

RESEARCH ARTICLE

Navigation System for Wheeled Robots Using SLAM GMapping and LiDAR

Muhammad Fasha Aqillah ¹, Muhammad Daffa Fadillah ^{*}, Erwin Susanto, Sony Sumaryo

¹ School of Electrical Engineering, Telkom University, Bandung, Indonesia

^{*}Corresponding author (muhammaddaffafadillah9@gmail.com).

Received: 15 November 2024 ; **Revised:** 4 February 2025 ; **Accepted:** 6 February 2025

Abstract

This project addresses the challenges of developing autonomous robots capable of operating in dynamic, unstructured environments by leveraging LiDAR (Light Detection and Ranging) technology and the SLAM (Simultaneous Localization and Mapping) algorithm. Utilizing the LiDAR A1 sensor, the robot can effectively map its surroundings and avoid obstacles, while the Raspberry Pi 4B serves as the central processing unit, enabling real-time processing of complex algorithms and facilitating communication between sensors and actuators. Precise motor control is achieved through an Arduino Mega, enhancing the robot's responsiveness to environmental changes. By integrating SLAM with LiDAR and robust motor control, this research aims to create a robot that autonomously perceives and navigates its environment, contributing to advancements in autonomous robotics, particularly in navigation and environmental mapping, and providing a practical solution for independent operation in complex settings.

Keywords: autonomous robot, LiDAR, raspberry pi, SLAM

I. INTRODUCTION

Autonomous robotics has been one of the highly developed research fields in recent years. One of the main challenges in the development of autonomous robots is the ability to navigate and operate in dynamic and unstructured environments, where the role of sensors and efficient computing systems becomes crucial [1]. Lidar (Light Detection and Ranging) technology has long been recognized as one of the best sensory solutions for autonomous robots, especially in terms of environment mapping and obstacle avoidance. Lidar uses laser light to measure distances to surrounding objects, generating data that can be used to create two-dimensional (2D) or three-dimensional (3D) maps of the robot's surroundings [2], [3]. Lidar A1, as one of the Lidar products often used in robotics, has the ability to provide highly accurate and detailed data, thus allowing robots to map the environment with high precision [4]. Lidar has the ability for mapping, but not only that, Lidar is also key in the implementation of SLAM (Simultaneous Localization and Mapping) algorithms. SLAM allows the robot to determine its position in the map being built, which is the basis of autonomous navigation [5].

In this project, the Lidar A1 is used to support SLAM, which runs on a Raspberry Pi 4B as the computing center. The Raspberry Pi 4B was chosen for its ability to run complex algorithms in real-time, as well as its extensive software ecosystem support, such as ROS (Robot Operating System), which allows integration with various sensors and actuators [6]. The Raspberry Pi 4B is not only responsible for running the SLAM algorithm, but also for managing communication between sensors and actuators through various protocols such as UART, I2C, or SPI [7]. Raspberry Pi 4B has the advantage of its quad-core processor and RAM, making it possible to handle heavy computational tasks such as SLAM algorithms and sensor data processing, without an external computer [8][9].

The combination of powerful computational capabilities and flexibility in communication makes Raspberry Pi 4B the right choice to integrate this entire robotic system [10]. Arduino Mega is used as the microcontroller in this project, specifically to control the MG513 motor through the L298N motor driver. Arduino Mega was chosen because it has a large number of input/output pins, allowing detailed control of the various actuators on the robot [11]. The MG513 motor, which is equipped with an encoder, allows for high-accuracy measurement of speed and position, which is crucial in ensuring precise robot movements that comply with the navigation commands given by the Raspberry Pi [12].

The L298N motor driver plays an important role in controlling the DC motor in this robot. As an H-bridge, this driver allows the motor to rotate in two directions and regulates its speed through a PWM (Pulse Width Modulation) signal sent by the Arduino Mega [13]. The use of appropriate motor drivers is essential to ensure that the robot's motion can be controlled efficiently and responsive to changing environmental conditions [14]. The robot utilizes a tank chassis, which provides stability and the ability to traverse a variety of difficult terrains. The tank chassis allows for more even load distribution and improves traction on various surfaces, making it ideal for outdoor as well as indoor applications with varying terrain conditions [15]. This design also supports high maneuverability, which is necessary for navigation in environments full of obstacles [16]. The integration between sensors, actuators, and computing systems in this robot requires efficient communication and good synchronization [17]. The Raspberry Pi 4B and Arduino Mega communicate via a reliable protocol to ensure that data from the Lidar sensor is processed in a timely manner and translated into accurate motion by the motors [18].

