

Autonomous navigation system for hexa-legged search and rescue robot using LiDAR

Aris Budiarto, Sarosa Castrena Abadi, Naufaldo

Department of Automation and Mechatronics Engineering, Politeknik Manufaktur Bandung, Bandung, Indonesia

Article Info

Article history:

Received Jul 26, 2023

Revised Dec 7, 2023

Accepted Dec 18, 2023

Keywords:

Hexapod
Navigation
Robot operating system
Robotics
Simultaneous localization and mapping

ABSTRACT

This study proposes an autonomous navigation system for hexapod robots, promising in complex rescue scenarios. The system is tested in simulations under three environments: rocky, cracked flooring, and inclined surfaces. Utilizing light detection and ranging (LiDAR) and simultaneous localization and mapping (SLAM), the robot recognizes positions and constructs environmental maps. Implemented via robot operating system, the research successfully applies navigation and mapping using hector_slam. LiDAR mapping yields satisfactory accuracy, with average errors of 0.21% for general mapping and 5.34% for circular paths. Within a 2-meter range, navigation achieves good accuracy, averaging 1.2% error on the x-axis and 0.011% on the y-axis during linear motion. Navigational repeatability improves, with reliable results showing an average error of 4.33 cm on the x-axis and 0.5 cm on the y-axis when returning to starting points. Arena testing with varied obstacles demonstrates successful obstacle traversal. However, in the second test, limitations in hardware, notably the Raspberry Pi 4 CPU usage reaching 97% during navigation, hindered reaching the third target.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Aris Budiarto

Department of Automation and Mechatronics Engineering, Politeknik Manufaktur Bandung

Jl. Kanayakan 21, Dago, Bandung, Jawa Barat, Indonesia

Email: aris@ae.polman-bandung.ac.id

1. INTRODUCTION

The hexa-legged robot, a six-legged marvel, has captivated widespread interest in recent years owing to its innovative design. This robotic system derives its leg morphology from biological creatures, a biomimetic approach aimed at enhancing adaptability across a spectrum of challenging terrains [1]–[5]. The unique leg configuration, inspired by natural biomechanics, endows the hexa-legged robot with unparalleled stability and versatility. Research indicates that in comparison to quadrupedal or four-legged counterparts, the hexa-legged robot demonstrates superior stability during locomotion. This heightened stability is attributed to its utilization of a static walking technique, in contrast to the dynamic walking techniques employed by other legged robots [6]. The static walking technique minimizes control complexity, enabling the robot to traverse complex terrains more efficiently. By mimicking the steady and reliable locomotion observed in certain biological organisms, this innovative robotic design holds promise for enhanced adaptability, stability, and efficiency, making it an ideal candidate for applications in diverse fields such as search and rescue operations, exploration of rugged terrains, and tasks requiring precise and stable movement in unpredictable environments.

The autonomous navigation system represents a critical paradigm in robotics, empowering robots to navigate from one point to another, regardless of whether the destination is familiar or uncharted. This

intricate system integrates multiple components to facilitate seamless movement. Firstly, perception becomes paramount, necessitating the incorporation of sensors that enable the robot to perceive its environment comprehensively. These sensors provide crucial data inputs, aiding the robot's understanding of its surroundings. Secondly, localization emerges as a pivotal aspect, allowing the robot to determine its precise position within the environment. This spatial awareness is achieved through sophisticated localization algorithms, ensuring accurate positioning in real-time. Furthermore, the system incorporates recognition capabilities, enabling the robot to make intelligent decisions autonomously. This decision-making process involves analyzing the gathered information, identifying obstacles or optimal paths, and formulating strategies to achieve its objectives effectively. Lastly, motion control emerges as the final link in this chain of operations, wherein the robot translates its decisions into physical actions. Through the actuation of precise motions, the robot executes its planned movements, responding to the environment's dynamics. Thus, the Autonomous Navigation System amalgamates perception, localization, recognition, and motion control, orchestrating a harmonious interplay of functionalities that enable robots to navigate autonomously with efficiency, adaptability, and intelligence [6]–[9].

The unpredictable and hazardous conditions arising from natural disasters in regions like Indonesia's ring of fire necessitate the deployment of advanced robotic solutions for efficient search and rescue operations [10], [11]. The use of a hexa-legged robot, capable of maneuvering through complex terrains, can significantly improve the effectiveness of rescue missions. This paper presents an autonomous navigation system for such a robot, utilizing light detection and ranging (LiDAR) technology to enhance perception, localization, and obstacle detection capabilities [12]–[14].

The central focus of this research endeavors to advance the implementation of an autonomous navigation system specifically customized for the hexapod robot. Building upon prior accomplishments utilizing Navstack_Pub and Hector_SLAM for mapping while employing LiDAR as the primary sensor, notable accuracy and repeatability were demonstrated within controlled environments. However, this study aims to further enhance this implementation by addressing synchronization challenges associated with encountered hardware limitations during subsequent trials. The overarching aim is to bolster the adaptability and reliability of the navigation system tailored for the hexapod robot, particularly emphasizing its performance in challenging scenarios like search and rescue operations amidst dynamic and hazardous conditions.

2. METHOD

This research encompasses the creation of an autonomous navigation system tailored for a hexa-legged search and rescue robot. The system distinguishes itself by its methodical integration of perception, localization, recognition, and motion control components. Through this cohesive framework, the robot acquires the ability to independently navigate and execute search and rescue operations with a high degree of efficiency. This study not only advances the field of robotics but also underscores the system's practical significance in various scenarios, including both familiar and unfamiliar environments [6], [7].

2.1. System overview

Figure 1 serves as a comprehensive illustration depicting the primary operational processes of the hexa-legged robot. At its core, the system relies on a main program running on the robot operating system (ROS), employing LiDAR for navigation, dynamic motors for leg movements, and an inertial measurement unit (IMU) for precise positioning. The robot configuration comprises a Raspberry Pi functioning as the central controller, 20 dynamixel Servos coordinated by a dedicated driver, and sensors including LiDAR and IMU, as showcased in Figure 1(a). Additionally, the robot employs a Raspberry Pi 4-based control system as its primary controller, enabling both automatic and manual control via a laptop connected to the Raspberry Pi through WiFi. The graphical user interface leverages the rviz application, providing a visual representation of the environment scanned by the LiDAR, as demonstrated in Figure 1(b). This integrated system design ensures robust functionality and adaptability, allowing the robot to perform autonomously or under manual control across various operational settings.

