activities. Robotics-based logistics devices, particularly within warehousing contexts, are among the technologies that can be seamlessly integrated into our surroundings [1], [2].

For mobile robots, navigation is an essential function [3], [4]. It enables the robot to explore and map unknown environments, creating a static map that serves as reference data for future navigation. This map allows the robot to travel from one point to another while avoiding obstacles [5]. There are general problems of navigation which are perception, localization, motion control, and path planning [6], [7], [8]. Perception in robotics refers to the ability of a robot to gather and interpret sensory information from its environment [9]. Localization is the process by which a robot determines its position and orientation within a given environment [10], [11]. Motion control refers to the algorithms and techniques used to direct the robot's movements, ensuring that it can follow a desired trajectory or path. Path planning is the process of determining the best route or sequence of movements that a robot should take to reach its destination while avoiding obstacles [12], [13], [14].

LiDAR (Light Detection and Ranging) is a key sensor used in navigation, employing laser beams to detect objects within a specified range [15]. By measuring the time, it takes for the laser signal to be reflected from objects, LiDAR facilitates the construction of a map and navigation system through the Simultaneous Localization and Mapping (SLAM) algorithm. SLAM allows the robot to concurrently map its environment and determine its location within it.

SLAM has evolved into various methodologies, such as Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping (ORB-SLAM), Large-Scale Direct Monocular Simultaneous Localization and Mapping (LSD-SLAM), and Hector Simultaneous Localization and Mapping (Hector SLAM) [16]. ORB-SLAM is known for its ability to create 3D maps using monocular, stereo, and RGB-D cameras with a feature-based approach. LSD-SLAM, on the other hand, utilizes a direct method to achieve similar outcomes. Hector SLAM, however, specializes in 2D mapping and does not require odometry; instead, it relies on laser scan data [17]. Despite this, it still provides accurate and robust localization [18]. Hector SLAM's ability to function effectively without odometry, using only laser scan data for mapping, makes it particularly suitable for environments where wheel odometry is unreliable or unavailable.

Mapping is conducted by the mobile robot as it navigates through the room and its surroundings. The environmental conditions and domain location must be carefully considered, as they significantly influence the robot's performance and the resulting map [19]. Hector SLAM is capable of producing maps that closely resemble the actual indoor environment [20], with the resulting maps demonstrating high precision and stability [21], [22].

The performance of Hector SLAM has been validated by various researchers [23], [24], [25]. In [23], Hector SLAM was combined with LiDAR and ROS in a mobile robot to map an L-shaped environment, proving its effectiveness in unknown settings. Yu and Zhang [24] enhanced Hector SLAM's accuracy by integrating odometry, IMU, and a Kalman filter, resulting in an improved method named information fusion. This method demonstrated enhanced mapping accuracy. Additionally, in [25], Hector SLAM was compared to Gmapping, with results showing that Hector SLAM produced more accurate maps.

This study aims to design and implement control and navigation for a differential drive mobile robot using PID control and Hector SLAM within a specified environment. The contributions of this research include:

1) Developing a PID control system for DC motor speed control in a differential drive mobile robot, including a GUI interface for PID tuning.
2) Designing and implementing robot navigation that leverages Hector SLAM using ROS and Rviz for visualization.

This paper consists of four parts. Section 2 reviews the mobile robot used and the proposed control and navigation algorithm. The simulation and experimental results are explained in Section 3. Finally, the conclusion is given in Section 4.

2. MATERIAL AND METHOD

2.1. Robot Hardware

This study utilized a non-holonomic differential drive mobile robot, a system that employs two independent motors to control the robot's movement vector. Robot motion production is influenced by the values of angular speed (ω) and linear speed (v) [26]. When every wheel travels at the same speed, the robot will move straight. When the wheels move at various rates, the robot will maneuver. The lower wheel rotation will be followed by robotic maneuvers. The differential drive system was chosen for its simplicity and effectiveness, particularly when coupled with DC motors due to their straightforward control mechanisms [27]. A DC motor is a simple type of motor to use [28]. But in its use, the load frequently causes the DC motor's speed to drop; as a result, the speed is not constant, necessitating the creation of a controller [29], [30], [31], [32].

The robot was equipped with two active wheels, two passive castor wheels, a DC motor driver, a Teensy 3.5 microcontroller, a Jetson Nano for processing, and a LiDAR sensor Fig. 1. The Jetson Nano served as the system's primary processor, handling SLAM and navigation tasks. A remote PC (NUC) was used to visualize mapping and navigation processes via a local network, using Virtual Network Computing (VNC) for communication. The Teensy 3.5 microcontroller managed motor speed control via a PID controller, with the LiDAR sensor providing the necessary laser data for Hector SLAM. The entire system primarily operates under ROS, excluding the PID control, with each ROS process running as a node that communicates to perform tasks [33]. The robot design and its implementation are illustrated in Fig. 2 and Fig. 3, respectively.