II. MECHANICAL AND ELECTRICAL SYSTEM DESIGN

A. Block Diagram System

This system robot setup that includes a LiDAR A1 sensor, Raspberry Pi 4B, Arduino Mega 2560, L298N motor driver, and two DC motors equipped with encoders. The LiDAR A1 sensor performs continuous 360° scans to detect

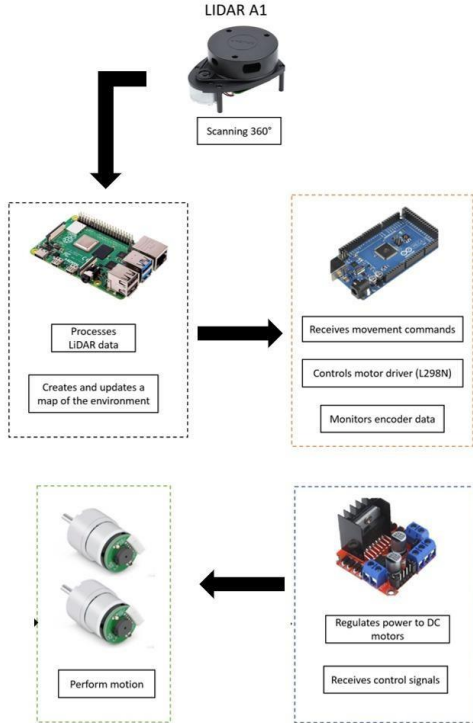


Fig. 1. Block diagram system.

Using ROS (Robot Operating System) with the Gmapping package, the Raspberry Pi 4B enables SLAM (Simultaneous Localization and Mapping). Gmapping utilizes the LiDAR data to create a 2D map of the environment, updating it in real-time as the robot moves. This map allows the robot to interpret its surroundings and recognize obstacles, guiding the operator in planning safe movement paths. Gmapping's spatial data is critical for understanding the environment and aids in manual navigation.

To execute movement commands, the Raspberry Pi sends instructions to the Arduino Mega 2560, which is connected to the L298N motor driver. The Arduino processes these commands to set the speed and direction of the DC motors while collecting real-time data from motor encoders. The encoder feedback is essential for accurate odometry, enabling the system to monitor the robot's precise position and orientation as it moves.

Odometry, derived from encoder data, is vital for controlling movement accurately. The Raspberry Pi uses this feedback to adjust motor speeds and directions, ensuring the robot follows the operator's planned path while compensating for any deviations. By combining LiDAR-based mapping from Gmapping with precise motor control via odometry, this setup provides the robot with spatial awareness and accurate movement capabilities, supporting autonomous navigation in complex environments.

B. Hardware Design

A tank chassis is the basic structure of a robot designed to provide stability and maneuverability in a variety of difficult terrains. The chassis is designed to distribute the load evenly and improve traction on various surfaces, such as rocky soil, sand, or other uneven surfaces. In this project, the tank chassis is used to support all robotic components, including the Lidar, Raspberry Pi, Arduino Mega, and MG513 motors, ensuring that the robot can move with high stability even in rough environments [17]. The following is a schematic of the system used in the mobile robot circuit.

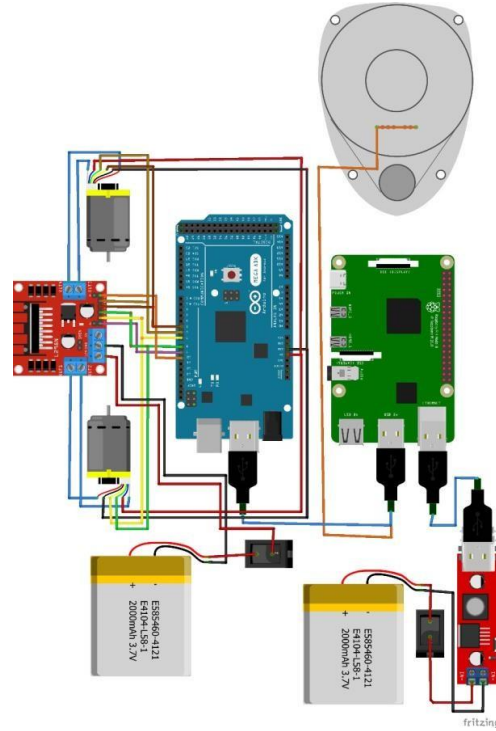


Fig. 2. Wiring system.