The propulsion system of the robot consists of a total of 20 Dynamixel servo motors, with 18 dedicated to the legs and 2 for the gripper and lifter components. These Dynamixel servo motors are smart servos, allowing each one to be controlled with just a single data cable and read by the computer through the U2D2 driver. Additionally, the robot is equipped with two sensors: LiDAR, which enables navigation scanning, and an inertial measurement unit (IMU) to determine the robot's position accurately. The robot's operation can be fully automated or manually controlled through a PC connected to the Raspberry Pi via a wireless connection. For manual control, the Teleop Keyboard is utilized over an SSH connection, while the Rviz application is employed to visualize the map generated from LiDAR scanning.

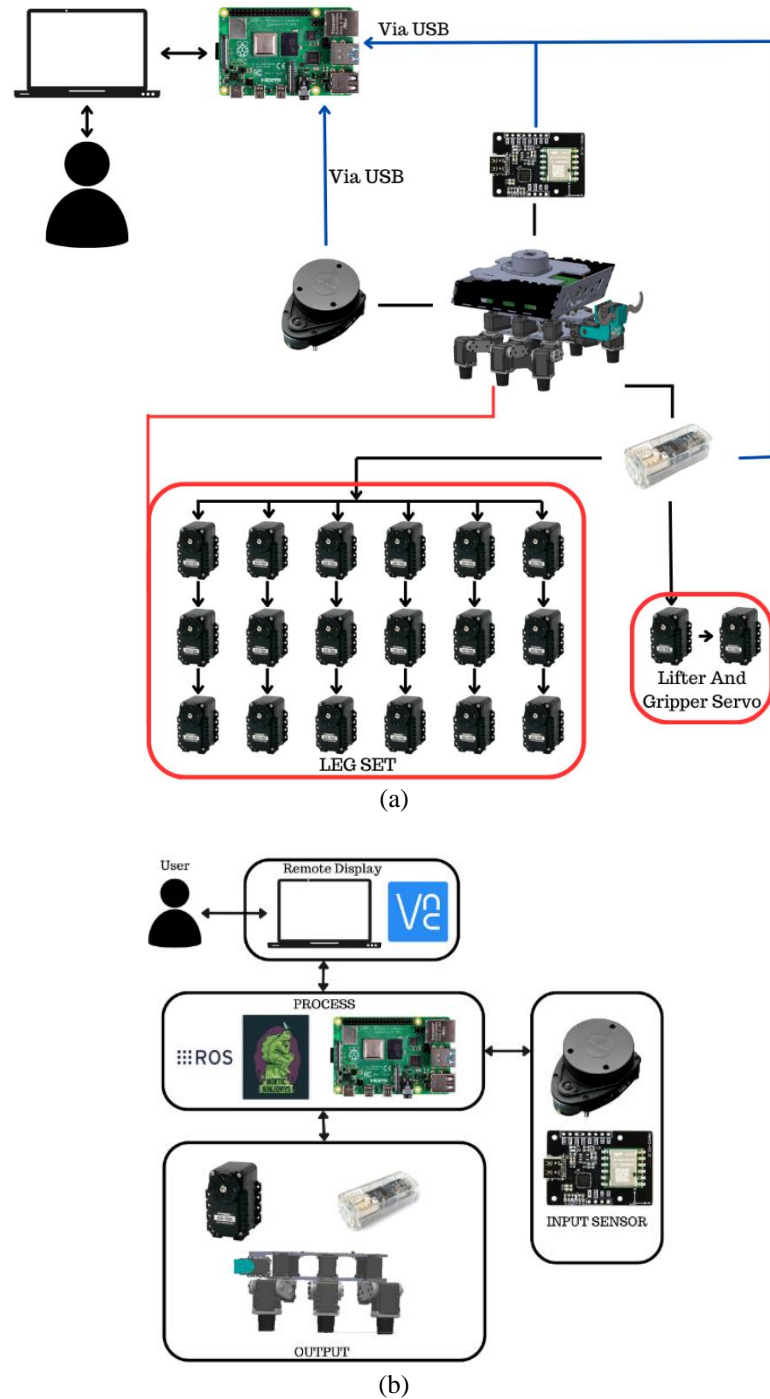


Figure 1. Illustration of hexapod System: (a) robot construction diagram and (b) general system diagram

Furthermore, in Figure 2, the design layout for the robot is elaborated, presenting both an overhead view as shown in Figure 2(a) and a side perspective view as shown in Figure 2(b). This meticulous design encapsulates the seamless integration of essential components. The strategic arrangement of the Dynamixel servo motors, LiDAR, and IMU sensors exemplifies the meticulous engineering employed in the robot's construction. The placement of these components not only ensures optimal functionality but also contributes to the robot's overall aesthetic coherence. The layout emphasizes the efficient utilization of space, fostering a compact yet efficient design. The positioning of the 18 servo motors dedicated to leg movements, along with the two motors allocated for the gripper and lifter modules, reflects a balanced distribution that augments the robot's stability and versatility. The LiDAR and IMU sensors are strategically positioned to enable

comprehensive environmental perception and precise localization, enhancing the robot's navigational capabilities in various scenarios. This detailed design overview underlines the thoughtful consideration given to both the technical requirements and the visual appeal, culminating in a robot that embodies efficiency, functionality, and a visually appealing form factor.