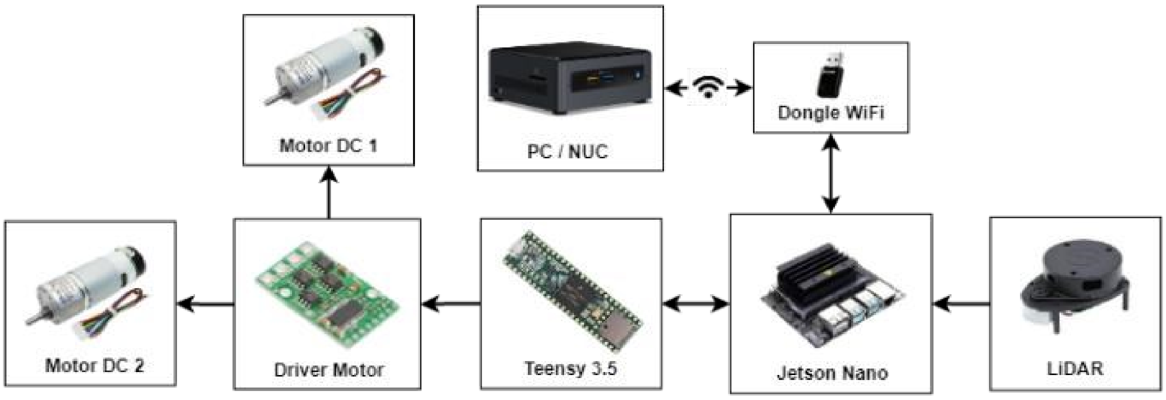


Fig. 1. System diagram

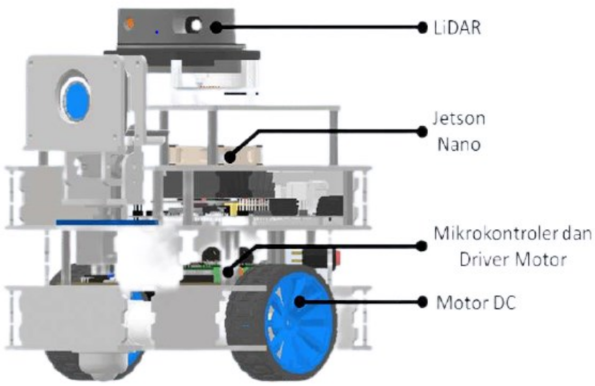


Fig. 2. Model of differential mobile robot

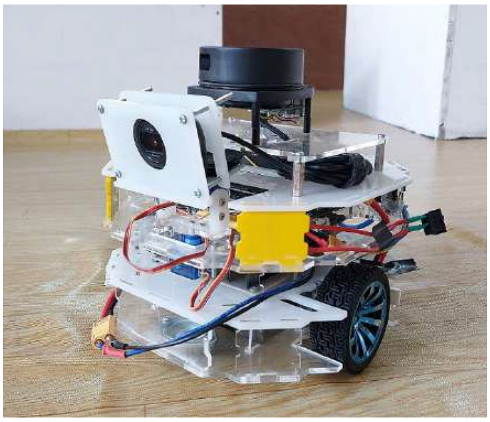


Fig. 3. Real-developed differential mobile robot

2.2. Methods

The study employed two primary methods: PID control for motor control and Hector SLAM for navigation and mapping.

2.2.1. PID Control

The PID controller in this system is used to control the speed of both motors so that they can have synchronous velocity. PID was chosen because of its simplicity, robustness, and ability to produce desired responses from the closed-loop system [34], [35], [36], [37]. Moreover, a mathematical model of the plant is not required to determine its gain value [38]. PID tuning is done manually with the help of an interface connected to the microcontroller. Tuning is important for PID to get the right gain which produces good system responses [39], [40], [41]. This interface is created using MATLAB application to simplify the PID tuning of both motors, Fig. 4. This interface consists of the slider as an input for the set point value, K_P, K_I, K_D, and a graph of the system response between the set points with feedback. The slider is used to provide constant values and set points to the PID controller and response graphs are used to analyze the effect of each change constant and set point to feedback. The type of data communication used is Serial Peripheral Interface (SPI). This communication was bidirectional so that a collection of devices could receive and send data simultaneously. In this system, the implementation of this communication is that the interface application will send data in the form of set point, K_P, K_I, and K_D to the microcontroller. The microcontroller will process the data to produce a PID value and feedback that will later be sent to the interface application. The interface application will receive and display the data in the system response graph.

Fig. 5 shows the PID block diagram used to regulate the speed of the robot's motors. The control system consists of two subsystems to manage the speeds of the right and left motors. The control system operates by taking linear and or angular velocity as inputs. The feedback used is data from the motor speed sensors. The output from the motors is the speed of both motors. The control results for the two motors are not always the same (in different directions) due to the different input values received for the right and left motors.

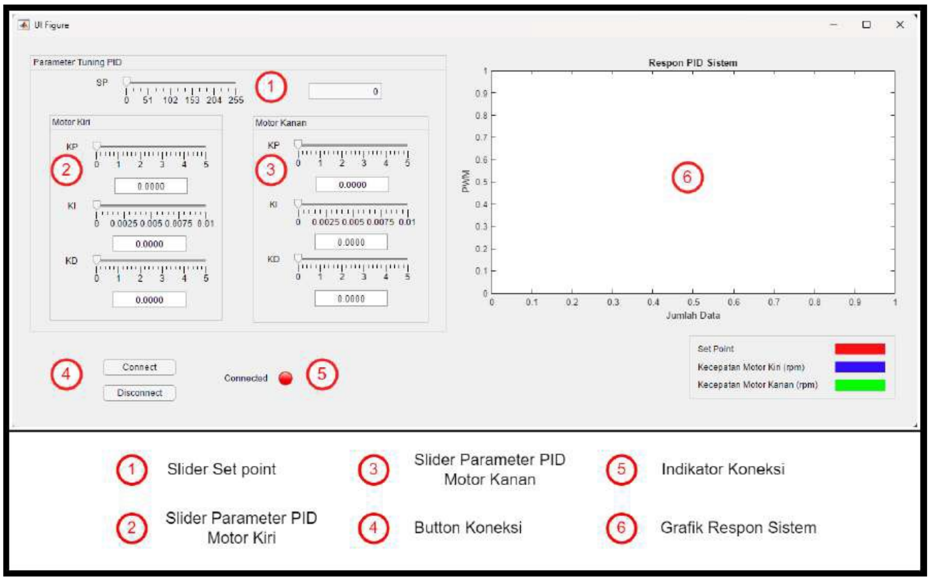


Fig. 4. PID tuning GUI

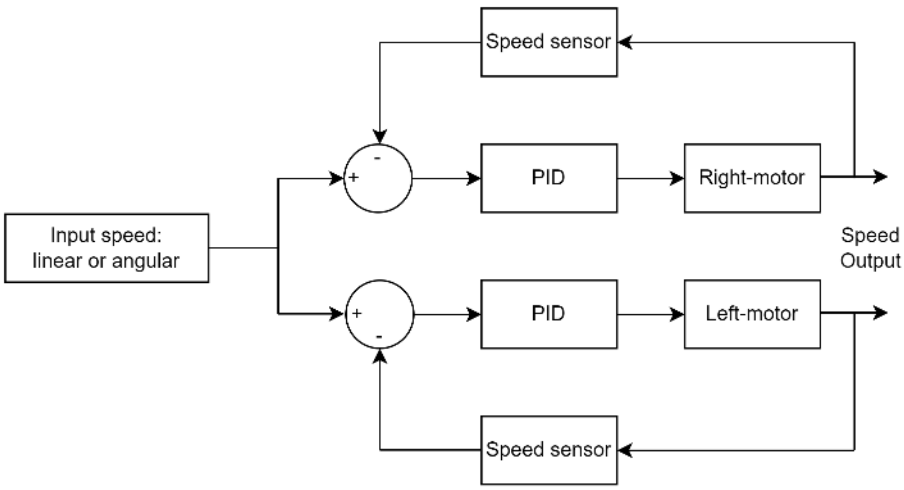


Fig. 5. PID controller block diagram

2.2.2. Hector SLAM and ROS

Navigation is determining a position and designing a path to reach a destination. In robotics, navigation is implemented as a tool to control and monitor the location of a moving robot. Navigation requires two essential components: localization and path planning [42]. Localization is needed to predict the robot's position within a map. One method that can be used is Adaptive Monte Carlo Localization (AMCL). This method is an algorithm that uses a particle filter to predict the position and orientation of the robot in a known environment. According to [43], [44], [45], path planning is the determination of a collision-free path in each environment. By knowing the starting point and the destination, this algorithm creates a path that the robot can follow. Additionally, the algorithm can incorporate an obstacle avoidance method, enabling the robot to reach its destination while avoiding obstacles in its surroundings.