The main advantage of a tank chassis is its ability to maneuver at low speed and high stability, which is very useful in navigating uneven or obstacle-filled terrain. The track design of the chassis tank allows the robot to overcome obstacles that may be difficult for a robot with conventional wheels to pass. This makes tank chassis an ideal choice for outdoor robotics applications or in conditions that require high mobility and the ability to navigate unstructured terrain [18].

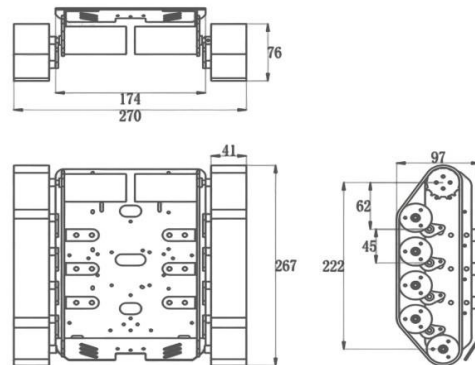


Fig. 3. 2D chassis design

C. Robot Automatic Movement

To initiate the robot, the first direction of movement is at an angle of 0°. The robot will run when it gets a command from the user and can stop at any time the user wants to stop the robot. To change the angular orientation of this robot, there are several conditions. When the robot's front distance is smaller than the specified threshold, the robot will change its orientation in the direction of movement by adding an angle of 90°. Meanwhile, when the robot has lost reference to the right wall, the robot will reduce its orientation in the direction of movement by 90°.

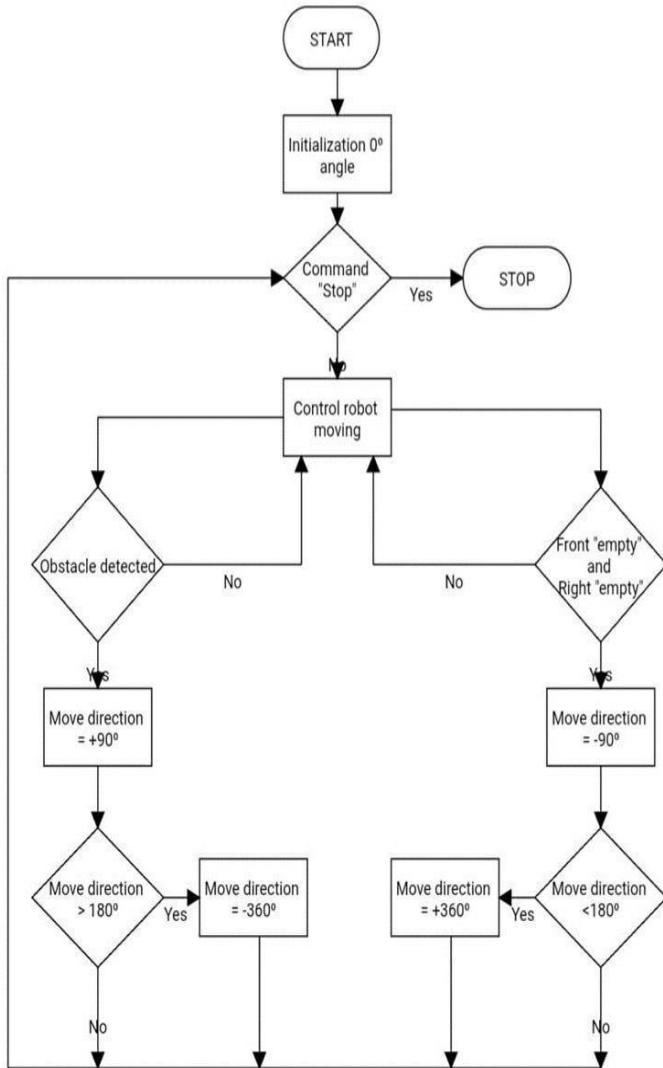


Fig. 4. Flowchart automatic movement.