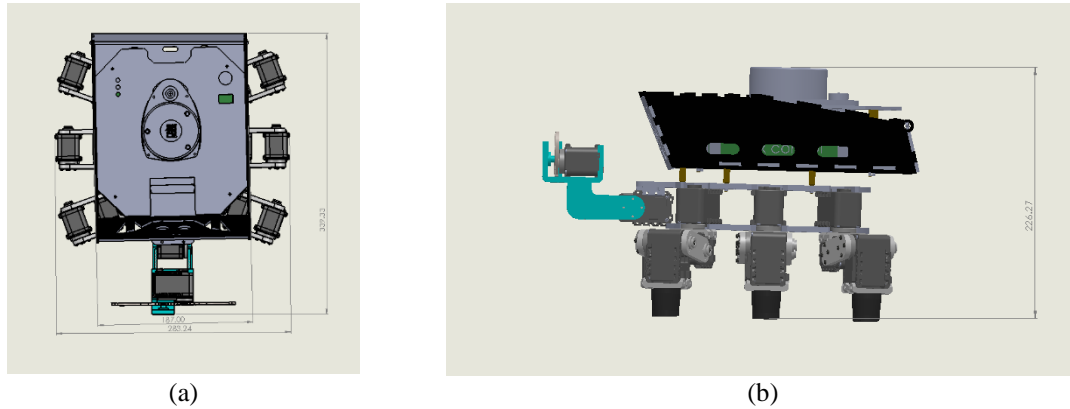


Figure 2. Robot design hexapod: (a) overhead view and (b) side perspective view

2.2. Leg movement system

The design of the subsystem is to design the movement method of the six-legged robot in such a way that the movement of the robot becomes dynamic and is not limited to certain movements. Inverse Kinematics is used to allow the position of the robot toes to reach the desired point and allow the robot legs to move to different positions. This Inverse Kinematics method processes the coordinate data of the desired endpoint and the length of the robot leg through geometry analysis, then generates an angle to move at each leg joint. The leg structure of the six-legged robot uses an insect leg structure consisting of 3 joints and 3 bone parts, namely the coxa, femur, and tibia as depicted in Figure 3. Inverse Kinematics is used to find the angle of the coxa, femur, and tibia joints so that the tip of the robot leg can reach the desired end-point position [15]–[18].

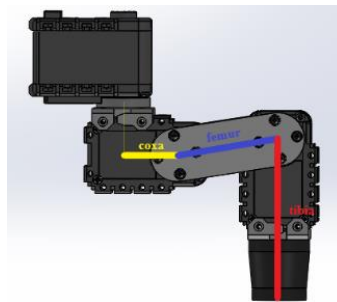


Figure 3. Hexapod leg structure

When driving the robot's steps, step motion is used, so the step motion is smooth. The trajectory is based on a curve formed by a 3rd-order polynomial equation. The gait pattern is then used to maintain the robot's balance during movement. This pattern is a sequential arrangement of each robot leg so that the robot can move dynamically [15]–[18].

The gait pattern is used to maintain the balance of the robot while moving. This pattern is a sequential order of each robot leg so that the robot can move dynamically and remain stable. Types of gait patterns that are often used include tripod gait, ripple gait and wave gait as depicted in Figure 4. In the tripod gait pattern, the step pattern is made using three feet to tread and three feet to step as depicted in Figure 4(a). Then in the ripple gait pattern, two robot legs are used alternately to step as depicted in Figure 4(b). While in the wave gait pattern, one leg of the robot is used alternately to step as depicted in Figure 4(c). Tripod gait, ripple gait, and wave gait patterns [19]–[21].

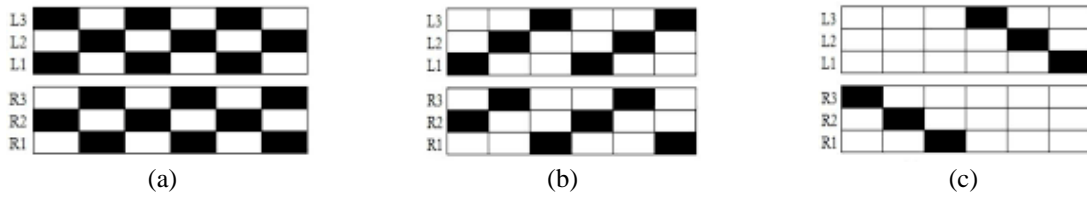


Figure 4. Gait Patterns: (a) tripod gait, (b) ripple gait, and (c) wave gait

2.3. Mapping system

For the mapping system, the hector_SLAM mapping package serves as the foundation, enabling the creation of a comprehensive environmental map based on the data acquired from the LiDAR sensor scans. This map delineates the surroundings of the robot with precision as depicted in Figure 5. Concurrently, the Teleop twist Keyboard is employed to govern the movement of the robot, providing an intuitive and interactive means of control. By leveraging these combined technologies, the robot gains the capability to navigate through its environment while simultaneously building a spatial understanding of its surroundings, which is imperative for effective autonomous movement and decision-making in search and rescue operations.

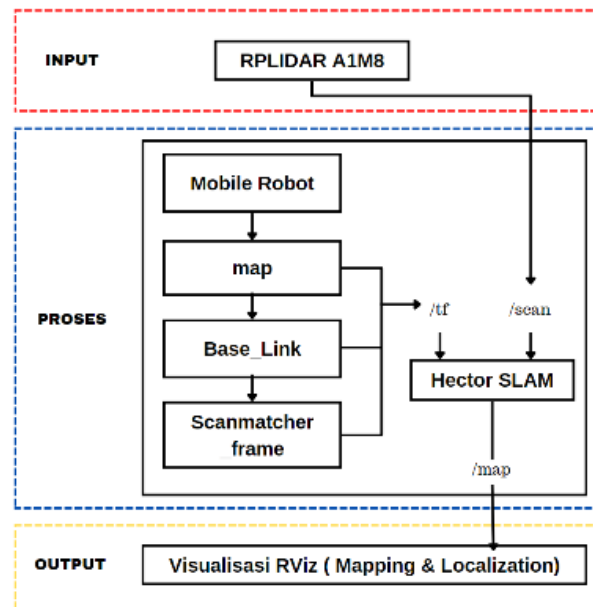


Figure 5. Hector SLAM block diagram system

The Hector SLAM algorithm is a scan matching-based technique that converts scans into localized coordinate frames by leveraging position estimates from the LIDAR system. The scans are transformed into a cloud point representation of endpoint positions, taking into account platform orientation estimates and combined values. Pre-processing steps, such as point reduction or outlier removal, can be applied to the cloud point based on the specific scenario. In the proposed approach, filtering is solely performed using the z-coordinate of the endpoint positions, allowing only endpoints within the desired scanning plane threshold to participate in the scan matching process [22].