Hector SLAM is one of the mapping algorithms that use LiDAR as the sensor and processes both the distance and angle of the object. Hector SLAM was chosen for its capability to produce precise 2D maps using LiDAR data, without relying on odometry—a significant advantage in environments where wheel odometry is unreliable. This method can give the robot position and create an occupancy map from some steps: scan matching, Bayes's rule, and model movement. Scan matching is the process of finding the transformation matrix (translation or rotation) from laser scanning. The result of this process is the estimation of the robot's position. The next step is Bayes's rule for occupancy maps which is a grid map. The last step is model movement which estimates the robot's position based on the previous position and the laser scanning data.

Fig. 6 shows the flow diagram to conduct the Hector SLAM. The system begins by initializing the necessary components. Next, it requires LiDAR data and motor speed data to be processed in the subsequent stages. The first step of the system involves communication between two devices, the NUC and the Jetson Nano, with the Jetson Nano acting as the ROS master. After that, the Jetson Nano and Teensy need to communicate to send and receive motor speed data. Once both communication processes are running, the next step is to perform SLAM to map the environment or room. To map the room, the robot needs to be manually moved to explore every part of it. This movement is controlled using the ROS package, teleop twist keyboard. Finally, once the room is fully mapped, the map is saved.

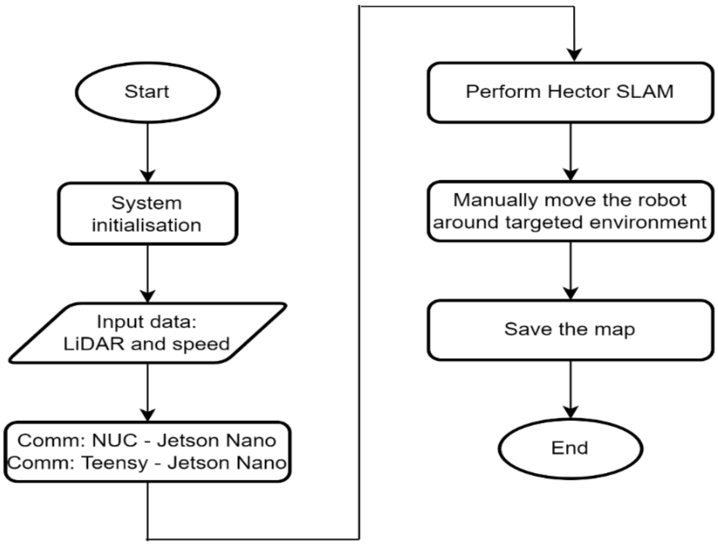


Fig. 6. Step by step to conduct the Hector SLAM

ROS frameworks provide libraries and tools for developing complex navigation algorithms [46], [47], [48]. An example of navigation implementation is the Navigation Stack in ROS, a collection of software packages that provide mobile robots with autonomous navigation capabilities. This stack includes tools for path planning, obstacle avoidance, localization, and map building. The key components of the Navigation Stack are the move_base package, AMCL, and the costmap. The Hector SLAM method requires several packages and commands to be run before it can be used to map the environment. The packages used are rplidar (depending on the type of LiDAR), Hector SLAM, and Rviz as visualization applications [49], [50].

The move_base package integrates various algorithms to enable the robot to move intelligently within its environment. It requires sensor data such as laser scans and odometry to generate and execute a navigation plan. The global planner component of move_base is responsible for creating a path from the robot's current position to the target location. It uses algorithms like Dijkstra's algorithm or A* search to ensure the path is collision-free. The local planner focuses on real-time obstacle avoidance and collision recovery. It adjusts the robot's speed and generates control commands to follow the global path while avoiding dynamic obstacles. The local planner relies heavily on the costmap package, which creates a map by assigning cost values to each cell based on the proximity of obstacles. The costmap is divided into two main parts: the global costmap, which handles static obstacles like walls and permanent structures, and the local costmap, which deals with dynamic obstacles such as moving objects. The configuration of these costmaps, including parameters like inflation radius, resolution, and update frequency, plays a crucial role in the robot's ability to navigate efficiently and safely.

Rviz is the visualization tool used to display the mapping and navigation processes. It shows the map being generated by the robot and the trajectory it follows. In Fig. 7, the frame coordinate transformation (TF) illustrates the relationship between the robot's position and the LiDAR sensor data. The green trajectory indicates the movement of the LiDAR as it scans the environment. Once the mapping is complete and the map is deemed accurate, it can be saved using the map_server package, which stores and provides the map for future navigation tasks.

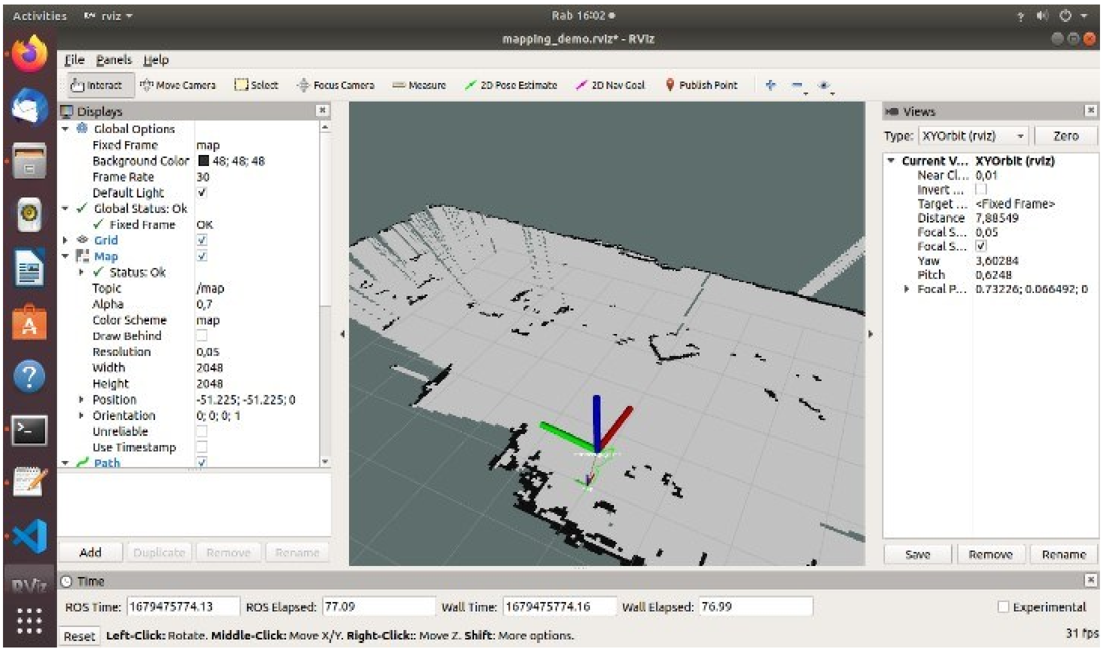


Fig. 7. Rviz visualizes Hector SLAM for mapping

3. RESULTS AND DISCUSSION

This research mainly focuses on the functionality of each system unit such as motor control and implementation of Hector SLAM in differential drive mobile Robot using ROS. The motors acquire a control system so the robot can maneuver across the arena with the desired velocity. The Hector SLAM has the role of mapping the environment without odometry. The robot used ROS as the operating system to combine and process every robot behavior that will be executed such as controlling the robot remotely and mapping the environment. Each of the system units requires testing of the variables that can affect the robot's performance when maneuvering and mapping results. Motor control needs the best response from the system to avoid losing speed in uneven terrain by providing the PID parameter value that produces the most robust response. The mapping process needs suitable linear and angular velocity to generate a map without changing the orientation of the map. There are two tests which are simulation and real-hardware testing.

