International Journal of Intelligent Robotics and Applications
https://doi.org/10.1007/s41315-024-00350-1

REGULAR PAPER

ROS-based multi-sensor integrated localization system for cost-effective and accurate indoor navigation system

Achmad Syahrul Irwansyah[1] · Budi Heryadi[1] · Dyah Kusuma Dewi[2] · Roni Permana Saputra[2] · Zainal Abidin[1]

Received: 6 March 2024 / Accepted: 27 May 2024

Abstract

Accurate localization is essential for enabling intelligent autonomous navigation in indoor environments. While global navigation satellite systems (GNSS) provide efficient outdoor solutions, applications in indoor environments require alternative approaches to determine the vehicle's global position. This study investigates a ROS-based multi-sensor integrated localization system utilizing wheel odometry, inertial measurement unit (IMU), and 2D light detection and ranging (LiDAR) based simultaneous localization and mapping (SLAM) for cost-effective and accurate indoor autonomous vehicle (AV) navigation. The paper analyzes the limitations of wheel odometry and IMU, highlighting their susceptibility to errors. To address these limitations, the proposed system leverages LiDAR SLAM for real-time map generation and pose correction. The Karto SLAM package from robot operating system (ROS) is chosen due to its superior performance according to the literature. Results indicate that the integration of these technologies reduces localization errors significantly, with the system achieving a high degree of accuracy in pose estimation under various test conditions. The experimental validation shows that the proposed system maintains consistent performance, proving its potential for widespread application in environments where GNSS is unavailable.

Keywords Simultaneous localization and mapping (SLAM) · Autonomous vehicle · Sensor fusion · Robot operating system · Autonomous navigation · Mobile robot · Unmanned vehicle · Karto SLAM

1 Introduction

Recently, the demand for developing smart and sustainable transportation systems such as autonomous vehicles has significantly increased. In modern manufacturing plants, autonomous vehicle (AV) is used in the production line which leads to a high level of productivity and adapts to flexible manufacturing (Sabattini et al. 2018; Quan and Chen 2019). In Sabattini et al. (2018), Stimming, et al. (2015), LiDAR is used to localize the vehicle with respect to the environment map, while computer-vision-based obstacle detection is used to classify the detected obstacles and make high-level decisions. This makes the autonomous vehicle applicable to mixed environments shared with human operators. At surface mines, autonomous vehicles can be used to eliminate the human factor problem that results in accidents because of intense and monotonous work (Voronov et al. 2020). According to the study reported in Voronov et al. (2020), implementing autonomous operation at the surface mines, could increase the productivity up to 13% as well as reduce the operating costs up to 15%.

One of the key components of the intelligent autonomous navigation in AV is localization (Quan and Chen 2019). Global Navigation Satellite System (GNSS), such as GPS, is widely used and has been proven effective in providing accurate vehicle localization in outdoor environments. This localization technique relies on signals from satellites to determine the position of receivers on Earth. These signals, however, are significantly weakened when passing through solid materials, making GNSS unreliable in indoor environments where the signals are often obstructed by roofs, walls, and other structures. This obstruction leads to poor signal quality or complete signal loss, rendering GNSS ineffective for precise localization within indoor settings. Furthermore, the multi-path effect, where signals bounce off

✉ Achmad Syahrul Irwansyah
ach.syahrul99@gmail.com

[1] Mechanical Engineering, Bandung Institute of Technology, Bandung, Indonesia

[2] Research Center for Smart Mechatronics, National Research and Innovation Agency, Bandung, Indonesia

Published online: 05 June 2024

surfaces before reaching the receiver, can cause additional errors and inaccuracies in position estimation. Thus, precise indoor localization or localization in environments where GNSS signals are compromised or unavailable remains a topic of active research and discussion (Nimura et al. 2023).

One of the widely used technique for localization in GNSS-absence environments is wheel odometry. This technique provides a simple and low-cost localization solution (He et al. 2023). However, this technique suffers from several limitations that affect its accuracy, including wheel slippage, wear, and tire pressure changes, and cumulative errors (He et al. 2023).

To accomplish global positioning estimation in indoor environments, simultaneous localization and mapping (SLAM) can be implemented using point cloud registration techniques with range-finder sensors such as LiDAR. However, point cloud registration can face challenges in unstructured environments that lack distinct geometric features, such as empty warehouses, long empty corridors, large open areas without significant furnishings or features, and environments with repetitive patterns that confuse sensor readings (Dai, et al. 2023). To address these challenges, in Junior et al. (2022), a data fusion framework incorporating LiDAR, wheel odometry, and inertial sensors has been proposed to enhance SLAM capabilities, particularly in homogeneous and unstructured environments where traditional methods may falter. This approach aims to leverage the strengths of each sensor type to improve the overall robustness and accuracy of the localization and mapping process.

This paper introduces a ROS-based multi-sensor integrated localization framework, leveraging wheel odometry, IMU, and 2D Lidar SLAM. An in-depth investigation into the error characteristics of wheel odometry and IMU in estimating robot poses (position and orientation) is conducted. The Karto SLAM package from ROS is used in this study based on its superior performance over other SLAM approaches as demonstrated in studies conducted in Zhao et al. (2022) and Bhargava et al. (2021). Pose estimation is first aided by wheel encoder and IMU data measurements, and then the SLAM algorithm uses LiDAR sensor observations to fine-tune the initial estimate. To evaluate the performance of the proposed localization system, a cost-effective prototype has been developed utilizing Raspberry Pi and Arduino Mega as its primary controllers. This implementation offers a cost-effective solution for achieving accurate indoor localization, thus providing an accessible alternative for wide-rage applications.

The main contributions of this study includes:

1. Implementation of ROS-based localization system combining wheel odometry, IMU, and 2D Lidar SLAM to achieve accurate and cost-effective indoor localization for autonomous vehicles.
2. Comprehensive analysis and evaluation of each sensor error characteristics (i.e., IMU and wheel odometry), and performance evaluation of Karto SLAM from ROS as the optimal SLAM algorithm for indoor localization technique.
3. Practical implementation of the proposed indoor localization technique using Raspberry Pi and Arduino Mega as controllers, showcasing the feasibility of achieving accurate indoor localization with affordable hardware.

The remaining sections of the paper are structured as follows: Sect. 2 reviews existing research on indoor localization, while Sect. 3 outlines the methodology employed in the study. Section 4 details the implementation process of the proposed system, and Sect. 5 presents the experimental setup. The results of the experiments are further discussed in Sect. 6, and finally, Sect. 7 concludes the paper by summarizing the key findings and contributions.

2 Related work

In GNSS-absent environments, such as in indoor navigation scenarios, wheel odometry is widely recognized for its simplicity and cost-effectiveness in localization tasks (He et al. 2023; Fazekas et al. 2021a, 2021b; Aqel et al. 2016; Yan et al. 2022). Despite its ubiquity, accuracy issues arise due to mechanical wear wheel slippage, tire pressure variations, and cumulative errors, necessitating innovative solutions (He et al. 2023). To address these challenges, He et al. (2023) and Brossard and Bonnabel (2019) propose deep learning-based wheel odometry error prediction algorithms, while Wu et al. (2017) and Quan et al. (2019) leverage a combination of wheel odometry, IMU, and visual observations from cameras to enhance localization accuracy.

Despite its benefits, wheel odometry alone fails to determine global position during indoor localization, prompting the exploration of simultaneous localization and mapping (SLAM) techniques (Liao et al. 2019; Nam and Gon-Woo 2021; Huang 2021; Khan, et al. 2021; Xu et al. 2022; Zou et al. 2021). SLAM in indoor environments often employs point cloud registration using range finder sensors like LiDAR (Liao et al. 2019). However, point cloud registration can falter in unstructured environments lacking geometric features, thus compromising LiDAR SLAM accuracy (Dai et al. 2023). To improve SLAM capabilities, (Junior et al. 2022) proposes a data fusion framework incorporating LiDAR, wheel odometry, and inertial sensors, particularly effective in homogeneous unstructured environments.

The development of multi-sensor fusion SLAM, particularly the incorporation of LiDAR data, plays a pivotal role in the advancement of stable and robust localization systems (Zhang et al. 2019; Qin and Liu 2021; Li 2020; Yang et al. 2022). The integration of multiple sensors enhances the robustness and accuracy of localization systems by leveraging the strengths of each sensor type to overcome their individual limitations. Research in this field is extensive and focuses on improving the accuracy, reliability, and versatility of SLAM systems under varied environmental conditions.

Several studies have highlighted the efficacy of combining GNSS, Inertial Measurement IMU, odometry, and LiDAR in a unified SLAM approach. For instance, Chang et al. (2020) propose an integrated navigation system that merges these technologies to ensure high data rate transmissions, enhanced accuracy, and reliable performance across diverse weather conditions. This approach is particularly advantageous in complex environments where single sensor systems may fail to provide reliable data due to physical or atmospheric interferences.

Further studies in SLAM technology include the application of intensity feature extraction from LiDAR measurements, which has been shown to significantly enhance the detection and distinction of environmental features. Studies by Dai et al. (2023) and Zhang et al. (2023) demonstrate how intensity-based feature extraction methods can be used to improve the environmental modelling capabilities of LiDAR systems, leading to more accurate mapping and localization in previously challenging scenarios.

The use of convolutional neural networks (CNNs) for odometry estimation represents another significant advancement in the field. Velas et al. (2018) have explored how CNNs can be trained to interpret complex sensor data and predict vehicle odometry with high precision. This method has proven particularly effective in environments where traditional odometry techniques struggle due to variable terrain or other complicating factors.

Additionally, the optimization of particle filters utilizing surface normal constraints has been proposed as a method to refine the pose estimation process. Andradi et al. (2023) illustrate how applying these constraints can minimize the common errors associated with particle filter algorithms in SLAM, particularly in cluttered or dynamically changing environments. This method enhances the accuracy of the localization process by ensuring that the particle filter more effectively represents the true state of the system.