The technique presented in Figure 6 encompasses bilinear filtering of the occupancy grid map, aiming to interpolate the value of point P_m . This method allows for approximating both the occupancy value $M(P_m)$ and the gradient $\nabla M(P_m) = \left(\frac{\partial M}{\partial x}(P_m), \frac{\partial M}{\partial y}(P_m) \right)$ using the four closest integer coordinates $P_{00..11}$ as depicted in Figure 6(a). Through linear interpolation along the x- and y-axis, the approximation for $M(P_m)$ is derived:

$$M(P_m) \approx \frac{y - y_0}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01}) \right) + \frac{y_1 - y}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{10}) + \frac{x_1 - x}{x_1 - x_0} M(P_{00}) \right)$$

The derivatives can be approximated by :

$$\frac{\partial M}{\partial x}(P_m) \approx \frac{y - y_0}{y_1 - y_0} (M(P_{11}) - M(P_{01})) + \frac{y_1 - y}{y_1 - y_0} (M(P_{10}) - M(P_{00}))$$

$$\frac{\partial M}{\partial y}(P_m) \approx \frac{x - x_0}{x_1 - x_0} (M(P_{11}) - M(P_{10})) + \frac{x_1 - x}{x_1 - x_0} (M(P_{01}) - M(P_{00}))$$

It's important to note that the sample points or grid cells of the map are arranged on a regular grid, each positioned at a distance of 1 in map coordinates from each other. This regularity in the grid simplifies the presented equations for gradient approximation, ensuring a more streamlined calculation process. Figure 6(b) visualizes the resulting occupancy grid map and its spatial derivatives, offering a comprehensive insight into the filtering process's outcomes [22], [23].

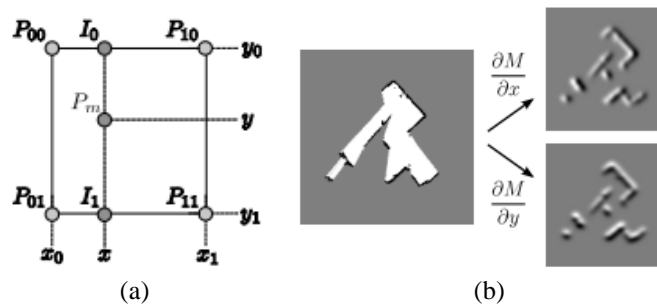


Figure 6. Bilinear filtering of the occupancy grid map (a). Point P_m is the point whose value shall be interpolated and (b) Occupancy grid map and spatial derivatives

2.4. Navigation system

The navigation system used is the ROS Navigation Stack package. This package is simply a program using the concept of global and local planners. It simultaneously creates a navigation stack called `move_base` and provides a standard interface for other plug-ins to interact with as depicted in Figure 7. This interface is based on the A to B navigation model, which means it has a starting pose and a destination pose.

The ROS navigation stack package encompasses a seamless integration of pivotal components meticulously designed to enhance path planning and facilitate precise robot navigation. At its core, this system harmoniously incorporates the global planner, local planner, global costmap, and local costmap modules. These cohesive elements operate synergistically, endowing the robot with the capability to execute global path planning with remarkable efficiency and safety, particularly in well-charted environments. The global planner orchestrates high-level route determination, while the local planner meticulously hones trajectories, adapting to real-time obstacles and constraints. In parallel, the dynamic interplay of the global and local costmaps furnishes the robot with a nuanced environmental representation, critical for astute obstacle avoidance and overall secure, streamlined movement. This holistic ROS navigation stack package significantly augments the robot's navigational prowess in familiar settings, underscoring its potential for diverse applications, including those demanding meticulous and safe global path planning.

The global planner is equipped with the Dijkstra algorithm, allowing it to compute the shortest path while considering obstacles and movement costs. On the other hand, the local planner utilizes the dynamic window approach (DWA) algorithm to generate responsive and secure local paths based on real-time sensor data. The `global_costmap` and `local_costmap` play a crucial role as map representations, providing obstacle and movement cost mapping for the entire environment and the immediate surroundings of the robot, respectively [24]–[28].

The seamless integration and collaboration among these components facilitate the robot's ability to navigate effectively and reach its target destination efficiently while avoiding obstacles and ensuring safety. This integrated system empowers the robot to autonomously navigate through known environments with

optimized path planning and responsive control, ensuring efficient and reliable performance in various scenarios. The presented navstack_pub system demonstrates the successful integration of state-of-the-art algorithms and mapping techniques within the Robot Operating System (ROS), providing a robust and versatile solution for robot navigation and path planning. The system's capabilities have been validated through comprehensive simulations and real-world experiments, showcasing its efficacy and potential for a wide range of applications in autonomous robotic systems [23], [29]–[31].

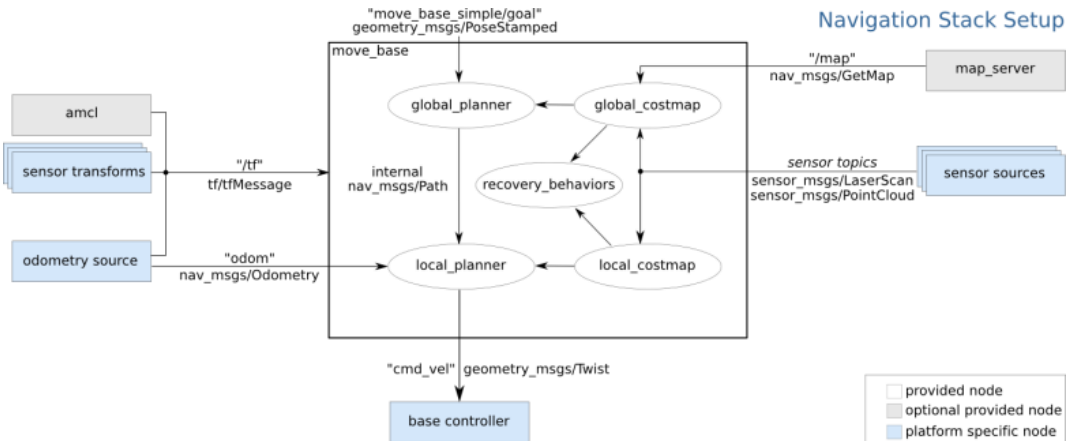


Figure 7. Navstack pub architecture system

3. RESULTS AND DISCUSSION

The robot, endowed with robust capabilities, adeptly navigates through diverse terrain, confidently overcoming a spectrum of potential obstacles. This resilience is seamlessly orchestrated through the teleop twist keyboard, a tool that empowers intuitive control. The mapping process has been equally accomplished with meticulous precision, underscored by the accurate positioning of the robot through the LiDAR technology. As depicted in Figure 8, the already built robot stands as a testament to these achievements, highlighting its tangible embodiment and readiness for complex real-world scenarios. The culmination of these achievements underscores the robot's capacity to surmount challenges effectively, showcasing its readiness for complex real-world scenarios.