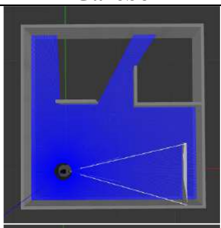
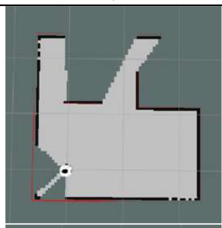
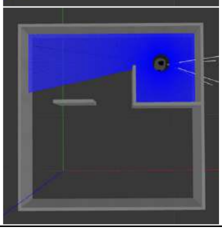
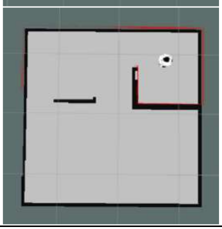
3.1. Simulation Testing

Simulation testing is crucial for evaluating the robot's capabilities in a controlled environment. The robot was modeled in three dimensions using Gazebo ROS, where the simulated environment closely mirrors real-world conditions. This includes accurate representations of physical properties, sensor inputs, and interaction dynamics. The simulation is divided into two main tests: SLAM and navigation, each targeting different aspects of the robot's functionality.

For the SLAM simulation, developers were able to test and optimize SLAM algorithms before applying them in a physical environment. This simulation saved time and costs, enabling the evaluation of complex and rare scenarios under controlled conditions. As shown in Table 1, the results from the Gazebo and Rviz simulators were summarized, demonstrating the accuracy and reliability of the SLAM algorithm in creating a detailed map of the environment.

The navigation test involved assessing the robot's behavior and response to commands within the simulation environment. Specifically, the test focused on the robot's ability to reach a designated target point while avoiding obstacles. The generated path, as shown in Table 2, met the desired criteria by passing through the center of the walls without collisions. The robot accurately followed the path, demonstrating effective navigation capabilities.

Table 1. SLAM simulation

Gazebo	Rviz	Remarks
		Starting condition
		Map created by a robot

3.2. Real Hardware Testing

3.2.1. Differential Drive Testing

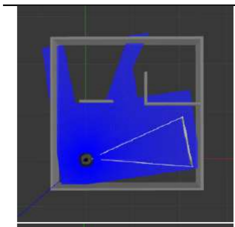
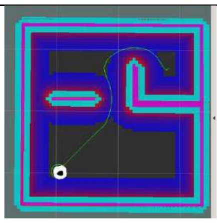
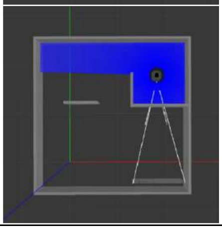
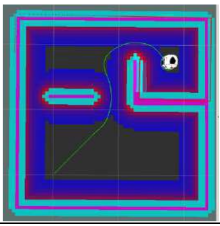
In the simulation testing, only the SLAM and navigation components were evaluated. However, in the real robot implementation, the PID control for the motors was also configured and tested. The differential drive system's performance hinges on two key aspects: tuning the PID parameters and testing the robot's maneuvers. Tuning the PID parameters provides an initial response, while maneuver testing ensures the robustness of the system under various conditions.

The PID tuning process was conducted iteratively to achieve the optimal balance between responsiveness and stability. The final tuning response of the PID control is shown in Fig. 8. The gain values for the proportional (K_P) and integral (K_I) terms were carefully selected based on the robot's performance during testing. The final K_P values for the right and left motors were 0.9333 and 0.9267, respectively, while the K_I values were 0.0026 for the right motor and 0.0022 for the left motor. The derivative (K_D) term was set to 0 for both motors, as it was observed that adding a derivative term did not contribute to performance improvements and sometimes led to instability in the control response of this tuning.

The choice of K_P and K_I values was influenced by the need for smooth and consistent motor speed control. The selected K_P values were high enough to ensure a quick response to changes in the input speed, while the K_I values were kept low to prevent excessive oscillations and to correct any steady-state errors over time. The K_D term was omitted because the differential drive system’s mechanical characteristics, such as friction and inertia, were already well-compensated by the proportional and integral terms.

After the PID tuning was completed, the next step was to test the robot's maneuvers. This test aimed to assess how well the input velocity commands translated into actual motor speeds, as displayed in ROS. Table 3 presents the results of the maneuver testing, where discrepancies between the input and output velocities were observed.

Table 2. Navigation simulation

Gazebo	Rviz	Remarks
		Starting condition
		Robots reach the destination

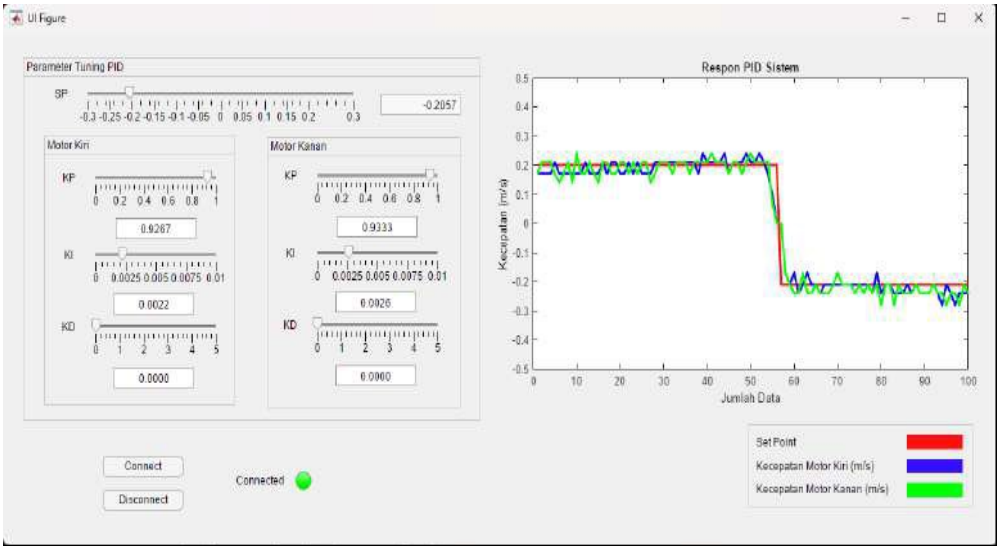


Fig. 8. PID tuning response

The tuning parameters of the PID controller played a crucial role in maintaining consistent motor speeds that aligned with the input commands. However, as indicated by the results in Table 3, the velocities of the left and right motors were not always uniform. Several factors contributed to this discrepancy, including environmental and mechanical influences. One significant factor was the uneven floor surface, which introduced variations in friction and traction experienced by the robot's wheels. This variation led to differences in the load on each motor, causing disparities in their output velocities. Additionally, the mechanical characteristics of the robot, such as differences in motor efficiency, gear ratios, or slight variations in wheel diameter, also contributed to these velocity differences.