Another recent notable development in SLAM technology is the advent of semantic SLAM, which integrates deep learning techniques to interpret and interact with the environment in a context-aware manner. Pu et al. (2023) explore how semantic cues from the environment can be leveraged to significantly improve the accuracy and efficiency of SLAM algorithms. By recognizing and understanding environmental features, semantic SLAM enables more intelligent decision-making in autonomous navigation, making it highly effective in complex and dynamically changing scenarios.

Additionally, the evolution of graph-based SLAM techniques represents a crucial enhancement in the field. Hoshi et al. (2022) discuss the implementation of robust graph optimization algorithms that are capable of solving large-scale and complex SLAM problems with greater efficiency. These algorithms form the backbone of modern SLAM systems, providing a structured approach to managing vast amounts of data and ensuring high precision in real-time localization and mapping tasks.

Moreover, the challenge of deploying SLAM in dynamic environments where objects and obstacles frequently change has spurred innovative solutions. Xing et al. (2022) investigate the use of dynamic SLAM algorithms that can adapt to changing conditions in real-time, which is crucial for navigating busy urban areas or complex industrial settings. These dynamic algorithms are designed to update and modify the SLAM map continuously as new information becomes available, thus maintaining the system's reliability and accuracy even in highly volatile environments.

In recent years, the use of open-source platforms like ROS (Robot Operating System) has become increasingly prevalent in the development of SLAM algorithms. This trend is represented in numerous studies that explore the diverse range of SLAM algorithms available within the ROS ecosystem, examining their performance metrics and practical applicability. For example, research documented in Zhao et al. (2022), Bhargava et al. (2021), Le et al. (2018) and Olalekan et al. (2021) investigates the capabilities and efficiencies of various SLAM technologies, providing a comparative analysis of their strengths and weaknesses.

A notable study by Zhao et al. (2022) critically evaluates three major SLAM algorithms: GMapping, Hector SLAM, and Karto SLAM. This evaluation is conducted using benchmark datasets designed to test the algorithms under a variety of environmental conditions and operational scenarios. The results from these evaluations are crucial for understanding how different algorithms perform in real-world settings. Furthermore, studies Bhargava et al. (2021) and Olalekan et al. (2021) utilize the structural similarity index measure (SSIM) as a metric to assess the similarity of maps produced by different SLAM processes and to evaluate their computational efficiency. This approach offers insights into the quality of the environmental mapping and the resource demands of each algorithm.

Among the algorithms assessed, Karto SLAM stands out, demonstrating superior performance in terms of mapping accuracy and CPU efficiency, as highlighted in Zhao et al. (2022) and Bhargava et al. (2021). This consistency makes Karto SLAM particularly attractive for applications requiring high precision and computational efficiency, such as in robotic navigation and automated systems in complex environments.

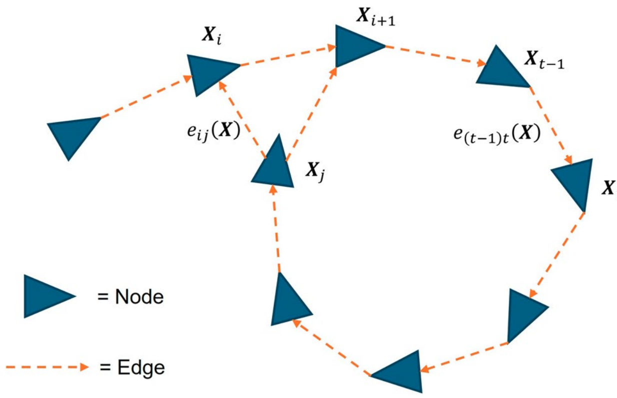


Fig. 1 Graph-based SLAM (Adapted from Grisetti et al. (2010))

3 Methodology

This section outlines the key components of the approach used in this study as well as the comprehensive evaluation performed to assess the proposed multi-sensor integrated localization system.

In this study we utilise the Karto SLAM algorithm, which is one of the commonly used SLAM techniques that relies on graph optimisation (Zhao et al. 2022). A graph-based approach creates a graph consisting of nodes and edges, as seen in Fig. 1. Each node in the graph represents the vehicle's pose X (Grisetti et al. 2010). In a two-dimensional (2D) scenario, the vehicle's state X is defined by its x and y position coordinates, as well as its orientation θ as shown in Eq. (1).

$$\boldsymbol{X}_i^T = \left[\boldsymbol{t}_i^T \ \theta_i\right] = \left[x_i \ y_i \ \theta_i\right] \tag{1}$$

The nearby nodes in the graph are connected by edges that model spatial constraints between vehicle poses resulting from measurements (Grisetti et al. 2010).

The t-th node, $\boldsymbol{X}_t$, will be connected to the $(t-1)$-th node, $\boldsymbol{X}_{t-1}$, by edges, $e_{(t-1)t}$, that contain spatial constraints from odometry measurements, while the other nodes (e.g. $\boldsymbol{X}_i$ to $\boldsymbol{X}_j$) will be connected by edges that represent spatial constraints from the observation of the same part of the environment. The error $e_{ij}(\boldsymbol{X})$ represent the error between the estimated transformation ($\Delta\boldsymbol{X}_{ij}$) with the observed transformation $\boldsymbol{z_{ij}}$ that defines as follow (Grisetti et al. 2010):

$$e_{ij}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{R}_{ij,z}\left(\boldsymbol{R}_i\left(\boldsymbol{t}_j^T - \boldsymbol{t}_i^T\right) - \boldsymbol{t}_{ij,z}^T\right) \\ \theta_j - \theta_i - \theta_{ij,z} \end{bmatrix} \tag{2}$$

where $\boldsymbol{R}_{ij,z}$ and $\boldsymbol{R}_i$ are the 2×2 rotation matrices $\theta_{ij,z}$ and θ_i with the following structure:

$$\boldsymbol{R}_{ij,z} = \begin{bmatrix} \cos\left(\theta_{ij,z}\right) & -\sin\left(\theta_{ij,z}\right) \\ \sin\left(\theta_{ij,z}\right) & \cos\left(\theta_{ij,z}\right) \end{bmatrix}. \tag{3}$$

Then the total error is computed as (Han et al. 2020):

$$F(\boldsymbol{X}) = \sum_{ij} e_{ij}^T \Omega_{ij} e_{ij} \tag{4}$$

where the information matrix Ω_{ij} is the inverse of covariance matrix between node $\boldsymbol{X}_i$ and node $\boldsymbol{X}_j$ (Han et al. 2020). Then, the optimized nodes of the graph will follow this equation:

$$\boldsymbol{X}_{opt} = \operatorname{argmin} F(\boldsymbol{X}) \tag{5}$$

that is the value of the corrected poses $\boldsymbol{X}_{opt}$ which will minimize the value of the function $F(\boldsymbol{X})$.

The Karto SLAM, developed by SRI International's Karto Robotic, utilizes a highly optimized and non-iterative Cholesky matrix decomposition for solving sparse linear systems, complemented by SPA (sparse pose adjustments) for scan matching and loopback detection (Zhao et al. 2022; Santos et al. 2013). The algorithm's framework is illustrated in Fig. 2. Karto SLAM uses odometry data to assists with the lidar scan matching process by giving an initial pose. As the lidar acquires new scan data, the algorithm continuously

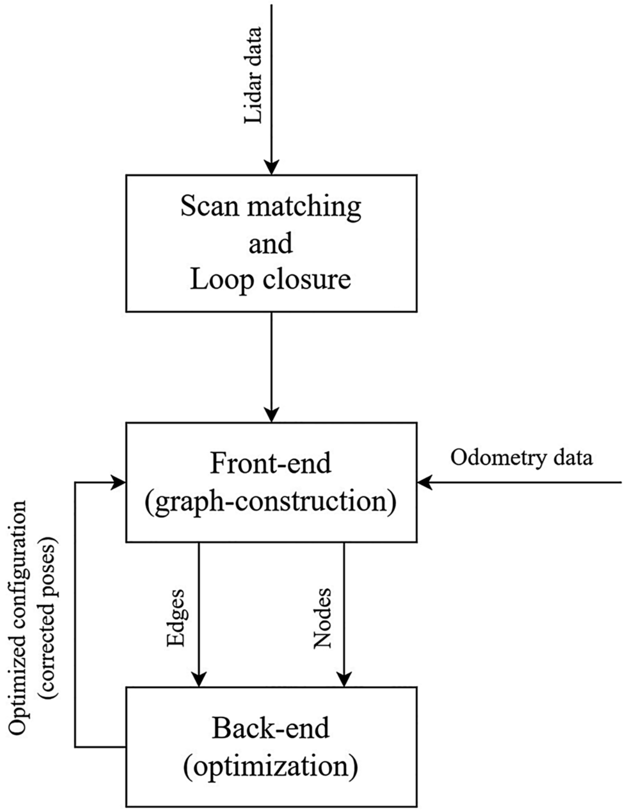

Lidar data
Scan matching and Loop closure
Front-end (graph-construction)
Odometry data
Edges
Nodes
Back-end (optimization)
Optimized configuration (corrected poses)

Fig. 2 Graph-based SLAM framework used in Karto SLAM (adapted from Han et al. (2020))

updates the pose estimation by constructing a map and aligning the new scan data with the current map. Karto SLAM optimises and corrects poses using nodes and edges at the back end after detecting a loop closure (Konolige, et al. 2010).

This work evaluates the implemented Karto SLAM algorithm to assess its impact on improving the accuracy of the proposed localization. Furthermore, mapping tests are conducted to evaluate the consistency of map construction and the precision of object measurements on the map. The accuracy of the produced object dimensions is assessed by comparing them to the actual object dimensions. The structured similarity indexing method (SSIM) value is calculated between multiple constructed maps to assess the consistency of the mapping process. The SSIM value is calculated using the equation (Le et al. 2018):

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{6}$$

where μ_x and μ_y are the mean of intensities value of the image x and the image y. On the other hand, σ_x and σ_y are the standard deviation of intensities value of the image x and the image y, while σ_{xy} the intensity correlation between the image x and the image y.

To evaluate the performance of the proposed localization system, a laboratory scaled vehicle prototype has been developed. This vehicle uses a differential-drive mechanism with dual motors operating on its driving wheels. Each motor is equipped with an incremental encoder sensor to estimate wheel odometry accurately. The vehicle prototype further incorporates an IMU MPU-6500 as the inertial sensor and RPLidar A1M8 as the laser range finder sensor. The vehicle is powered by a Raspberry Pi Model 3B running the robot operating system (ROS) on Ubuntu operating system.