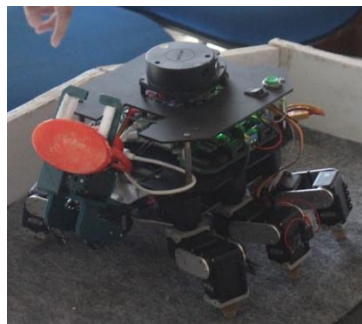


Figure 1. Hexapod robot

3.1. Hector slam

With this program, the hexapod robot can be controlled through the teleop twist keyboard, and its motion data and status will be used by hector_slam for environment mapping as depicted in Figure 9. The built map can be visualized through Rviz for further navigation and mapping purposes. The Hector_trajectory_server also provides information about the robot's trajectory, which can be useful for specific objectives. All communications between these nodes enable the hexapod robot to operate in a coordinated manner and efficiently carry out its tasks.

3.3. Mapping result

The testing of mapping results using hector_SLAM is meticulously evaluated against real-world conditions, with each pixel of the mapping result precisely representing an area of 2.5 cm. The comprehensive testing is carried out within a square area measuring 118×80 cm, featuring varying heights of 28 and 30 cm. These tests are conducted under three distinct conditions, rigorously examining the mapping efficacy and accuracy of the hector_SLAM algorithm in capturing and representing the physical environment.

3.3.1. Mapping result without obstacle

The test results encompass mapping outputs depicting unobstructed environments. These mapping results are illustrated within boxed sections, where obstacles are intentionally absent. This controlled scenario allows for a focused examination of the algorithm's ability to accurately capture and reproduce spatial features without the influence of obstacles. To delve into the finer details of these results, please refer to Table 1 for a comprehensive tabulated overview. Additionally, Figure 11 visually portrays the mapping outcomes, providing a vivid representation of the algorithm's performance in a controlled setting devoid of obstacles. This comprehensive presentation underscores the algorithm's capacity to faithfully render environmental features and patterns even in the absence of hindrances.

Mapping Value (cm)	Real Value (cm)	Error (%)
117.5	118	0.42
80	80	0.00
Error Average (%)		0.21

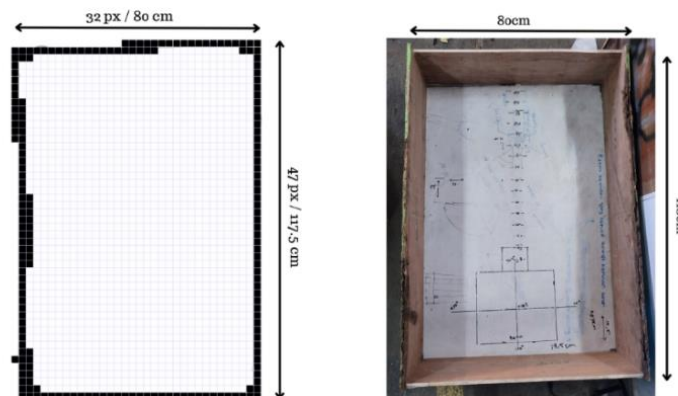


Figure 11. Mapping result without obstacle

3.3.2. Mapping result with 1 obstacle

Within the context of mapping results featuring a single obstacle, the focus shifts to scenarios where an individual obstacle is introduced. These mapping outcomes are presented within delineated sections, with the presence of a solitary obstacle serving as a test parameter. This controlled configuration allows for a meticulous assessment of the algorithm's ability to accurately map the environment while accommodating an obstacle. For a detailed breakdown of these results, kindly refer to Table 2, which offers a structured presentation of the hector_SLAM performance under this specific testing condition. Moreover, Figure 12 visually encapsulates the mapping outcomes, providing an illustrative depiction of the algorithm's efficacy in handling scenarios with a single obstacle. Through this comprehensive visualization, the algorithm's capability to seamlessly incorporate obstacles into its mapping process is effectively highlighted.

3.3.3. Mapping result with 2 obstacle

In the examination of mapping results involving dual obstacles, the investigation expands to scenarios where two distinct obstacles are introduced. These mapping outputs are distinctly delineated within designated sections, accounting for the presence of two obstacles as part of the assessment parameters. This controlled experimental setup enables a detailed evaluation of the algorithm's performance in accurately mapping environments featuring multiple obstacles. For an intricate breakdown of these outcomes, please

consult Table 3, which systematically presents the hector_SLAM algorithm's performance in the context of dual obstacle scenarios. Furthermore, Figure 13 visually encapsulates the mapping results, providing an insightful portrayal of the algorithm's competence in navigating and representing environments marked by the presence of two obstacles. Through this visual representation, the algorithm's proficiency in handling multi-obstacle scenarios becomes palpably evident.

Table 2. Mapping with 1 obstacle

Mapping Value (cm)	Real Value (cm)	Error (%)
117.5	118	0.42
77.5	80	3.13
32.5	33	1.52
40	40	0.00
35	35.5	1.41
15	11.5	30.43
Error Average (%)		6.15

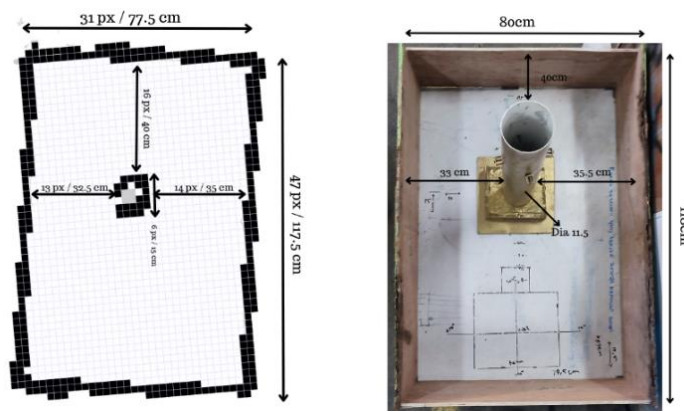


Figure 12. Mapping with 1 obstacle

Table 3. Mapping with 2 obstacle

Mapping Value (cm)	Real value (cm)	Error(%)
117.5	118	0.42
77.5	80	3.13
17.5	16.5	6.06
32.5	35	7.14
72.5	75	3.33
90	89.5	0.56
10	9	11.11
Error Average (%)		4.54