To mitigate these issues in future implementations, several strategies could be considered. Improving the physical environment, such as ensuring a more even floor surface, would reduce friction-related discrepancies. Additionally, enhancing the mechanical design to ensure more uniform motor and gear performance could help achieve more consistent velocities. Furthermore, advanced control strategies, such as adaptive PID tuning or implementing a feedforward control mechanism, could be explored to dynamically compensate for these environmental and mechanical variations, ensuring that the motor speeds remain closely aligned with the desired input, thereby enhancing the robot's overall performance and stability.

Table 3. Robot maneuver testing

No.	Input Velocity (m/s)	Output Velocity (m/s)	
		Left Motor	Right Motor
1.	0.15	0.13	0.13
2.	0.20	0.17	0.20
3.	0.25	0.27	0.27
4.	0.30	0.34	0.38
5.	0.40	0.48	0.48

3.2.2. Mapping with Hector SLAM

The process of testing and data collection was carried out in an indoor environment. The dimensions of the environment were 2.4 × 2.4 meters Fig. 9. The purpose of this test was to provide evidence and analyze the effect of variations of linear and angular velocity on the result of the mapping process. The variations were conducted with three different values on both linear and angular.

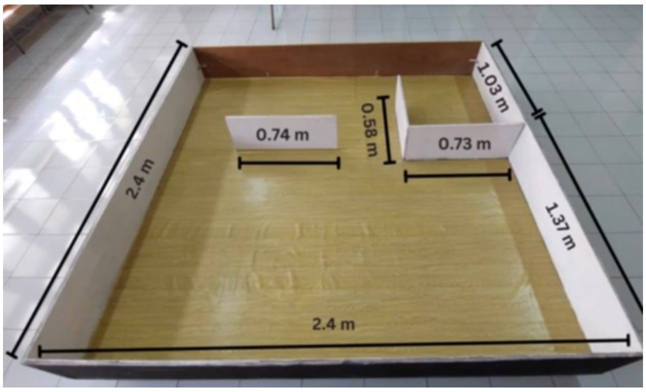


Fig. 9. The dimension of the robot's arena

In the linear velocity testing, the velocity values will be divided into three categories, slow, medium, and fast. The angular velocity used was the ideal angular speed which was 0.18 rad/s. The specified velocity values are 0.16 m/s, 0.23 m/s, and 0.5 m/s. The value set was classified using testing and chosen according to the result of the robot's capabilities. The slow category was the minimum velocity of the robot, the medium category was the ideal velocity of the robot, and the fast category was the maximum speed of the robot. Of the three velocities, medium and fast category values can produce maps that are relatively under the initial orientation of the robot. However, the slow category results in an inappropriate map due to the robot's shifting orientation causing the map to be overwritten. From the map result, the robot with linear speeds of 0.23 m/s and 0.5 m/s can be implemented into the system due to good mapping results. The results are listed in Table 4.

Next is the angular velocity variation. The test was carried out by giving three variations of angular velocity with a constant linear velocity of 0.23m/s. The angular velocity values will be divided into three categories namely, medium, fast, and very fast. The results are listed in Table 5. The selected velocity values are 0.18 rad/s, 0.29 rad/s, and 0.38 rad/s. The angular velocity of 0.18 rad/s was the ideal velocity of the robot, the angular velocity of 0.29 rad/s was the maximum velocity of the robot, and the angular velocity of 0.38 rad/s was the threshold speed. Ideal angular velocity was suitable to generate a map, whereas using a faster angular velocity will alter the orientation of the map and the robot. If the robot rotates swiftly, the robot cannot collect enough data to predict the map orientation. This condition occurs because the mapping and localization process uses LiDAR data which is sensitive to changes in angle. So, the angular velocity value that can be used in the system is 0.18 rad/s.

Maps created with Hector SLAM can produce relatively identical dimensions to the original. The way to measure the dimensions of the map results is to multiply the resolution (meters/pixels) with the map pixels. Each pixel represents 0.05 meters. The dimensions of the original arena are 2.40 × 2.40 meters, while the mapped dimensions are 2.30 × 2.30 meters. The dimension value error is 0.10 × 0.10 meters. The dimension of the mapped environment has an error value because the counted pixels are pixels that were free space. If the whole pixel is considered as the entire map, then the dimensions will be the same.

Table 4. Map results of linear velocity

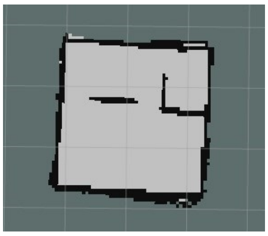
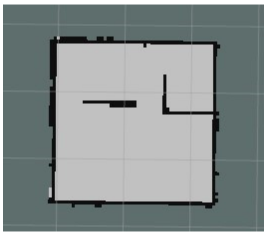
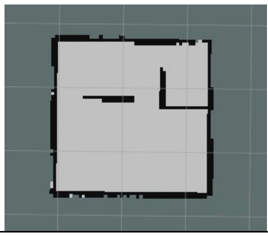
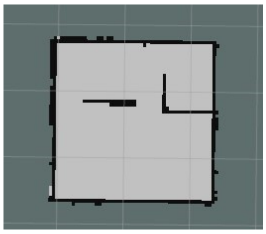
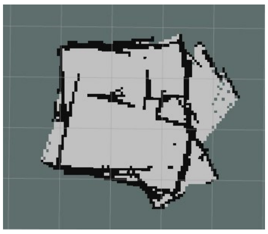
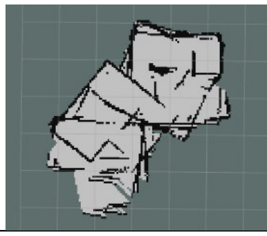
No.	Linear Velocity (m/s)	Map Result
1.	0.16	
2.	0.23	
3.	0.5	

Table 5. Map results of angular velocity

No.	Angular Velocity (rad/s)	Map Result
1.	0.18	
2.	0.29	
3.	0.38	

3.2.3. Navigation Testing

The navigation testing was divided into two parts: localization and path planning. The data is presented as images depicting the robot's path relative to the surrounding obstacles or walls. Accurate localization is critical, as the Adaptive Monte Carlo Localization (AMCL) algorithm determines the robot's position. The robot's position is represented by a particle cloud of arrows, where the pointed ends indicate the robot's orientation. These arrows initially scatter around predetermined points and orientations.