The comprehensive evaluation of the ROS-based multi-sensor integrated localization with wheel odometry, IMU, and LiDAR based SLAM performance involves a series of experiments, covering localization and mapping assessments. The main objective of the localization tests is to evaluate the performance of wheel odometry against IMU localization, allowing for a better understanding of the error characteristics associated with these two sensors. Subsequently, to efficiently combine the wheel odometry data, a sensor fusion technique will be identified. Then, using the ROS Karto SLAM module, this revised data will be sent into the LiDAR-based SLAM algorithm.

Throughout these experiments on the designed prototype, we aim to evaluate the performance of each individual sensor as well as develop an optimal sensor fusion method for improved the SLAM capability. With a comprehensive understanding of the system's capabilities, we firmly believe that our proposed localization system holds significant potential for application across a wide range of scenarios in diverse environments.

4 System implementation

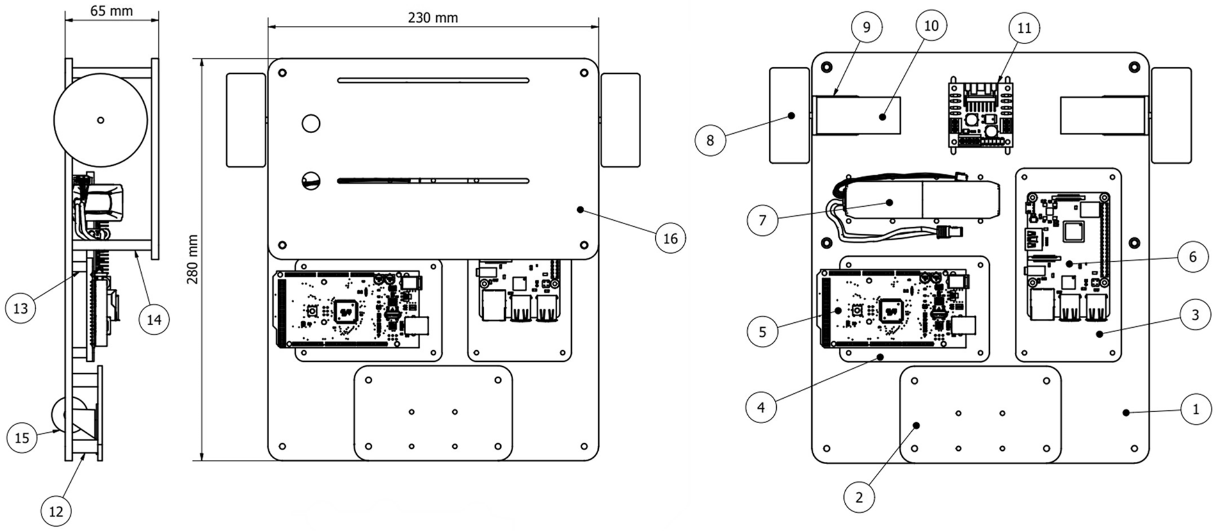
4.1 Hardware architecture

The prototype consists of a 2-stage platform constructed from 5 mm acrylic plates, with 55 mm aluminum spacers between them. Figure 3 illustrates the layout of the prototype components, with dimensions of 230 × 280 mm, designed for easy movement in indoor settings. Most components are placed on the first stage to keep the centre of gravity lower. A lower center of gravity helps stabilize the platform by reducing the likelihood of tipping over. This is particularly important in environments where the robot may encounter uneven surfaces or perform maneuvers such as sharp turns or sudden stops. By placing most components on the first stage, closer to the ground, the prototype's mass is distributed in a way that enhances its ability to remain upright and stable under various operational conditions.

The LiDAR sensor is mounted on the second stage to provide a 360-degree field of view for efficient scanning. The prototype is powered by a differential drive system using two DC motors and a caster wheel for stability, with the IMU MPU-6500 module mounted on the Arduino Mega shield. The complete assembly of the system is depicted in Fig. 4.

Figure 5 illustrates the hardware architecture of the prototype. The prototype is powered by a LiPo battery 3S 2200mAh and is driven by two JGA25-370-21.3 K (12 V) series DC motors, each having an incremental encoder. The battery has a nominal voltage of 11.1 V and can be charged up to 12.6 V. It regulates the voltage supplied to the motors via the L298N motor driver, which is adjusted by pulse width modulation (PWM) values from the microcontroller. Additionally, the Raspberry Pi is powered by the same battery, which has its voltage step down to 5 V using a UBEC 5 V 3A.

The developed ROS programs operate on two devices: a personal computer (PC) and a Raspberry Pi Model 3B with a Quad-Core 1.2 GHz Broadcom BCM2837 64bit CPU, 1 GB RAM, and Ubuntu OS. The ROS programs on the PC run within an Ubuntu 18.04 virtual machine. Figure 6 illustrates the software architecture, revealing the interconnection between ROS nodes and topics. Both machines utilize a shared ROS master integrated into a local network via Wi-Fi connection.



Components:

1. Stage 1 platform
2. Caster wheel housing
3. RaspberryPi platform
4. Microcontroller platform
5. Microcontroller (Arduino Mega)
6. RaspberryPi
7. LiPo battery 12V
8. Driving wheel
9. DC motor mounting
10. DC motor
11. Motor Driver L298N

12, 13, 14. Spacer

15. Caster wheel
16. Stage 2 platform

Fig. 3 The components layout of the prototype

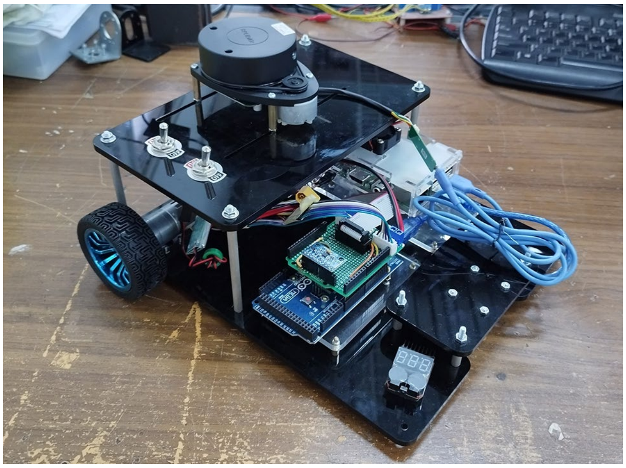


Fig. 4 The outlook of the prototype

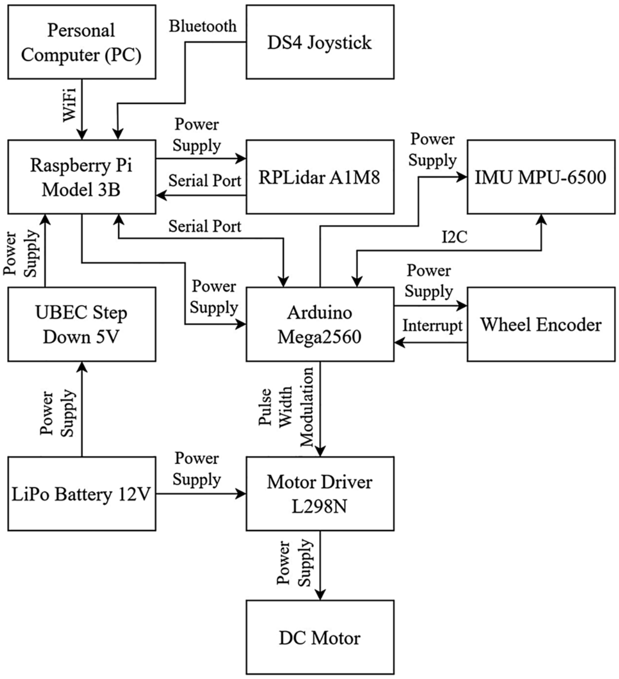


Fig. 5 Hardware architecture of the prototype

4.2 Software architecture

The proposed system in this work comprises two major computational devices, Raspberry Pi and PC, as well as one microcontroller device, as described in the previous section. The Raspberry Pi manages the processing and communication tasks related to the Joystick, LiDAR sensors, and serial connection with the Arduino microcontroller. The computational operation on the PC manages all the computational

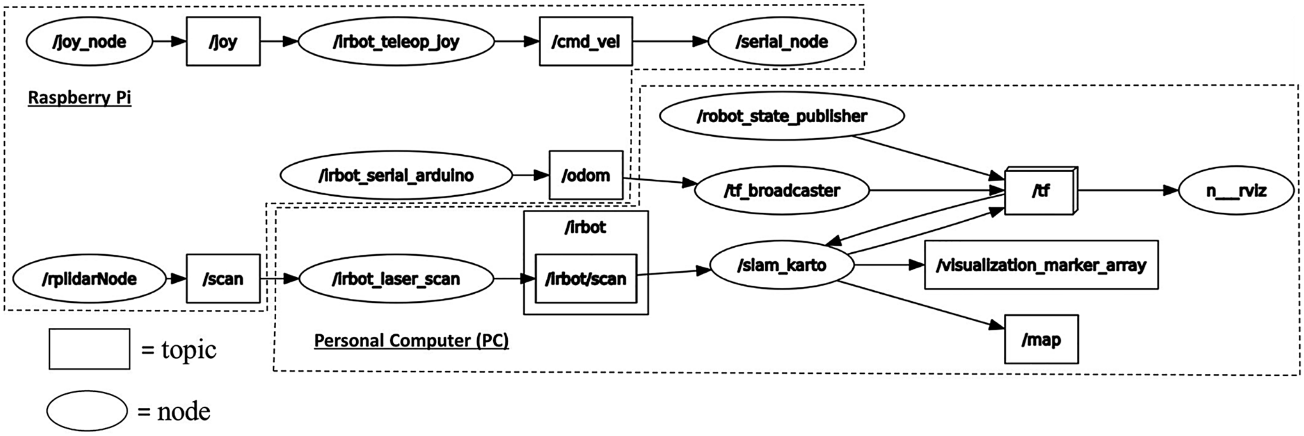


Fig. 6 The software architecture of the software implementation of the proposed ROS-based multi-sensor localization system running on Raspberry Pi and PC

processes related to the Karto SLAM algorithm. The Arduino microcontroller manages the low-level operation of the driving motor and measures the wheel odometry using encoder sensors.