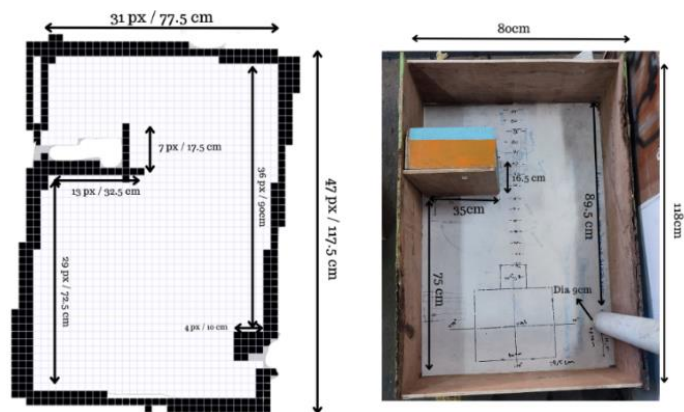


Figure 13. Mapping with 2 obstacle

3.4. Navigation result

The testing of the navigation system is comprehensively executed through a series of evaluations, encompassing various key aspects. These evaluations entail an assessment of the accuracy of position transitions in relation to the ground truth, the repeatability of selected paths, the determination of the maximum achievable range, and an evaluation of the path planning efficiency. This robust testing regimen provides a holistic appraisal of the navigation system's capabilities across different parameters. The navigation tests are methodically carried out within a controlled indoor environment, facilitated by the installation of a precise 5-meter long reference line as depicted in Figure 14. This reference line significantly aids in accurate measurements, further enhancing the precision and reliability of the conducted tests.



Figure 14. Navigation system testing

The implementation navigation testing in this arena is conducted to verify the successful execution of the robot within a designed arena, which simulates various road conditions and predefined paths in the main program. By commanding the robot to navigate obstacles such as cracked paths, inclined surfaces, rocks, and marbles, denoted as targets 1, 2, and 3, as depicted in Figure 15. This comprehensive testing approach ensures that the robot's implementation aligns with the intended functionality across diverse terrain challenges and predefined routes.

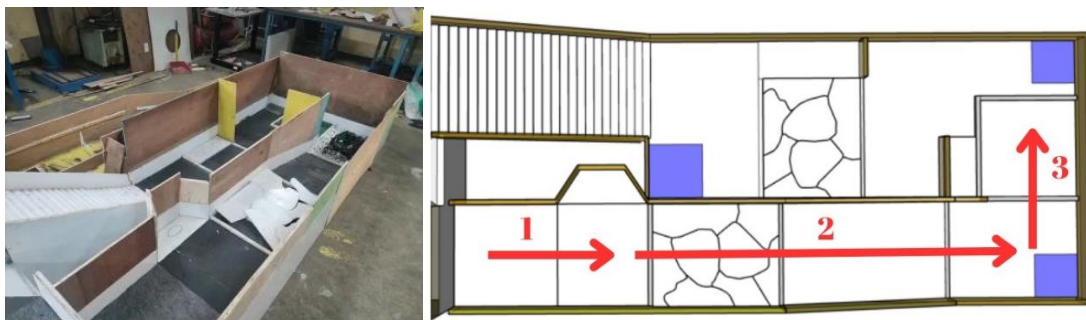


Figure 15. Implementation navigation system testing

3.4.1. Navigation accuracy result

Accuracy navigation testing constitutes a fundamental aspect of the evaluation process. This testing methodology involves supplying coordinate inputs to the system as can be seen in Figure 14, subsequently prompting the robot to navigate towards the designated coordinate points. This orchestrated movement is meticulously observed, and measurements of the resulting values are recorded within the Rviz platform. These recorded values are then methodically compared with the corresponding ground truth values.

To present the findings in a structured manner, the accuracy navigation results are succinctly captured in a dedicated table, specifically Table 4. This table encompasses target coordinates, denoted in meters, alongside the resultant values in both the x and y axes. Additionally, the calculated errors in both the x and y axes are thoughtfully incorporated within the same table, providing a comprehensive depiction of the accuracy achieved by the navigation system.

Table 4. Accuracy navigation

Goal Coordinate	Real Position (Meter)	Error x (%)	Error y (%)
(0.5,0)	(0.5,0)	0	0
(1,0)	(1,0)	0	0
(1.5,0)	(1.55,0.01)	3.33	0.01
(2,0)	(2.08,0.01)	4	0.01
(0,-0.5)	(0,-0.52)	0	0.02
(0,0.5)	(0,0.53)	0	0.03
Error Average (%)		1.22	0.011

3.4.2. Navigation reaptibility result

The assessment of repeatability is a pivotal facet of the evaluation process, focusing on the system's consistent performance over multiple iterations. This testing methodology involves consistently inputting the same coordinates to the robot as can be seen in Figure 14. Consequently, the robot executes repetitive movements, and the degree of deviation in movement for each iteration is keenly observed and meticulously analyzed.

To concisely present the results of the repeatability testing, all findings are succinctly compiled within Table 5. This table intricately documents essential parameters, including the repeatability turn, the designated goal coordinates, the robot's actual position during each repetition, and the corresponding errors observed in the x and y axes. The comprehensive compilation of these metrics within the same table provides a detailed and insightful perspective into the navigation system's repeatability performance.

Table 5. Navigation repeatability

Repeatabily Turn	Goal Coordinate	Real position	Error x (cm)	error y (cm)
0	(1,0)	(1.05,0.0)	5	0
0	(0,0)	(-0.07,0.01)	7	1
1	(1,0)	(1.03,0.1)	3	1
1	(0,0)	(-0.3,0.0)	3	0
2	(1,0)	(1.06,0.0)	6	0
2	(0,0)	(0.02,-0.01)	2	1
Error Average (cm)			4.33	0.5

3.4.3. Navigation implementation result

The results of the implementation testing affirm the robot's capability to navigate effectively within the simulated conditions, encompassing cracked paths, inclined surfaces, as well as obstacles like rocks and marbles as can be seen in Figure 15. These challenges are emblematic of real-world scenarios, and the robot adeptly maneuvers through them. The outcomes of this testing are meticulously documented in Table 6, providing a comprehensive overview of the robot's successful navigation performance under these simulated conditions. This comprehensive analysis attests to the system's reliability and its potential to navigate through diverse challenges, vital for its practical application in various scenarios.