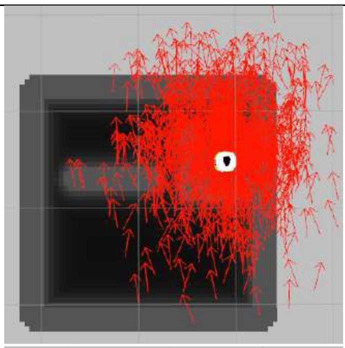
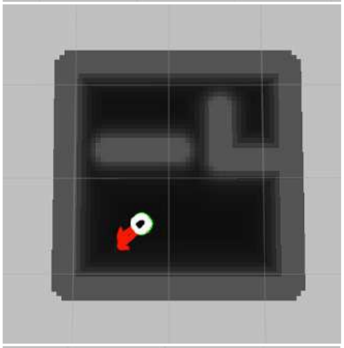
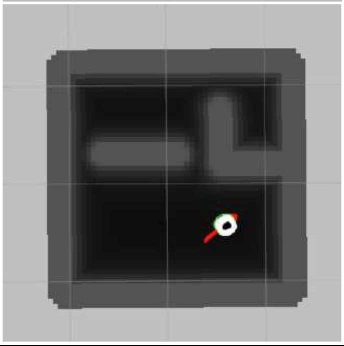
As shown in Table 6, AMCL accurately predicts the robot's position and orientation. Initially, the particle cloud is dispersed around the robot, but as the robot moves (translates and rotates), the cloud condenses and converges toward a single point, representing the robot's precise position and orientation. Over time and with more data, the particle cloud condenses further, resulting in a highly accurate localization.

For the path planning tests, the global planner parameters—cost factor and neutral factor—were manually tuned to optimize path generation relative to static obstacles or walls on the map. The goal was to create a smooth path that avoids collisions with obstacles. After manual tuning, the optimal values were determined to be 0.8 for the cost factor and 70 for the scaling factor. The resulting path is illustrated in Fig. 10.

The local planner was then tested by varying the `occdist_scale` parameter, with manual tuning determining the optimal value to be 0.02 Fig. 11. The testing environment featured small and tight spaces, necessitating a low `occdist_scale` value. This allowed the robot to navigate through narrow passages, such as doorways, by maintaining a closer distance from obstacles. However, a low `occdist_scale` is not universally ideal, as it may cause the robot to select paths through tight spaces, potentially leading to it getting stuck. This parameter should be adjusted based on specific user needs and environmental dynamics.

Finally, the robot was tested in a real-world environment Fig. 12. The robot successfully navigated to the target position while avoiding obstacles. Additionally, Rviz provided real-time visualization of both localization and path planning.

Table 6. AMCL localization test results

No	AMCL results	Remarks
1		Condition when the robot is first positioned on the map
2		Condition after the robot is moved on the map
3		Condition after the robot has been moved for quite a long time

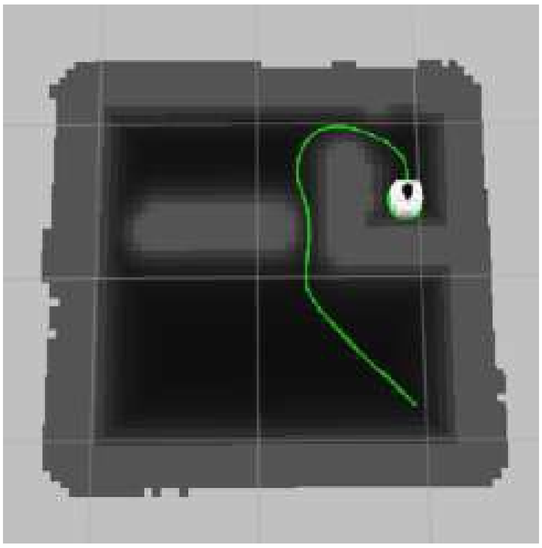


Fig. 10. Optimal path planning from manual tuning

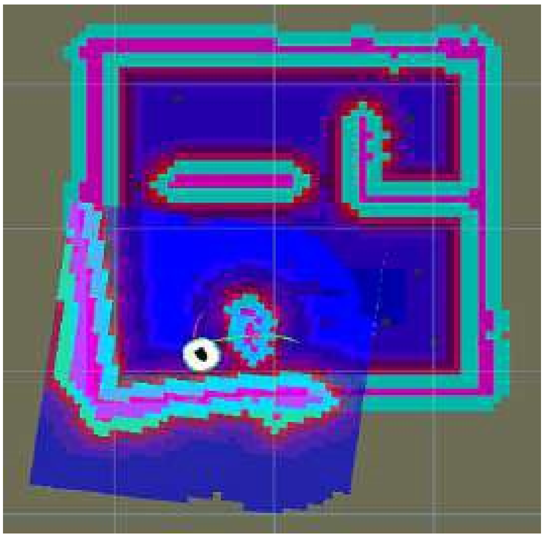


Fig. 11. Local planner testing

Fig. 12. Control and navigation testing

4. CONCLUSION

The simulation and implementation of control and navigation of the differential drive robot are already done. The PID tuning can give good results with the manual tuning through the developed MATLAB GUI. The optimal PID gains are $K_P = 0.9333$ and $K_I = 0.0022$ for the left motor $K_P = 0.9333$ and $K_I = 0.0026$ for the right motor. Hector SLAM produced accurate maps closely matching the original arena, proving effective in environments where odometry is unreliable. The robot successfully navigated to target points without collisions, confirming the system's overall functionality. Future work should include a comparative study with other SLAM algorithms and navigation strategies, as well as exploring enhancements in dynamic environments. Additionally, further investigation into the PID controller under varied conditions would provide a more comprehensive understanding of its impact on system performance.

Acknowledgments

This research was funded by HGR-Penelitian UNS No. 194.2/UN27.22/PT.01.03/2024.

REFERENCES

[1] Y. Liu, L. Wang, H. Huang, M. Liu, and C. Z. Xu, “A novel swarm robot simulation platform for warehousing logistics,” *in IEEE Int. Conf. Robot. Biomimetics*, pp. 2669–2674, 2018, https://doi.org/10.1109/ROBIO.2017.8324822.

[2] A. Ubaidillah and H. Sukri, “ Application of Odometry and Dijkstra Algorithm as Navigation and Shortest Path Determination System of Warehouse Mobile Robot,” *Journal of Robotics and Control (JRC),* vol. 4, no. 3, pp. 413-423, 2023, https://doi.org/10.18196/jrc.v4i3.18489.