The Raspberry Pi establishes a Bluetooth connection with the Dualshock 4 Joystick in order to control the motion of the vehicle. The *"/joy_node"* and *"/irbot_teleop_joy"* nodes operate on the Raspberry Pi, generate the *"/cmd_vel"* topic as velocity commands for the motors. The RPLidar A1M8 Lidar sensor establishes an interface with the Raspberry Pi by leveraging the seamless data extraction capability of ROS framework. The *"rplidarNode"* node operate on the Raspberry Pi publishes the "/scan" topic, and subsequently the topic is republished by the *"/irbot_laser_scan"* node running on the PC. By performing this topic remapping, the timestamp is synchronised with the PC timestamp. Establishing bidirectional communication, the *"/irbot_serial_arduino"* node publish odometry transformations *"/odom"* from Arduino to the *"/slam_karto"* node, while the *"/serial_node"* publish the *"/cmd_vel"* topics to Arduino. Figure 6 illustrates the communication connections among nodes in the ROS platform utilized for the prototype, providing a graphical representation of the system's complexity and interconnections.

The Karto SLAM algorithm is implemented and run on the Ubuntu Virtual Machine on the PC. The Karto SLAM publish the *"/map"*, *"/visualization_marker_array"*, and *"/tf"* topics as a results of its SLAM algorithm. Figure 7 shows a visual representation of the mapping results generated by the Karto SLAM in Rviz. The *"/visualization_marker_array"* represented in red on the map, corresponds to the locations of vehicles, as illustrated in Fig. 7.

A reliable low-level controller for the driving motors is required to implement the vehicle control command that was published in the *"/cmd_vel"* topic. Controlling the DC motors involves setting the desired wheel velocity (N_{target}) for each motor. The desired input is compared with the actual wheel velocity (N) to calculate the error (e). Prior to the comparison, a low pass filter (LPF) is used to filter the measured wheel velocity and remove high-frequency noise. The error indicates the action control (u) of the proportional integral derivative (PID) controller, as shown in the block diagram in Fig. 8.

The target wheel velocity value is computed based on the vehicle's linear velocity command (V) and angular velocity command (ω), following the kinematic model of the differential drive. The inverse kinematic equation is expressed as:

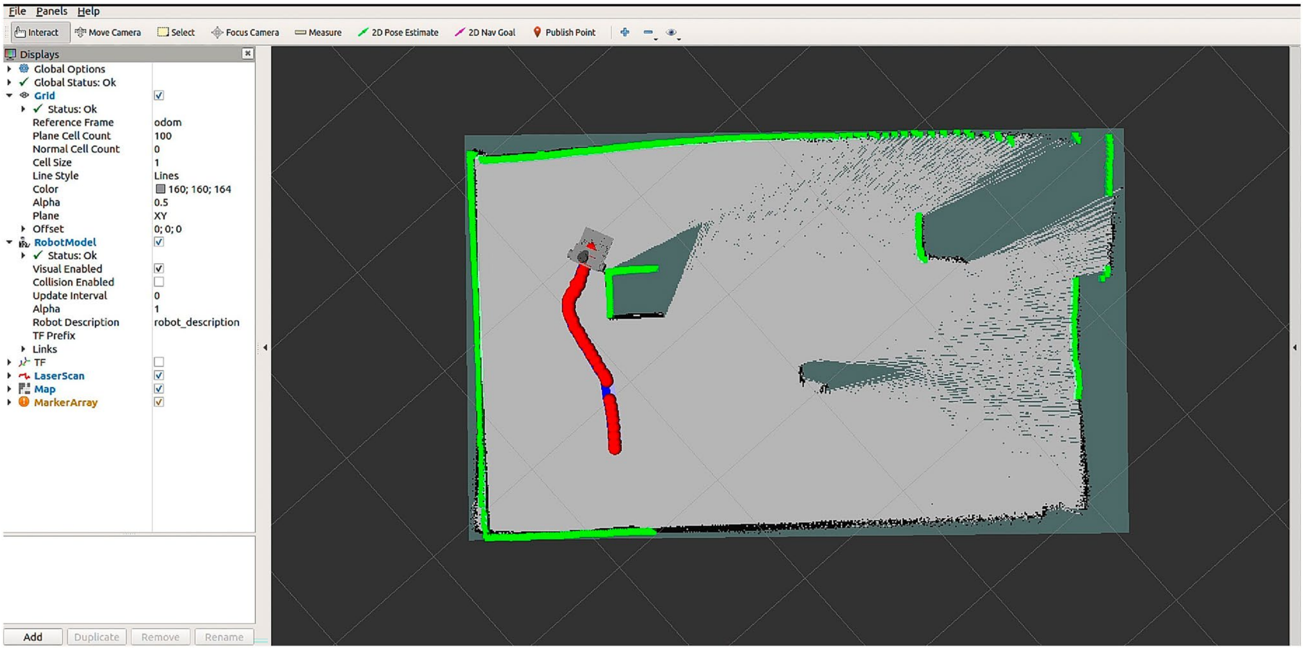
$$N_r = 9{,}55\left(\frac{1}{r}V + \frac{L}{2r}\omega\right) \tag{7}$$

$$N_l = 9{,}55\left(\frac{1}{r}V - \frac{L}{2r}\omega\right) \tag{8}$$

where N_r and N_l are the target values for the right and left wheels, respectively, in revolutions per minute (RPM), while r is the wheel radius in meters, and L is the wheel distance in meters.

On the other hand, the vehicle's pose (position and orientation) in the global coordinate can be known from the forward kinematics of the differential drive. In this context, there are three special cases of vehicle movement to get its pose from the forward kinematics. In the first case, when the vehicle moves along the straight line, the vehicle's pose is computed as follow:

$$\begin{bmatrix}\theta_t\\ x_t\\ y_t\end{bmatrix} = \begin{bmatrix}\theta_{t-1}\\ x_{t-1}\\ y_{t-1}\end{bmatrix} + \begin{bmatrix}0\\ \frac{\Delta\phi_r+\Delta\phi_l}{2}\cos(\theta_t)\\ \frac{\Delta\phi_r+\Delta\phi_l}{2}\sin(\theta_t)\end{bmatrix} \tag{9}$$



Displays

Global Options	
✓ Global Status: Ok	
Grid	☑
✓ Status: Ok	
Reference Frame	odom
Plane Cell Count	100
Normal Cell Count	0
Cell Size	1
Line Style	Lines
Color	▨ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	0; 0; 0
RobotModel	☑
✓ Status: Ok	
Visual Enabled	☑
Collision Enabled	□
Update Interval	0
Alpha	1
Robot Description	robot_description
TF Prefix	
Links	
TF	□
LaserScan	☑
Map	☑
MarkerArray	☑

Fig. 7 Rviz map visualization using Karto SLAM program

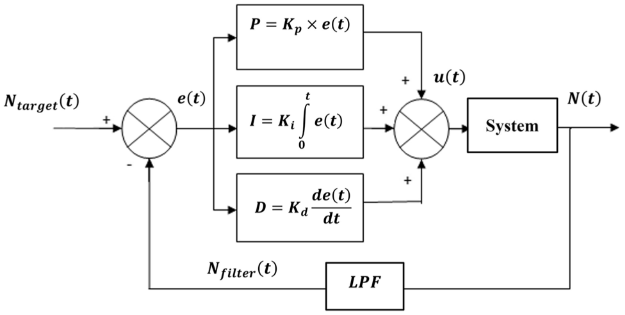


Fig. 8 The block diagram of the motor velocity control using PID controller

where $\Delta\phi_r$ and $\Delta\phi_r$ are the value of change in the right and the left wheel angle respectively during a sampling time interval. In the second case, when the vehicle e is only rotating, the vehicle's pose is computed as follow:

$$\begin{bmatrix} \theta_t \\ x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \theta_{t-1} \\ x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{r}{L}\left(\Delta\phi_r - \Delta\phi_l\right) \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

where r is the wheel radius and L is the wheel distance. The last case, when the vehicle moves in a circular motion with a certain instantaneous center of curvature (ICC), the vehicle's pose is computed as follow:

$$\begin{bmatrix} \theta_t \\ x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \theta_{t-1} \\ x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{r}{L}\left(\Delta\phi_r - \Delta\phi_l\right) \\ -\frac{L}{2}\left(\frac{\Delta\phi_r+\Delta\phi_l}{\Delta\phi_r-\Delta\phi_l}\right)\sin\left(\theta_{t-1}\right) + \frac{L}{2}\left(\frac{\Delta\phi_r+\Delta\phi_l}{\Delta\phi_r-\Delta\phi_l}\right)\sin\left(\theta_t\right) \\ \frac{L}{2}\left(\frac{\Delta\phi_r+\Delta\phi_l}{\Delta\phi_r-\Delta\phi_l}\right)\cos\left(\theta_{t-1}\right) - \frac{L}{2}\left(\frac{\Delta\phi_r+\Delta\phi_l}{\Delta\phi_r-\Delta\phi_l}\right)\cos\left(\theta_t\right) \end{bmatrix} \tag{11}$$

All variables with the subscript t are the current poses of the vehicle, while variables with the subscript $(t-1)$ are the vehicle e pose value at the previous sampling time.

The inertial measurement unit (IMU) is crucial for improving the accuracy of the localization system by providing important information on changes in vehicle angles and accelerations. When combined with accurate wheel encoder measurements and the detailed environmental mapping capabilities of LiDAR sensors, the IMU data forms a robust combination that significantly enhances the overall accuracy and reliability of estimating the vehicle's position and orientation. This integration employs advanced filtering algorithms, such as Kalman filters and their variants or particle filters, which are adept at optimally combining data from various sensors. These filters effectively correct cumulative errors and stabilize the estimation process, substantially increasing the system's navigational reliability without relying on GPS.