Table 6. Navigation implementation result

Repeatability	Target	Obstacle	Goal Coordinate	Real position	Passthrough obstacle	Error
0	1	Carpet	(0.5,0)	(0.5,0.02)	Yes	0.1
0	2	Cracked paths, inclined surface, rocks	(2.3,0)	(2.45,0.05)	Yes	0.1
0	3	Rocks, marble	(2.4,0.5)	(2.50,0.51)	Yes	0.055
1	1	Carpet	(0.5,0)	(0.60,0.01)	Yes	0.055
1	2	Cracked paths, inclined surface, rocks	(2.3,0)	(2.3,0.02)	Yes	0.01
1	3	Rocks, marble	(2.4,0.5)	(2.35,0.20)	No	0.175

4. CONCLUSION

The implementation of the navigation system on the hexapod robot has been successfully achieved using Navstack_Pub, utilizing the mapping results from Hector_SLAM. The mapping process employed LiDAR as the sole sensor for mapping and localization. The LiDAR mapping results, with a pixel resolution of 2.5cm, exhibit an average error of 0.21%. For circular or rounded surfaces, the average error is slightly higher at 5.34%.

The accuracy of the navigation system holds promise, especially within a 2-meter range, showing an average error of 1.2% on the x-axis and 0.011% on the y-axis during straight-line motion. The system's repeatability is reliable, evident in its ability to return to the home position with a minimal average error of 4.33 cm on the x-axis and 0.5 cm on the y-axis, as demonstrated in the repeatability table.

Furthermore, the implementation testing within a designated arena showcased the robot's capability to navigate through various obstacles like cracked paths, inclined surfaces, rocks, and marbles. The robot successfully navigated these challenges and reached its intended destinations during the initial trials. In a subsequent iteration, however, the robot encountered difficulty reaching target 3 due to synchronization issues attributed to hardware limitations. Notably, the Raspberry Pi 4 CPU reached usage rates of up to 97%, affecting program synchronization and resulting in navigational challenges.

These results collectively underscore the successful integration of the navigation system, highlighting its potential for various applications. The achieved accuracy, repeatability, and obstacle navigation capability substantiate its effectiveness in real-world scenarios.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Politeknik Manufaktur Bandung for providing sponsorship and financial support for the development of the robot used in this research. Without their generous support, this project would not have been possible. Acknowledging the invaluable contributions of sponsors and financial support is vital in research papers, as it highlights the institutions' commitment to fostering advancements in technology and promoting scientific endeavors. The authors acknowledge the significant role of Politeknik Manufaktur Bandung in making this research a reality and extend their appreciation for the resources and opportunities provided. The support received from Politeknik Manufaktur Bandung not only enabled the successful completion of this project but also contributed to the authors' professional growth and development. The guidance and assistance provided by the institution's faculty and staff were instrumental in the research's execution, and the authors are grateful for the invaluable mentorship they received.

Furthermore, the authors would like to acknowledge the collaboration and coordination with the institution's various departments and facilities, which played a pivotal role in the successful implementation of the robot and experimentation process. The authors also extend their appreciation to all the individuals who directly or indirectly supported the research project. Their contributions, whether technical, logistical, or intellectual, have greatly enriched the outcome of this work. In conclusion, the authors are immensely grateful to Politeknik Manufaktur Bandung for their unwavering support, which has been instrumental in the successful execution of this research. Their sponsorship and financial backing have made a profound impact, and the authors are proud to acknowledge their invaluable contribution to the advancement of technology and robotics research.




REFERENCES

- [1] Z. Chen, J. Li, S. Wang, J. Wang, and L. Ma, "Flexible gait transition for six wheel-legged robot with unstructured terrains," *Robotics and Autonomous Systems*, vol. 150, Apr. 2022, doi: 10.1016/j.robot.2021.103989.
- [2] Z. Chen, J. Liu, and F. Gao, "Real-time gait planning method for six-legged robots to optimize the performances of terrain adaptability and walking speed," *Mechanism and Machine Theory*, vol. 168, Feb. 2022, doi: 10.1016/j.mechmachtheory.2021.104545.
- [3] J. Chen, Z. Liang, Y. Zhu, and J. Zhao, "Improving kinematic flexibility and walking performance of a six-legged robot by rationally designing leg morphology," *Journal of Bionic Engineering*, vol. 16, no. 4, pp. 608–620, Jul. 2019, doi: 10.1007/s42235-019-0049-9.
- [4] B. Qin, Y. Gao, and Y. Bai, "Sim-to-real: six-legged robot control with deep reinforcement learning and curriculum learning," in *2019 4th International Conference on Robotics and Automation Engineering*, Nov. 2019, pp. 1–5, doi: 10.1109/ICRAE48301.2019.9043822.
- [5] M. Schilling, K. Konen, and T. Korthals, "Modular deep reinforcement learning for emergent locomotion on a six-legged robot," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, Nov. 2020, pp. 946–953, doi: 10.1109/BioRob49111.2020.9224332.
- [6] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, Mar. 2019, doi: 10.1177/1729881419839596.
- [7] A. Nguyen, N. Nguyen, K. Tran, E. Tjiputra, and Q. D. Tran, "Autonomous navigation in complex environments with deep