The flowchart represents an algorithm for controlling a robot in an environment with obstacles. The algorithm starts with initializing the robot's angle to 0 degrees and then checks for a "Stop" command. If the command is received, the robot stops. Otherwise, the robot continues to move and checks for obstacles. If an obstacle is detected, the robot changes its direction to 90 degrees. After moving 90 degrees, the robot checks if its current direction is greater than 180 degrees.

If it is, the robot changes direction to -360 degrees. If the current direction is not greater than 180 degrees, the robot continues its current direction. If no obstacle is detected, the robot checks if the front and right paths are empty. If both paths are empty, the robot changes its direction to -90 degrees. If either path is not empty, the robot checks if its current direction is less than 180 degrees. If it is, the robot changes direction to +360 degrees. If the current direction is not less than 180 degrees, the robot continues its current direction.

III. ODOMETRY CALCULATION

Distance between wheels (L): 178.6 mm (0.1786 meters). Wheel radius (R): 22.25 mm (0.02225 meters). PPR (Pulses per Revolution): 920. Chassis width: 175.9 mm Odometry Calculation Steps:

A. Calculating the Distance Traveled by Each Wheel

Suppose the encoders on the left and right wheels generate the number of pulses N_L and N_R respectively during a given period.

$$d_L = \frac{N_L \cdot 2\pi R}{P} \quad (1)$$

$$d_R = \frac{N_R \cdot 2\pi R}{P} \quad (2)$$

B. Calculating Total Displacement and Change in Orientation

Total distance traveled by the chassis center (d_c)

Change of orientation ($\Delta\theta$)

C. Calculating Position Change (x,y)

Change of position in the axis x (Δx):

$$\Delta x = d_c \cdot \cos\left(\theta + \frac{\Delta\theta}{2}\right) \quad (3)$$

Change of position in the axis y (Δy):

$$\Delta y = d_c \cdot \sin\left(\theta + \frac{\Delta\theta}{2}\right) \quad (4)$$

D. Robot's New Position

New position on the axis x (x_{after}):

$$x_{after} = x_{before} + \Delta x \quad (5)$$

New position on the axis y (y_{after}):

$$y_{after} = y_{before} + \Delta y \quad (6)$$

New robot orientation (θ after):

$$\theta_{after} = \theta_{before} + \Delta\theta \quad (7)$$

Calculation Example: With $N_L = 460$ and $N_R = 500$ and other parameters, the calculation result is:
Distance traveled by the left wheel:

$$d_L = \frac{420 \cdot 2\pi \cdot 0,02225}{920} = \frac{64,46}{920} \approx 0,07006 \text{ meter} \quad (8)$$

Distance traveled by the right wheel:

$$d_R = \frac{500 \cdot 2\pi \cdot 0,02225}{920} = \frac{69,64}{920} \approx 0,0757 \text{ meter} \quad (9)$$

Total distance:

$$d_c = \frac{0,07006 + 0,0757}{2} \approx 0,07288 \text{ meter} \quad (10)$$

Change of orientation:

$$\Delta\theta = \frac{0,0757 - 0,07006}{0,1786} \approx 0,0316 \text{ radian (1,81 degrees)} \quad (11)$$

Change of position in the axis x :

$$\begin{aligned} \Delta x &= 0,07288 \cdot \cos\left(0 + \frac{0,0316}{2}\right) \\ &\approx 0,07288 \cdot \cos(0,0158) \\ &\approx 0,07288 \cdot 0,99987 \\ &\approx 0,07287 \text{ meter} \end{aligned} \quad (12)$$

Change of position in the axis y :

$$\begin{aligned} \Delta y &= 0,07288 \cdot \sin\left(0 + \frac{0,0316}{2}\right) \\ &\approx 0,07288 \cdot \sin(0,0158) \\ &\approx 0,07288 \cdot 0,0158 \\ &\approx 0,00115 \text{ meter} \end{aligned} \quad (13)$$