An IMU comprises gyroscope and accelerometer sensors that measure the vehicle's angular rates and accelerations, respectively. This setup enables vehicle pose estimation through a mechanization process, as illustrated in Fig. 9. This process involves a complex integration of data from both gyroscopes and accelerometers, a method essential for navigation in GPS-denied environments.

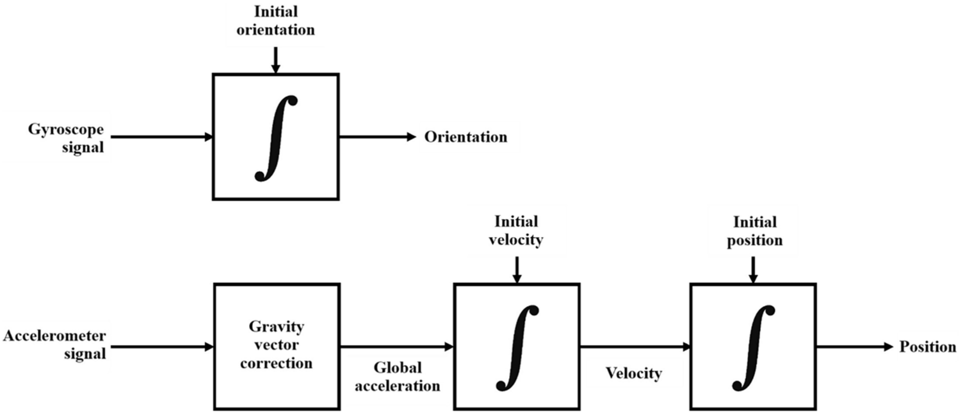


Fig. 9 The diagram block of localization using IMU (redrawn from Woodman (2007))

Gyroscopes measure the angular velocities, providing crucial data about the rate of rotation around the vehicle's axes. This information is vital for continuously updating the vehicle's orientation. Orientation is computed by integrating these angular velocities from a known initial orientation, using techniques such as quaternion algebra or rotation matrices. These methods are preferred for their efficiency and their ability to minimize computational errors over time.

Accelerometers, on the other hand, capture the linear accelerations experienced by the vehicle, which include both motion-induced accelerations and the gravitational pull. To focus solely on the accelerations caused by the vehicle's movement, a gravity vector correction is applied to the raw accelerometer data. This correction is achieved by subtracting the gravitational force (g) from the measured acceleration (a_{raw}), isolating the acceleration due to motion.

$$a = a_{raw} - g \tag{12}$$

Once the gravity is subtracted, the acceleration data must be transformed from the sensor frame to the global frame, accounting for the vehicle's current orientation. This transformation ensures that the accelerations are correctly oriented relative to the Earth, allowing for accurate velocity updates.

The velocity of the vehicle is then determined by integrating this corrected and oriented acceleration over time, starting from an initial known velocity.

$$v(t) = v(t-1) + a\Delta t \tag{13}$$

Finally, the position of the vehicle (p) is updated by integrating the velocity (v). Starting from a known initial position, the system integrates the velocity to calculate the new position over time, providing a trajectory of the vehicle's movement.

$$p(t) = p(t-1) + v(t)\Delta t \tag{14}$$

A more thorough examination of the mathematical intricacies and a comprehensive comprehension of the mechanization equations employed in this procedure can be found in the work of Groves (2015).

The integration of the IMU data and wheel odometry data into the KARTO SLAM algorithm starts with the strategic use of wheel odometry. The linear velocity data collected from wheel odometry is crucial as it provides a continuous input on the vehicle's motion over the ground. This initial layer of data serves as the base for estimating changes in position, offering a fundamental trajectory and speed of the vehicle essential for the early stages of map creation and pose estimation.

In conjunction with odometry, the orientation data from the IMU plays a pivotal role. The IMU complements the odometry by providing additional insights into the vehicle's orientation, which is especially beneficial in environments where the vehicle might experience tilting or other forms of angular motion not fully captured by odometry alone. The integration of IMU data aids in refining the pose estimation, ensuring the vehicle's orientation is accurately represented, particularly in scenarios where odometry data alone might be insufficient.

A critical component of the KARTO SLAM algorithm is laser scan matching, which involves comparing successive laser scans obtained from the LiDAR. This method is essential for detecting and aligning features in the environment, such as walls, obstacles, and other significant landmarks. Laser scan matching significantly enhances pose estimation by providing a spatial context and pinpointing stable landmarks, which help correct any drift or errors that have accumulated from the IMU and odometry data, thus improving both the map's accuracy and the robustness of the localization.

The KARTO SLAM algorithm integrates these diverse data streams through graph optimization. In this method, each sensor input adds nodes and edges to a graph that represents the vehicle's trajectory and the surrounding environment. The optimization process then adjusts the nodes (poses) within the graph to minimize the overall system error. This adjustment results in a highly accurate map and vehicle trajectory by leveraging the strengths of each sensor type and effectively combining them using sophisticated SLAM techniques. This integration strategy ensures that the vehicle's pose is estimated with high precision, which is crucial for applications that require precise navigation and mapping.

5 Experimental setup

5.1 Localization experiments

The localization experiments conducted in this study began with a comprehensive evaluation of the wheel encoder and inertial measurement unit (IMU) sensors, specifically to assess their performance on estimating vehicle position. This evaluation comprised two primary testing scenarios: straight motion and rotational motion. These experiments were aimed to investigate the error characteristics of the sensors in estimating the vehicle's pose, thereby informing the determination of a suitable sensor fusion method.

For straight motion testing (as depicted in Fig. 10a), the vehicle was positioned at the starting line (see Fig. 10b), and then moved along a straight path while maintaining a fixed vehicle orientation of 0 degrees. Data from the sensors were collected after the vehicle covered several meters. Rotational motion testing involved the use of a laser pointer device (Fig. 11a) to project a vertical line onto the screen, aligned with the orientation reference line on the floor (Fig. 11b, c). The vehicle underwent rotation along the z-axis, with data measurements collected for predetermined angles marked on the floor. The sensor characteristics identified from these experiments then would be used for implementation of odometry method for vehicle position and orientation prediction that will be used to aid the implementation of Karto SLAM algorithm.

The next evaluation on the localization experiments is the SLAM localization test. The experiment proceeded with an evaluation of the localization performance using the Karto SLAM program, an integral component of the robot operating system (ROS) package. The experiments were conducted within the area illustrated in Fig. 12. Eight predefined test points were the focus of the experiment, in which the vehicle was initially placed at the coordinate center (0,0). The vehicle subsequently traversed each test location, and its position (x and y) was measured using the SLAM algorithm's estimation. The experiment was then

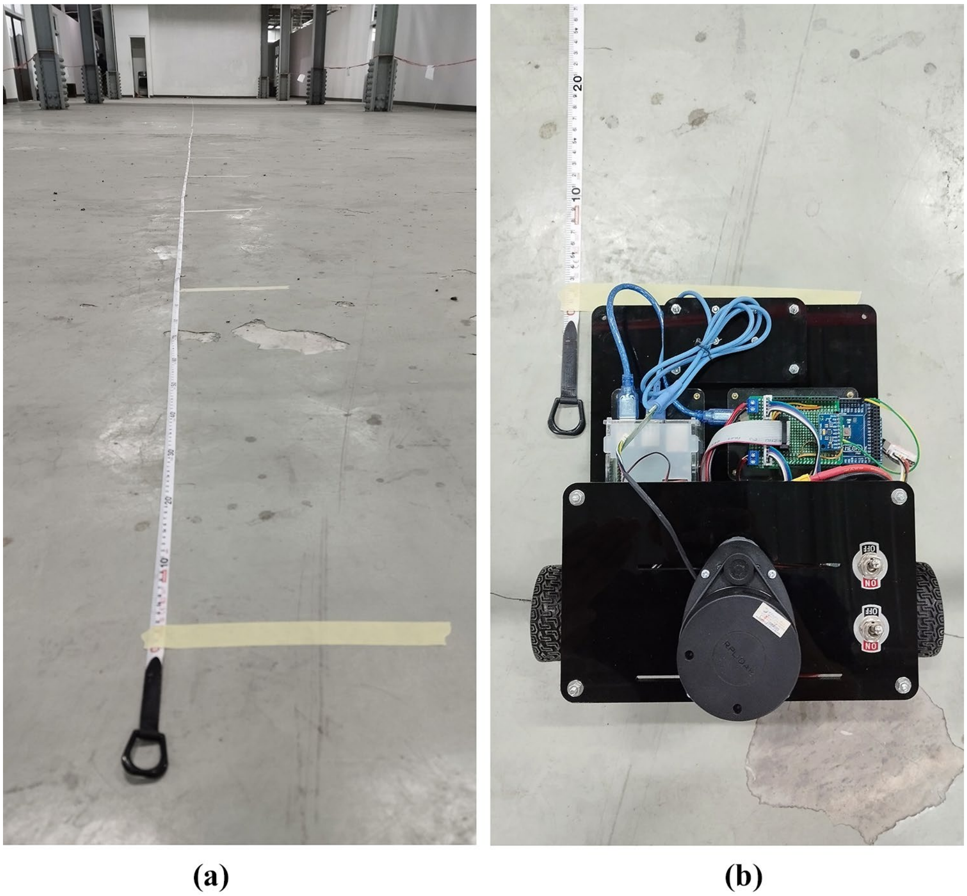


Fig. 10 (**a**) Straight motion testing setup, (**b**) the vehicle placement at the starting line of the testing setup