- multimodal fusion network,” in *IEEE International Conference on Intelligent Robots and Systems*, Oct. 2020, pp. 5824–5830, doi: 10.1109/IROS45743.2020.9341494.
- [8] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, “Autonomous visual navigation for mobile robots: A systematic literature review,” *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–34, Feb. 2020, doi: 10.1145/3368961.
- [9] S. A. S. Mohamed, M. H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, “A survey on odometry for autonomous navigation systems,” *IEEE Access*, vol. 7, pp. 97466–97486, 2019, doi: 10.1109/ACCESS.2019.2929133.
- [10] T. H. Siagian, P. Purhadi, S. Suhartono, and H. Ritonga, “Social vulnerability to natural hazards in Indonesia: driving factors and policy implications,” *Natural Hazards*, vol. 70, no. 2, pp. 1603–1617, Oct. 2014, doi: 10.1007/s11069-013-0888-3.
- [11] A. Boucher *et al.*, “The AROUND project: adapting robotic disaster response to developing countries,” Nov. 2009, doi: 10.1109/SSRR.2009.5424156.
- [12] T. Yoshiike *et al.*, “The experimental humanoid robot E2-DR: design for inspection and disaster response in industrial environments,” *IEEE Robotics and Automation Magazine*, vol. 26, no. 4, pp. 46–58, Dec. 2019, doi: 10.1109/MRA.2019.2941241.
- [13] S. Park, Y. Oh, and D. Hong, “Disaster response and recovery from the perspective of robotics,” *International Journal of Precision Engineering and Manufacturing*, vol. 18, no. 10, pp. 1475–1482, Oct. 2017, doi: 10.1007/s12541-017-0175-4.
- [14] R.; Siegwart *et al.*, “Legged and flying robots for disaster response conference paper legged and flying robots for disaster response,” 2015, doi: 10.3929/ethz-a-010643912.
- [15] F. Delcomyn and M. E. Nelson, “Architectures for a biomimetic hexapod robot,” *Robotics and Autonomous Systems*, vol. 30, no. 1, pp. 5–15, Jan. 2000, doi: 10.1016/S0921-8890(99)00062-7.
- [16] U. Saranli, M. Buehler, and D. E. Koditschek, “Design, modeling and preliminary control of a compliant hexapod robot,” in *Proceedings-IEEE International Conference on Robotics and Automation*, 2000, vol. 3, pp. 2589–2596, doi: 10.1109/ROBOT.2000.846418.
- [17] U. Saranli, M. Buehler, and D. E. Koditschek, “RHex: a simple and highly mobile hexapod robot,” *International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, Jul. 2001, doi: 10.1177/02783640122067570.
- [18] J. Coelho, F. Ribeiro, B. Dias, G. Lopes, and P. Flores, “Trends in the control of hexapod robots: a survey,” *Robotics*, vol. 10, no. 3, Aug. 2021, doi: 10.3390/robotics10030100.
- [19] W. Chen, G. Ren, J. Zhang, and J. Wang, “Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 67, no. 3–4, pp. 255–270, Mar. 2012, doi: 10.1007/s10846-012-9661-1.
- [20] M. A. Lewis, A. H. Fagg, and G. A. Bekey, “Genetic algorithms for gait synthesis in a hexapod robot,” in *World Scientific Series in Robotics and Intelligent Systems*, WORLD SCIENTIFIC, 1994, pp. 317–331.
- [21] D. Li, Y. Gao, W. Wei, and X. Liu, “Terrain adaptation of hexapod robot based on ground detection and sliding mode control,” in *ACM International Conference Proceeding Series*, Jun. 2022, pp. 59–65, doi: 10.1145/3556267.3556271.
- [22] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in *9th IEEE International Symposium on Safety, Security, and Rescue Robotics*, Nov. 2011, pp. 155–160, doi: 10.1109/SSRR.2011.6106777.
- [23] X. Zhang, J. Lai, D. Xu, H. Li, and M. Fu, “2D LiDAR-based SLAM and path planning for indoor rescue using mobile robots,” *Journal of Advanced Transportation*, vol. 2020, pp. 1–14, Nov. 2020, doi: 10.1155/2020/8867937.
- [24] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, “Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, May 2020, doi: 10.1177/1729881420929498.
- [25] O. C. Barawid, A. Mizushima, K. Ishii, and N. Noguchi, “Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application,” *Biosystems Engineering*, vol. 96, no. 2, pp. 139–149, Feb. 2007, doi: 10.1016/j.biosystemseng.2006.10.012.
- [26] J. Li, H. Qin, J. Wang, and J. Li, “OpenStreetMap-based autonomous navigation for the four wheel-legged robot via 3D-LiDAR and CCD camera,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2708–2717, Mar. 2022, doi: 10.1109/TIE.2021.3070508.
- [27] A. A. Zhilenkov, S. G. Chernyi, S. S. Sokolov, and A. P. Nyrkov, “Intelligent autonomous navigation system for UAV in randomly changing environmental conditions,” *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 6619–6625, May 2020, doi: 10.3233/JIFS-179741.
- [28] A. N. A. Rafai, N. Adzhar, and N. I. Jaini, “A review on path planning and obstacle avoidance algorithms for autonomous mobile robots,” *Journal of Robotics*, vol. 2022, pp. 1–14, Dec. 2022, doi: 10.1155/2022/2538220.
- [29] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2D LIDAR SLAM,” in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2016, pp. 1271–1278, doi: 10.1109/ICRA.2016.7487258.
- [30] X. Yang, “Slam and navigation of indoor robot based on ROS and LiDAR,” *Journal of Physics: Conference Series*, vol. 1748, no. 2, Jan. 2021, doi: 10.1088/1742-6596/1748/2/022038.
- [31] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in *9th IEEE International Symposium on Safety, Security, and Rescue Robotics*, Nov. 2011, pp. 155–160, doi: 10.1109/SSRR.2011.6106777.

BIOGRAPHIES OF AUTHORS







Aris Budiarto    is an esteemed lecturer at Politeknik Manufaktur Bandung, specializing in Automation and Mechatronics Engineering. With a teaching tenure since 1992, Aris imparts knowledge in subjects like digital electronics, microprocessor and interface, maintenance & repair, industrial electronics, and industrial automation. Holding degrees from institutions including Bandung Institute of Technology, he has a Diplom Degree in Electronics Engineering, a bachelor's in Electronics Engineering (Computer Engineering Science), a master's in Electronics Engineering (Computer and Control Eng. Science), and a doctorate in Higher Education Management. Aris is an active member of academic societies like ISAS and SME. He can be contacted at email: aris@ae.polman-bandung.ac.id.



Sarosa Castrena Abadi     is a devoted lecturer in the Department of Automation and Mechatronics Engineering at Politeknik Manufaktur Bandung, Indonesia, holds a bachelor's degree in electronics and a master's in electrical and electronic engineering. His enthusiasm extends to various domains, including the internet of things, SCADA systems, embedded systems, networking, and computer systems. With a solid educational foundation and a diverse range of interests, Sarosa contributes to education and research within the field of Automation and Mechatronics Engineering. He can be contacted at email: sarosa@ae.polman-bandung.ac.id.



Naufaldo     is a fresh graduate from Politeknik Manufaktur Bandung, Indonesia, has a fervent passion for Robotics, internet of things (IoT), and programmable logic controllers (PLC). With hands-on experience in industrial robotics, including AGV and robotic arm projects, he has gained practical insights into the field. Naufaldo's versatility extends to PLC projects, reflecting his commitment to industrial automation. His eagerness to learn and apply his knowledge positions him as a promising contributor to the dynamic realms of robotics, IoT, and PLCs. He can be contacted at email: Naufaldo1301@gmail.com.