New position of the robot:

$$\begin{aligned} x_{after} &= x_{before} + 0,07287 \text{ meter} \\ y_{after} &= y_{before} + 0,00115 \text{ meter} \\ \theta_{after} &= \theta_{before} + 0,0316 \text{ radian (1.81 degrees)} \end{aligned} \quad (14)$$

IV. RESULT AND ANALYSIS

This test is carried out indoors by determining the starting point of the robot and the end point of the robot as shown in Fig.6. The encoder data generated is useful for knowing the position of the robot on the mapping based on the odometry value generated. The encoder functions to calculate motor rotation and convert it into odometry data to provide information about the distance traveled by the mobile robot and the resulting movement. Each motor has an encoder, so that the accuracy of the movement and displacement of the mobile robot can be known well based on the data it has.



Fig. 5. Robot testing area.

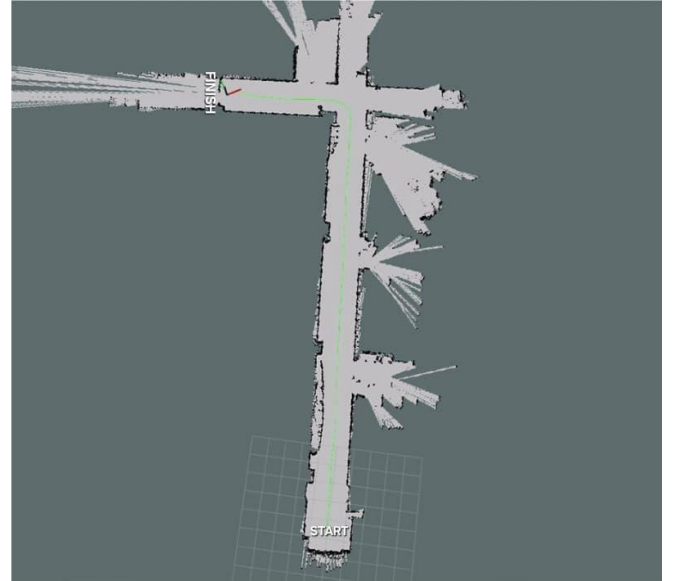


Fig. 6. Output mapping by mobile robot.

The figure 6 mapping of the movement of a robot within an indoor environment, where its path is represented by a green line. Starting from the bottom center of the image, the robot proceeded straight upward through a narrow corridor. Upon reaching the top section of the image, it made a left turn and continued moving until it reached its final position in the top-left corner. The path showcases the robot's ability to navigate accurately within the confined space. Surrounding the path, black points and lines represent the robot's laser scan data, capturing the structural features of the

environment. These scans indicate that the robot moved through a structured indoor corridor, with obstacles and openings detected on both sides. This visualization highlights the robot's efficiency in mapping and navigation, ensuring that it can avoid obstacles and follow a precise trajectory to its destination.

V. CONCLUSION

We developed an intelligent guide system using a 2D LiDAR A1 and designed its structure, which is mounted on a tank chassis. The system utilizes a Raspberry Pi 4B and the Hector SLAM algorithm to generate a 2D map of the surrounding environment and plan navigation routes. It is capable of real-time obstacle detection and avoidance, assisting visually impaired individuals in both indoor and outdoor settings. Experimental results show that the system offers accurate mapping and localization, providing reliable navigation assistance. Odometry is a fundamental aspect of robotic navigation, providing critical data on a robot's movement in terms of distance and orientation. The odometry system employed in this analysis utilizes encoders with a resolution of 920 pulses per revolution (PPR) and a wheel radius of 22.25 mm. This setup allows for precise measurements of the displacements of both the left and right wheels. For instance, in a real-world application, autonomous vehicles use similar odometry systems to navigate complex environments, relying on accurate wheel rotation data to maintain their intended paths. The calculations derived from the odometry data indicate that the left wheel recorded 460 pulses, translating to a distance of approximately 0.07006 meters, while the right wheel, with 500 pulses, covered about 0.0757 meters. This disparity in wheel movement is crucial for understanding the robot's trajectory and ensuring it can adjust its path accordingly..