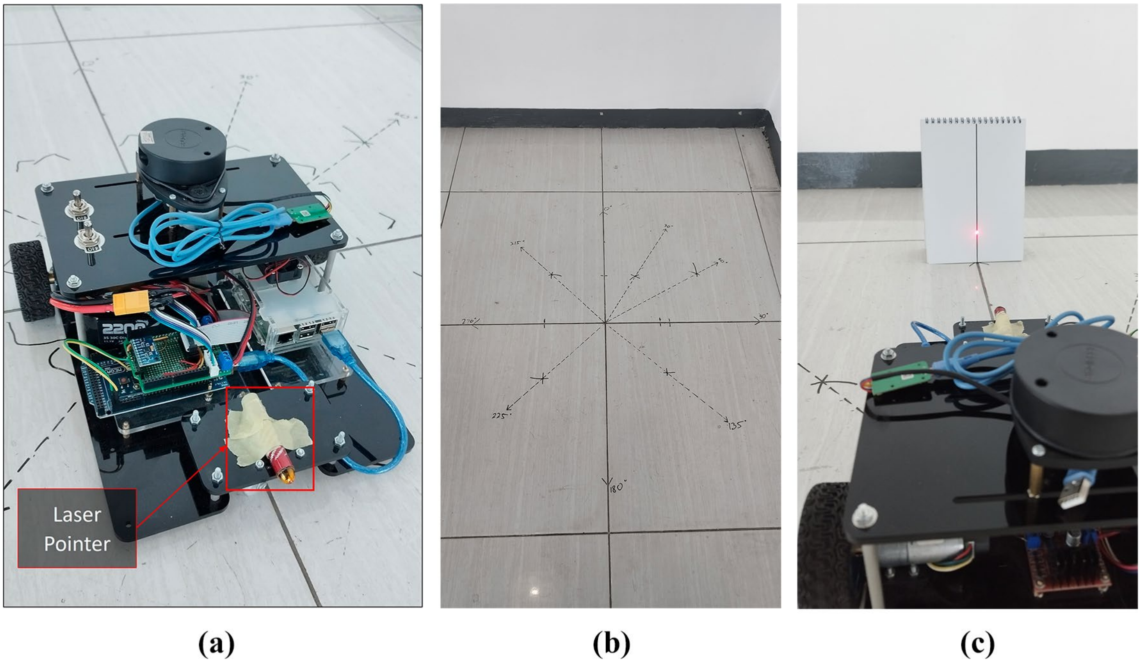


Fig. 11 (**a**) Laser pointer installation on the vehicle, (**b**) area setup for rotational motion testing, (**c**) firing the laser pointer at the vertical line representing the reference angle

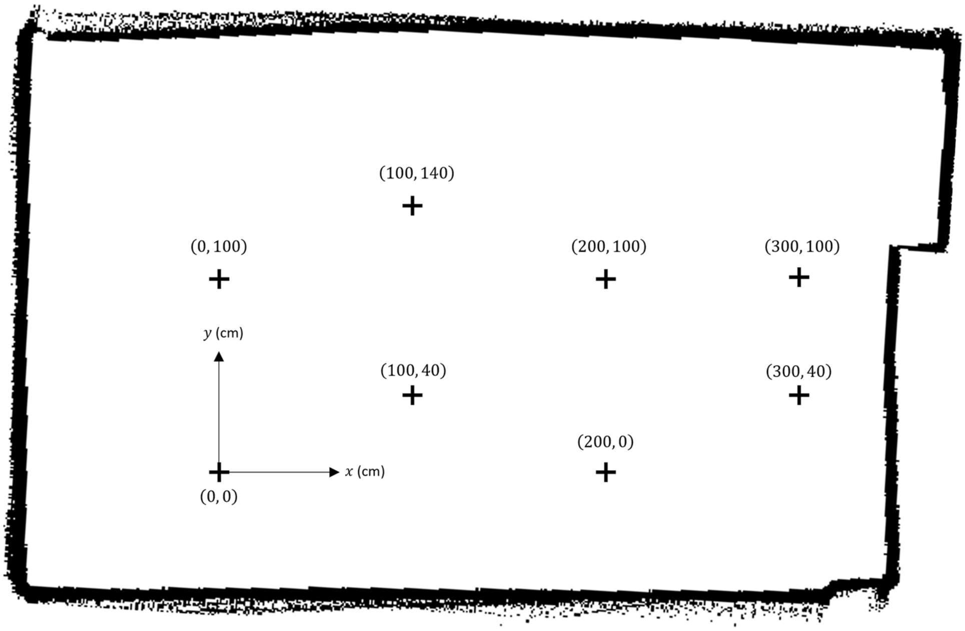


Fig. 12 Localization measurement point in the testing area

repeated up to six times with different routes to each test point to provide insight into measurement repeatability.

5.2 Mapping experiments

Mapping experiments were performed to evaluate both the consistency of the map and the accuracy of the obstacles' dimensions. In the map repeatability test, the vehicle systematically moved through the test area until the SLAM algorithm successfully generated a map. The vehicle was then reset to its initial position, restarting the mapping procedure. The sequence was repeated three times, resulting in three sample map views of the test area. The maps, as images, were subsequently evaluated for similarity using the structural similarity index measure (SSIM) value.

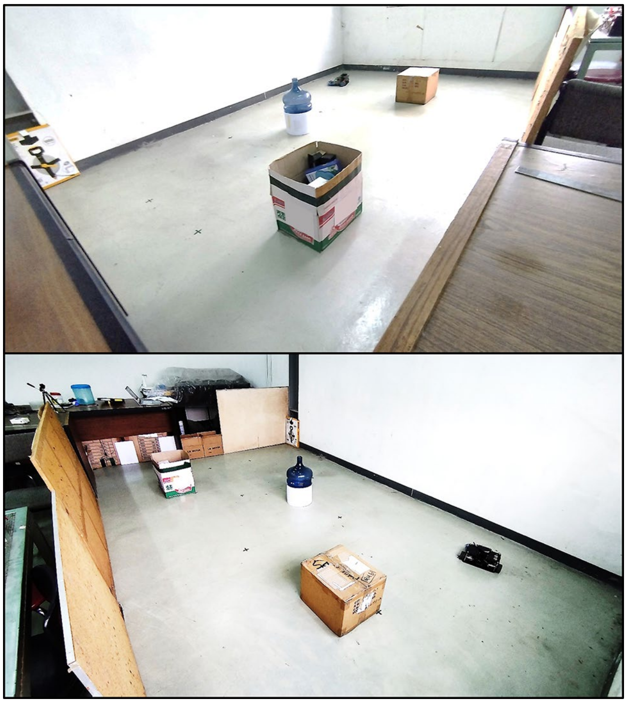


Fig. 13 Mapping test area with obstacle

In addition to the evaluation of the map's repeatability, further evaluation was carried out to investigate the mapping algorithm capability to identify and represent obstacles. This evaluation involves strategically placing different objects in the test area, as shown in Fig. 13. The vehicle navigated the test area until a map, showing the objects represented as obstacles, appeared. A detailed comparison was then conducted to assess the correctness of the mapping by evaluating the size of the objects on the map in comparison to their actual dimensions.

By implementing these sequential testing procedures, we could evaluate the SLAM algorithm capability to map the environment consistently as well as assess its precision in accurately representing the spatial dimensions of obstacle objects.

6 Results and discussion

6.1 Localization experiment results

During the straight motion test, the vehicle travelled 30 m with a rotation angle of 0°. Figure 14 demonstrates a graph showing the linear relationship between the distance measured by the wheel encoder and the reference distance. The results showed a slight discrepancy, where the encoder's measured distance was consistently slightly lower than the reference value. This discrepancy could be attributed to inaccurate encoder orientation readings, where deviations within an error range of [– 5°, 5°] were observed during straight movement.

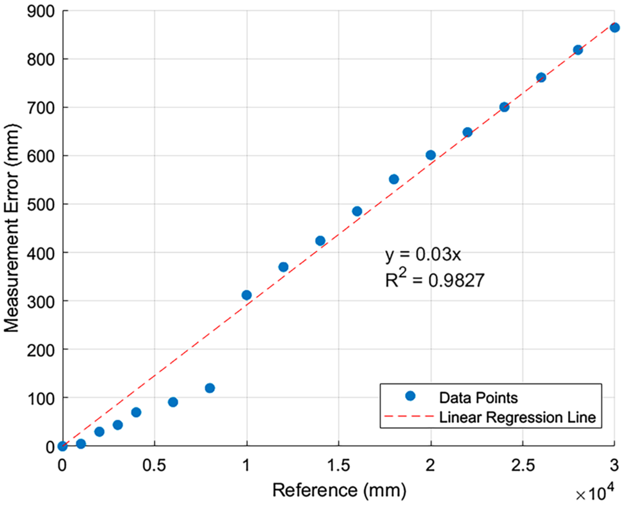


Fig. 14 Graph of the relationship between the error of the wheel encoder distance measurement and the reference distance

Figure 15 shows the position measurement graph of the vehicle straight motion recorded by the IMU. In this scenario, the vehicle advanced 10 cm before stopping. The speed measurement did not return to zero once the vehicle stopped as a result of the inaccuracies in acceleration measurement affecting integration results. This inaccuracy affected the location graph, leading to its continued growth even after the vehicle had come to a stop. When the vehicle stopped, the IMU sensor estimated a distance of approximately 4.2 cm, which was significantly shorter than the actual distance of 10 cm.

In the rotation motion test, the vehicle was subjected to both clockwise and anticlockwise z-axis rotations. In order to determine the orientation of the vehicle, both wheel encoder and IMU sensors were utilised. Figure 16 illustrates the relationship between the orientation estimated by the wheel encoder and the actual relative angle value, while Fig. 17 shows the results of the IMU measurement for the same setup.