[3] M. Wahdan, “Motion planning for autonomous mobile robots,” *AEJ - Alexandria Eng. J.,* vol. 44, no. 1, pp. 51–57, 2005.

[4] I. Reguii, I. Hassani, and C. Rekik, “Mobile robot navigation using planning algorithm and sliding mode control in a cluttered environment,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 166-175, 2022, https://doi.org/10.18196/jrc.v3i2.13765.

[5] K. Zhu and T. Zhang, “Deep Reinforcement Learning based Mobile Robot Navigation: A Review,” *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021, https://doi.org/10.26599/TST.2021.9010012.

[6] C. Zheng, R. Green, “Vision-based autonomous navigation in indoor environments,” *in Proceedings of the 2010 25th International Conference of Image and Vision Computing*, pp. 1-7, 2010, https://doi.org/10.1109/IVCNZ.2010.6148876.

[7] E. A. Oyekanlu et. al., “A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications,” *IEEE Access*, vol. 8, pp. 202312–202353, 2020, https://doi.org/10.1109/ACCESS.2020.3035729.

[8] N. Sariff, N. Buniyamin, “Ant colony system for robot path planning in a global static environment,” in *Proceedings of the 9th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE’10),* pp. 192–197, 2010, https://dl.acm.org/doi/10.5555/1938907.1938939.

[9] N. S. Abu, W. M. Bukhari, M. H. Adli, S. N. Omar, S. A. Sohaimeh, “A comprehensive overview of classical

and modern route planning algorithms for self-driving mobile robots," *Journal of Robotics and Control (JRC)*, vol. 3, no. 5, pp. 666-678, 2022, https://doi.org/10.18196/jrc.v3i5.14683.

[10] S. B. Marwanto, and R. D. Puriyanto, "IMU Sensor Based Omnidirectional Robot Localization and Rotary Encoder," *Control Systems and Optimization Letters*, vol. 1, no. 2, pp. 103-110, 2023, https://doi.org/10.59247/csol.v1i2.39.

[11] B. Miodrag, M. Lukić, J. Bajić, B. Dakić, and M. Vukadinović, "Hardware realization of autonomous robot localization system," in *2012 Proceedings of the 35th International Convention MIPRO,* pp. 146-150, 2012, https://ieeexplore.ieee.org/abstract/document/6240630.

[12] T. Mac, C. Copot, D. Tran, R. Keyser, "Heuristic approaches in robot path planning: A survey," *Robot. Auton. Syst.*, vol. 86, pp. 13-28, 2016, https://doi.org/10.1016/j.robot.2016.08.001.

[13] H. Y. Zhang, W. M. Lin, and A. X. Chen, "Path planning for the mobile robot: A review," *Symmetry (Basel).*, vol. 10, no. 10, 2018, https://doi.org/10.3390/sym10100450.

[14] S. Nurmaini and B. Tutuko, "Intelligent robotics navigation system: Problems, methods, and algorithm," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3711–3726, 2017, https://doi.org/10.11591/ijece.v7i6.pp3711-3726.

[15] M. Luˇcan, F. Duchoˇn, and M. Gulan, "Simultaneous Localization and Mapping on A Mobile Robot Platform," *in 23rd International Symposium on Measurement and Control in Robotics (ISMCR),* pp. 1–6, 2020, https://doi.org/10.1109/ISMCR51255.2020.9263729.

[16] I. A. Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual SLAM", *Expert Systems with Applications*, 205, 2022, https://doi.org/10.1016/j.eswa.2022.117734.

[17] S. Saat, W. N. Abd Rashid, M. Z. M. Tumari, and M. S. Saealal, "Hectorslam 2d mapping for simultaneous localization and mapping (slam), *in Journal of Physics: Conference Series,* vol. 1529, no. 4, p. 042032, 2020, https://iopscience.iop.org/article/10.1088/1742-6596/1529/4/042032/meta.

[18] J. Liu, Q. Sun, Z. Fan, and Y. Jia, "ToF LiDAR Development in Autonomous Vehicle," in *3rd Optoelectronics Global Conference (OGC)*, pp. 185–190, 2018, https://doi.org/10.1109/OGC.2018.8529992.

[19] Y. Wu and Z. Ding, "Research on Laser Navigation Mapping and Path Planning of Tracked Mobile Robot based on Hector SLAM," *in International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS),* vol. 3, pp. 59–65, 2018, https://doi.org/10.1109/ICIIBMS.2018.8549954.

[20] B. Zhang, M. Zhu, C. Lin, and D. Zhu, "Research on AGV Map Building and Positioning based on SLAM Technology," *in IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pp. 707–713, 2022, https://doi.org/10.1109/AUTEEE56487.2022.9994471.

[21] A. F. Olalekan, J. A. Sagor, M. H. Hasan, and A. S. Oluwatobi, "Comparison of Two SLAM Algorithms provided by ROS (Robot Operating System)," in *2nd International Conference for Emerging Technology (INCET),* pp. 1–5, 2021, https://doi.org/10.1109/INCET51464.2021.9456164.

[22] Z. Xuexi, L. Guokun, F. Genping, X. Dongliang, and L. Shiliu, "SLAM Algorithm Analysis of Mobile Robot based on LiDAR," *in Chinese Control Conference* (CCC), pp. 4739–4745, 2019, https://doi.org/10.23919/ChiCC.2019.8866200.

[23] S. Nagla, "2d hector slam of indoor mobile robot using 2d lidar," *in Int. Confe. on Power, Energy, Control and Transmission Systems* (ICPECTS), pp. 1-4, 2020, https://doi.org/10.1109/ICPECTS49113.2020.9336995.

[24] N. Yu and B. Zhang, "An improved hector slam algorithm based on information fusion for mobile robot," *in 5th IEEE Int. Conf. on Cloud Computing and Intelligence Systems* (CCIS), pp. 279–284, 2018, https://doi.org/10.1109/CCIS.2018.8691198.

[25] C. Nehate, R. Shinde, S. Naik, M. Aradwad, A. Bhurke, and F. Kazi, "Implementation and evaluation of slam systems for a mobile robot," *in EEE 18th India Council International Conference* (INDICON), pp. 1–6, 2021, https://doi.org/10.1109/INDICON52576.2021.9691761.

[26] M. M. J. Samodro, R. D. Puriyanto, and W. Caesarendra, "Artificial potential field path planning algorithm in differential drive mobile robot platform for dynamic environment," *International Journal of Robotics and Control Systems*, vol. 3, no. 2, pp. 161-170, 2023, https://doi.org/10.31763/ijrcs.v3i2.944.

[27] M. Alfiyan, and R.D. Puriyanto, "Mecanum 4 Omni Wheel Directional Robot Design System Using PID Method," *Journal of Fuzzy Systems and Control*, vol. 1, no. 1, pp. 6-13, 2023, https://doi.org/10.59247/jfsc.v1i1.27.