REFERENCES

- [1] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020.
- [2] D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini, and D. G. Sorrenti, "Global visual localization in lidar-maps through shared 2d-3d embedding space," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4365–4371.
- [3] Pang, Conglin, Liqing Zhou, and Xianfeng Huang. "A Low-Cost 3D SLAM System Integration of Autonomous Exploration Based on Fast-ICP Enhanced LiDAR-Inertial Odometry." *Remote Sensing* 16.11 (2024): 1979.
- [4] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A.-a. Agha-Mohammadi, "Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 421–428, 2020.
- [5] M. U. Khan, S. A. A. Zaidi, A. Ishtiaq, S. U. R. Bukhari, S. Samer, and A. Farman, "A comparative survey of lidar-slam and lidar based sensor technologies," in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. IEEE, 2021, pp. 1–8.
- [6] D. Talwar and S. Jung, "Particle filter-based localization of a mobile robot by using a single lidar sensor under slam in ros environment," in *2019 19th international conference on control, automation and systems (ICCAS)*. IEEE, 2019, pp. 1112–1115.
- [7] Wang, Zixiang, et al. "Research on autonomous robots navigation based on reinforcement learning." *2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC)*. IEEE, 2024.
- [8] Yang, Xiaozhe, Pizheng Chen, and Huiting Lu. "Design of an Improved Autonomous Tracked Vehicle Based on SLAM Algorithm." *Artificial Intelligence, Medical Engineering and Education*. IOS Press, 2024. 366-373.
- [9] Abeideh, Ahmad, Abdul Rahman Zamrik, and Wisam Elmasry. "Autonomous Driving Robot for Indoor 2D Mapping." *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2024.
- [10] H. D. Ghael, L. Solanki, and G. Sahu, "A review paper on raspberry pi and its applications," *International Journal of Advances in Engineering and Management (IJAEM)*, vol. 2, no. 12, p. 4, 2020.
- [11] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, "A systematic literature review on prototyping with arduino: Applications, challenges, advantages, and limitations," *Computer Science Review*, vol. 40, p. 100364, 2021.
- [12] Z. Zhang, X. Cheng, B. Yang, and D. Yang, "Exploration of indoor barrier-free plane intelligent lofting system combining bim and multisensors," *Remote Sensing*, vol. 12, no. 20, p. 3306, 2020.
- [13] L. Yin, F. Wang, S. Han, Y. Li, H. Sun, Q. Lu, C. Yang, and Q. Wang, "Application of drive circuit based on I298n in direct current motor speed control system," in *Advanced laser manufacturing technology*, vol. 10153. SPIE, 2016, pp. 163–169.
- [14] J. Li, "Research on dc motor driver in automobile electric power steering system," in *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. IEEE, 2020, pp. 449–453.
- [15] S. Chittaliya, H. Anghan, and U. Vaghani, "A methodological study to analyze and design the car chassis," *International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume*, vol. 9, 2021.
- [16] C. A. Pedraza Yepes, J. Gonz'alez-Coneo, C. A. Mancera-Trejos, L. E. Florez-Sarmiento, G. A. TORRES DIAZ, and C. A. Gonzalez Olier, "Design and construction of tank-chassis and lifting structure for centrifugal pump hl260 m powered by a diesel engine," *ARNP Journal of Engineering and Applied Sciences*, 2020.
- [17] H. Tang, "Autonomous mobile robot indoor navigation using multi-sensor integration," 2023.
- [18] Rajendran, Rakesh, et al. "Design and comparative analysis of multicontrol line tracking robot using Arduino, at mega 2560 and m-bot for multilevel training purpose imparting quality education." *AIP Conference Proceedings*. Vol. 3216. No. 1. AIP Publishing, 2024.
- [19] M. Kelemen, F. Filakovsk'y, and P. Feren'cik, "Adaptable mobile robot for rough terrain," *Acta Mechatronica*, vol. 4, no. 4, pp. 11–15, 2019.
- [20] X. Zhao, J. Yang, Y. Zhong, C. Zhang, and Y. Gao, "Study on chassis leveling control of a three-wheeled agricultural robot," *Agronomy*, vol. 14, no. 8, p. 1765, 2024.