Upon analysing the test results, it was observed that the correlation between the reference angle value and the orientation angle reading error exhibited a non-linear trend for both the wheel encoder and IMU readings. The wheel encoder measurement error reached a maximum of 4.38° at an angle of 135° and subsequently fell to 2.24° at an angle of 180°. The IMU measurement error peaked at 2.81° at a measurement angle of 135°, then decreased to 1.69° at a measurement angle of 180°. The IMU had a lower root mean square error (RMSE) of 2.90° than the encoder, which had an RMSE of 4.23°. The results demonstrated

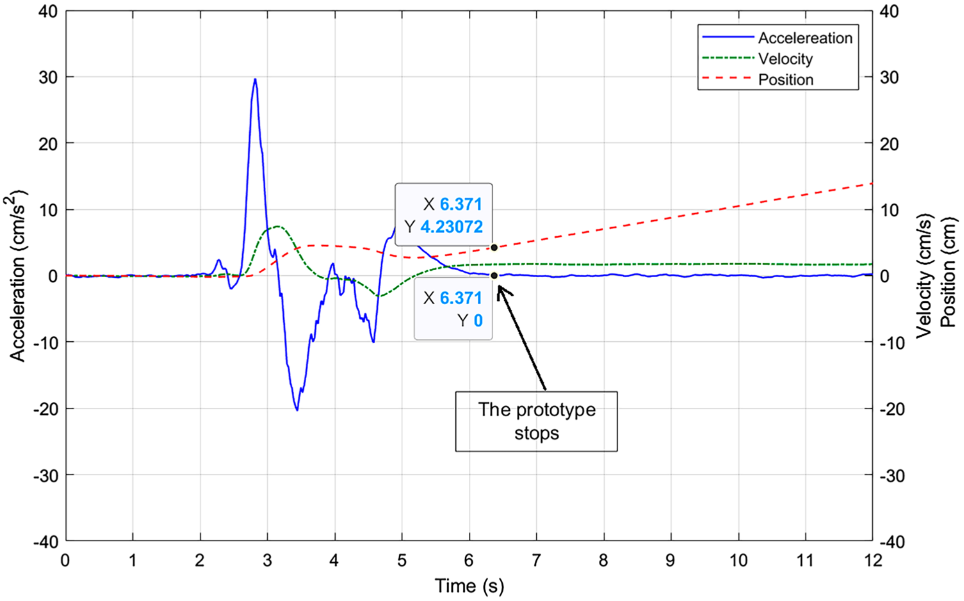


Fig. 15 The result of accelerometer acceleration integration into position

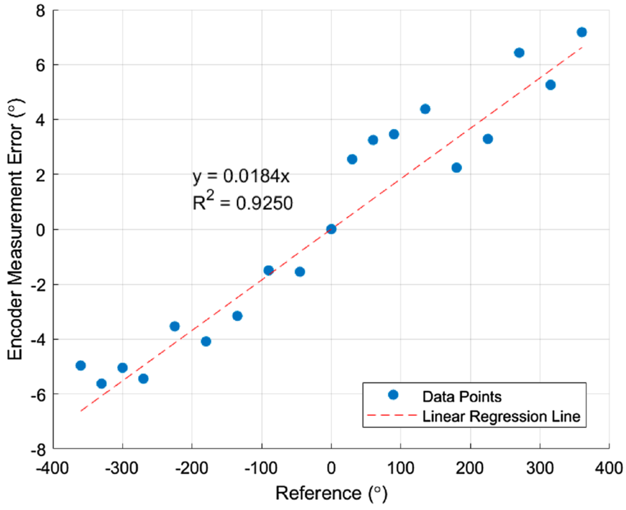


Fig. 16 Graph of the relationship between the angle measurement error from wheel encoder readings with the reference angle value

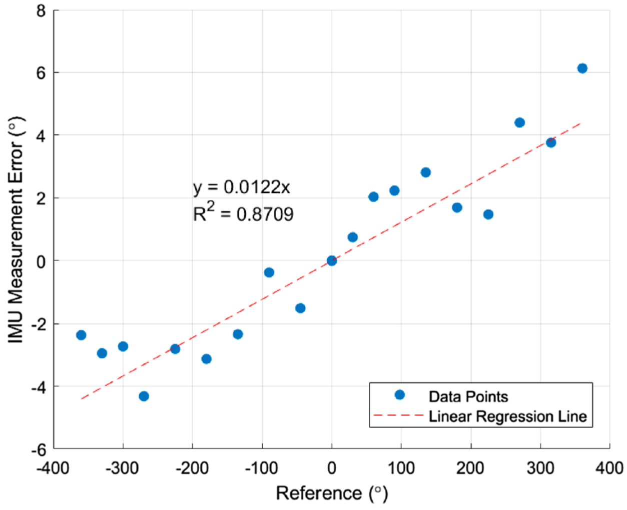


Fig. 17 Graph of the relationship between the angle measurement error from IMU readings with the reference angle value

that the IMU outperformed the wheel encoder sensor in estimating vehicle orientation, making it the preferred choice for measuring orientation angles.

During the SLAM localization test, the vehicle was randomly moved across eight measurement points and subjected to repeated measurements up to 6 times to evaluate its repeatability. Figure 18 demonstrates the results of the localization test utilising only proprioceptive sensors (wheel encoder and IMU), while Fig. 19 shows the results of the proposed localization approach. By utilising the proposed localization method, an average localization error of 8 cm with a standard deviation of 4 cm was achieved, demonstrating a significant 33.3% enhancement compared to relying solely on the wheel encoder and IMU for localization.

Nevertheless, Fig. 19 shows the existence of outliers in the experimental results of the SLAM-based localization approach. The outliers arise from inaccuracies in the SLAM scan matching process, leading to the LiDAR measurement showing a different understanding of the environment compared to the previously mapped environment. The main reason is attributed to the rapid motion of the vehicle, along with a minimum movement requirement for the scan matching procedure that might be 7.5 too small, resulting in an accumulation of scan data within a short period. Consequently, when the buffer reaches its capacity, some scan data

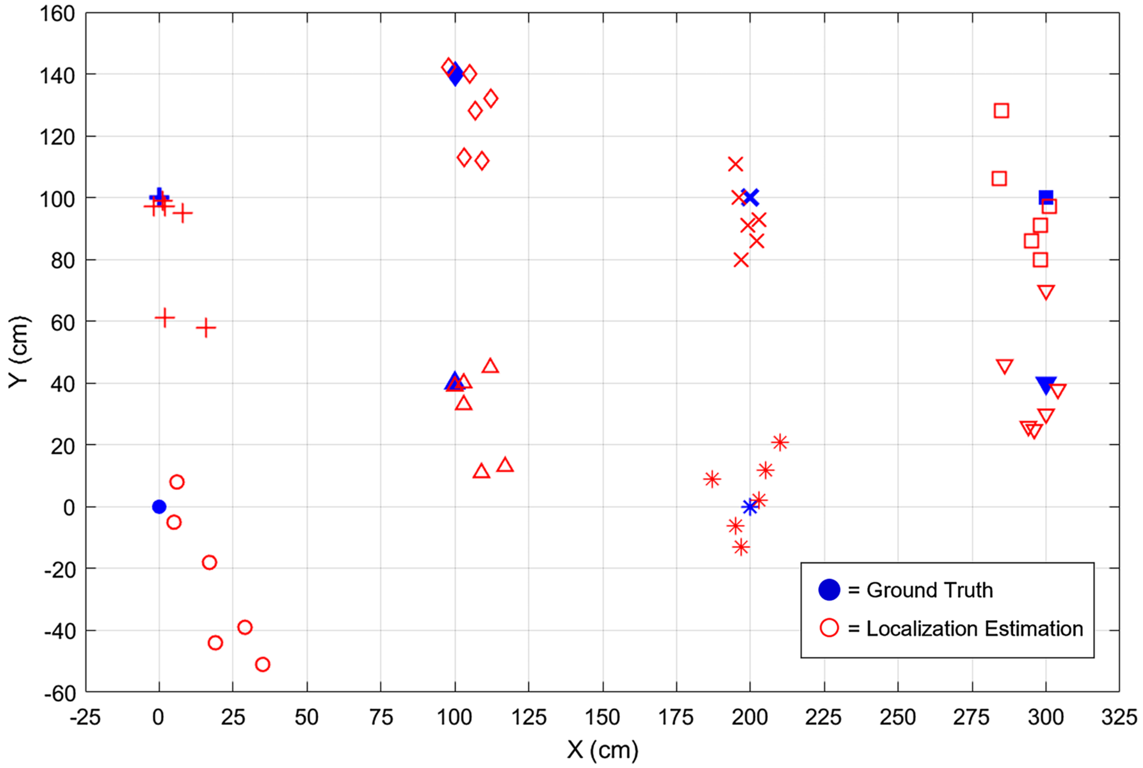


Fig. 18 Localization using proprioceptive sensors only: IMU-wheel encoder odometry (mean error: 12 cm, standard deviation: 9 cm)

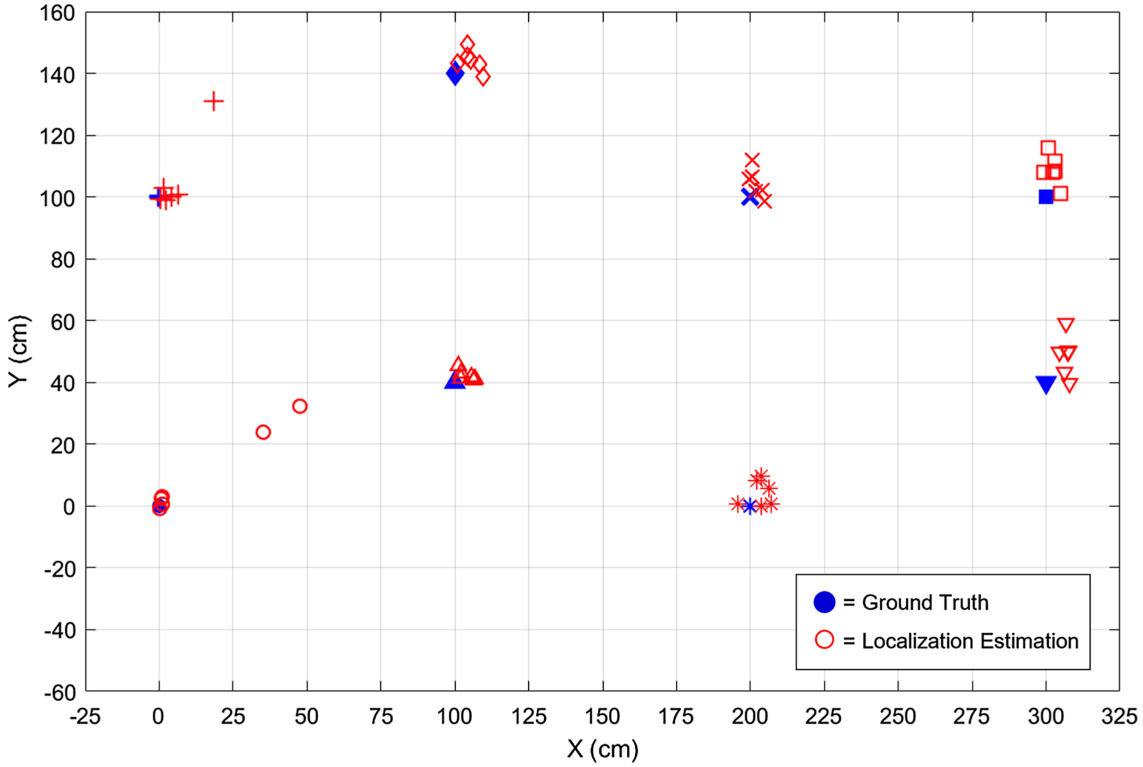


Fig. 19 Localization result using SLAM Lidar with odometry reference from proprioceptive sensors (average error: 8 cm, standard deviation: 4 cm)

is discarded, introducing gaps and inconsistencies in the map due to the loss of information.