[28] H. Maghfiroh *et al*., "A Survey on Traction Motor and Its Prototyping Method for Electric Vehicle Application," *J. Electr. Electron. Inf. Commun. Technol*., vol. 05, no. 1, pp. 21–26, 2023, https://doi.org/10.20961/jeeict.5.1.71317.

[29] J. Agarwal, G. Parmar, R. Gupta, and A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsystem Technologies*, vol. 24, no. 12, pp. 4997-5006, 2018, https://doi.org/10.1007/s00542-018-3920-4.

[30] S. Ekinci, B. Hekimoğlu, and D. Izci, "Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor," *Engineering Science and Technology, an International Journal*, vol. 24, no. 2, pp. 331-342, 2021, https://doi.org/10.1016/j.jestch.2020.08.011.

[31] M. Khairudin, Efendi, N. Purwatiningsih, and W. Irawan, "The implementation of silicon controlled rectifiers for DC motor control," In *AIP Conference Proceedings AIP Publishing LLC*, vol. 1977, no. 1, p. 030035, 2018,

https://doi.org/10.1063/1.5042955.

[32] Q. I. Fatimah, R. Marselino, and A. Asnil, "Web-Based DC Motor Speed Design and Control," *Motiv. J. Mech. Electr. Ind. Eng.*, vol. 3, no. 3, pp. 101–112, 2021, https://doi.org/10.46574/motivection.v3i3.99.

[33] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an Open-source Robot Operating System", *in ICRA Workshop on Open Source Software*, vol. 3, no. 3, p. 5, 2009, http://lars.mec.ua.pt/public/LAR%20Projects/BinPicking/2016_RodrigoSalgueiro/LIB/ROS/icraoss09-ROS.pdf.

[34] R. R. Carmona, H. G. Sung, Y. S. Kim and H. A. Vazquez, "Stable PID Control for Mobile Robots," *in 15th International Conference on Control, Automation, Robotics and Vision* (ICARCV), Singapore, pp. 1891-1896,2018. https://doi.org/ 10.1109/ICARCV.2018.8581132.

[35] Q. Ariyansyah, and A. Ma'arif, "DC Motor Speed Control with Proportional Integral Derivative (PID) Control on the Prototype of a Mini-Submarine," *Journal of Fuzzy Systems and Control*, vol. 1, pp. 18-24, 2023, https://doi.org/10.59247/jfsc.v1i1.26.

[36] B. H. Nguyen *et al*., "Application of Genetic Algorithm for Optimizing Continuous and Discrete PID to Control Antenna Azimuth Position,"*Journal of Fuzzy Systems and Control*, vol. 2, no. 1, pp. 1-5, 2024, https://doi.org/10.59247/jfsc.v2i1.154.

[37] H. Maghfiroh, O. Wahyunggoro, A. I. Cahyadi, and S. Praptodiyono, "PID-hybrid tuning to improve control performance in speed control f DC motor base on PLC," *in 2013 3rd International Conference on Instrumentation Control and Automation* (ICA), pp. 233-238, 2013, https://doi.org/10.1109/ICA.2013.6734078.

[38] S. Istiqphara, A.U. Darajat, and M.F. Ferdous, "Movement Control of Three Omni-Wheels Robot using Pole Placement State Feedback and PID Control," *Journal of Fuzzy Systems and Control*, vol. 1, no. 2, pp. 44-48, 2023, https://doi.org/10.59247/jfsc.v1i2.36.

[39] F. Fahmizal, A. Priyatmoko, E. Apriaskar, and A. Mayub, "Heading control on differential drive wheeled mobile robot with odometry for tracking problem", *in 2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation* (ICAMIMIA), pp. 47-52, 2019, https://doi.org/10.1109/ICAMIMIA47173.2019.9223412.

[40] H. Maghfiroh, M. Nizam, and S. Praptodiyono, "PID optimal control to reduce energy consumption in DC-drive system," *Int. J. Power Electron. Drive Syst.*, vol. 11, no. 4, pp. 2164–2172, 2020, https://doi.org/ 10.11591/ijpeds.v11.i4.pp2164-2172.

[41] H. Maghfiroh, J. S. Saputro, C. Hermanu, M. H. Ibrahim, and A. Sujono, "Performance Evaluation of Different Objective Function in PID Tuned by PSO in DC-Motor Speed Control," *IOP Conf. Ser. Mater. Sci. Eng*., vol. 1096, no. 1, p. 012061, 2021, https://doi.org/ 10.1088/1757-899x/1096/1/012061.

[42] F. Fahmizal, M. Arrofiq, H. Maghfiroh, H.P. Santoso, P. Anugrah, and A. Molla, "Path Planning for Mobile Robots on Dynamic Environmental Obstacles Using PSO Optimization, *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika* (JITEKI), vol. 10, no. 1, pp. 166-172, 2024, https://doi.org/10.26555/jiteki.v10i1.28513

[43] X. Zhou, T. Bai, Y. Gao, and Y. Han, "Vision-based robot navigation through combining unsupervised learning and hierarchical reinforcement learning," *Sensors* (Switzerland), vol. 19, no. 7, pp. 1–23, 2019, https://doi.org/ 10.3390/s19071576.

[44] A. Molina-Leal, A. Gómez-Espinosa, J. A. Escobedo Cabello, E. Cuan-Urquizo, and S. R. Cruz-Ramírez, "Trajectory planning for a Mobile robot in a dynamic environment using an LSTM neural network," *Applied Sciences*, vol. 11, no. 22, p. 10689, 2021, https://doi.org/10.3390/app112210689.

[45] C. Wang and J. Mao, "Summary of AGV Path Planning," *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE),* pp. 332-335, 2019, https://doi.org/ 10.1109/EITCE47263.2019.9094825.

[46] I. R. Fahmi and D. J. Suroso, "A Simulation-Based Study of Maze-Solving-Robot Navigation for Educational Purposes," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 48–54, Aug. 2021, https://doi.org/ 10.18196/jrc.v3i1.12241.

[47] O. Y. Ismael, M. Almaged, and A. I. Abdulla, "Nonlinear Model Predictive Control-based Collision Avoidance for Mobile Robot," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 142–151, Jan. 2024, https://doi.org/ 10.18196/jrc.v5i1.20615.

[48] W. A. H. Sandanika, S. H. Wishvajith, S. Randika, D. A. Thennakoon, S. K. Rajapaksha, and V. Jayasinghearachchi, "ROS-based Multi-Robot System for Efficient Indoor Exploration Using a Combined Path Planning Technique," *Journal of Robotics and Control (JRC)*,vol. 5, no. 5, pp. 1241-1260, 2024, https://doi.org/10.18196/jrc.v5i5.22494.

[49] StefanKohlbrecher. hector slam. [Online]. Available: http://wiki.ros.org/hector slam.

[50] TheoKanning. rplidar. [Online]. Available: http://wiki.ros.org/rplidar.