To reduce this discrepancy, it is essential to fine-tune critical factors such as the scan data buffer value, the minimum movement required for the scan matching process, and the speed of the vehicle. Having an oversized buffer value can result in significant storage memory consumption, whereas an undersized buffer may cause data loss during rapid vehicle movement. It is essential to carefully adjust the minimum vehicle movement value to ensure that the SLAM algorithm does not overlook processing the nearest scan data, which could lead to missed opportunities for refining the vehicle's position estimate through scan matching.

Furthermore, a high vehicle speed introduces motion blur, which distorts the scanned environment's geometry in comparison to its actual structure. The disparity in data complicates the scan-matching process, making it challenging to interpret environmental geometry consistently. In order to tackle this issue, one possible approach involves using a LiDAR sensor with an increased scan rate. This will facilitate quicker full-circle scans, minimising scanning distortion and maintaining the consistency of creating the environmental geometry.

6.2 Mapping tests results

In the map repeatability test, the vehicle moved through the testing area multiple times to generate the test area map, as shown in Fig. 20. An assessment of the map's accuracy was conducted by comparing the structural similarity index measure (SSIM) values of the three resulting maps. Table 1 presents the analysis of the SSIM values, showing a considerably high average of 80.89%. This result indicates a high degree of similarity among the maps, with variations with nearly imperceptible differences. Therefore, it can be inferred that the utilised SLAM implementation demonstrates excellent accuracy.

Table 1 Structural similarity index measure (SSIM) values between map samples

$SSIM_{12}$	0.8130
$SSIM_{13}$	0.8039
$SSIM_{23}$	0.8097
$SSIM_{avg}$	0.8089

Proceeding to the final test, the accuracy of the implementation of LiDAR-based SLAM in reconstructing obstacle objects was evaluated. In order to accomplish this evaluation, a variety of obstacles were strategically placed in the testing area, including block-shaped obstacles and a one-gallon jug. The vehicle navigated the area, enabling the SLAM algorithm to generate a detailed map, as shown in Fig. 21. Measurements of object dimensions were compared, and the results are presented in Table 2.

Determining object dimensions on the map involved measuring the length of pixel series that make up the objects, using a pixel size of 10 mm × 10 mm. Table 2 presents the error values for object dimensions on the map, ranging from 26 to 50 mm, with an average error of 37 mm. The error can be attributed less precise scan matching, leading to perceived shifts in object positions. As the vehicle travels and performs multiple scans, these inaccuracies accumulate, resulting in the appearance of objects larger than their actual dimensions.

Despite these errors, the LiDAR-based SLAM implementation in this work demonstrated a considerable ability to detect objects as can be seen in the experimental results. The interpretation errors, while potentially making an object appear larger, contribute to enhanced clarity when implementing an autonomous control program. In this case, the vehicle might be able to navigate a more secure route without colliding with obstacles.

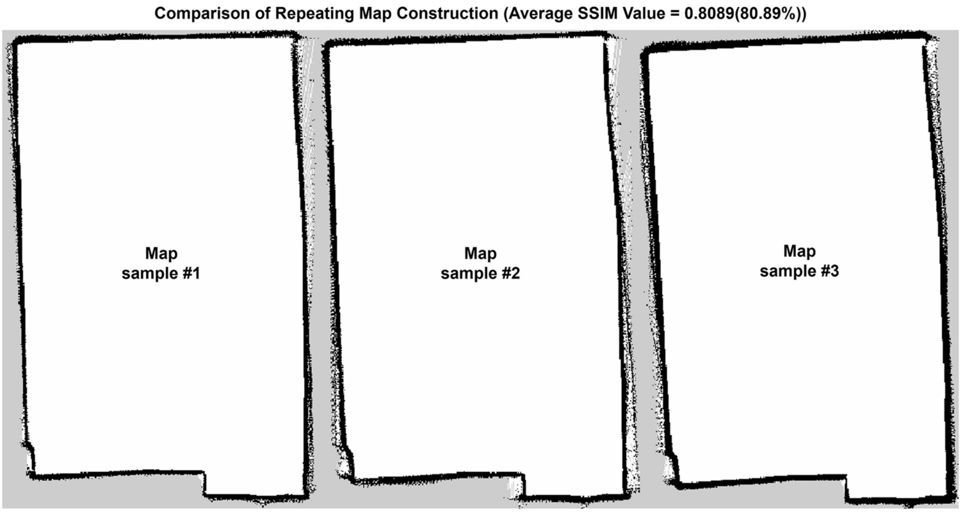

Comparison of Repeating Map Construction (Average SSIM Value = 0.8089(80.89%))
Map sample #1
Map sample #2
Map sample #3

Fig. 20 Comparison of three samples of test area map images of Lidar mapping results

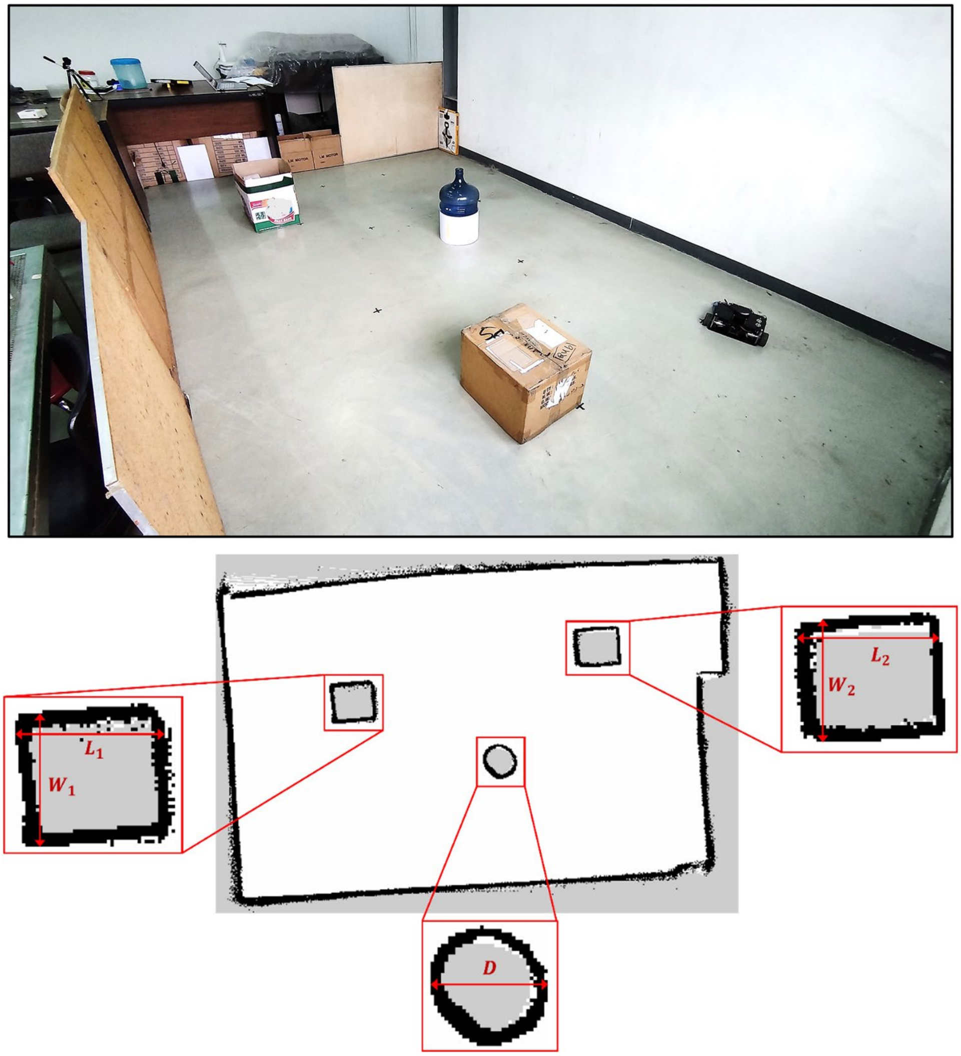


Fig. 21 Map reconstruction result of the test area with obstacles

Table 2 Comparison of obstacle dimension values

Dimension	True Dimension (mm)	Dimension constructed in the map (mm)	Error (mm)
L_1	402	430	28
W_1	345	380	35
D	280	330	50
L_2	394	440	46
W_2	324	350	26
Average error			37

7 Conclusion

The proposed localization system, which combines wheel odometry, IMU, and 2D LiDAR SLAM within a ROS framework, has shown promising results in overcoming sensor imperfections and environmental challenges. The comprehensive assessment and evaluation of each sensor's performance, alongside the comparison of SLAM algorithms in ROS, established a strong basis for development of a reliable and accurate localization system.

Despite considerable challenges in sensor fusion and SLAM implementations, the study provides valuable insights for improving autonomous navigation performance. The evaluation of wheel odometry and IMU, in conjunction with the adoption of the Karto SLAM algorithm, resulted in enhanced localization accuracy, in which the IMU emerging as the preferred choice for measuring the vehicle's orientation. The study also emphasised the importance of optimizing factors, such as scan data buffer sizes and vehicle speed to reduce errors in the SLAM process.

The experimental phase, which included localization and mapping evaluations, highlighted the effectiveness of the proposed approach. In the localization test, the proposed localization method successfully enhanced the accuracy by up to 33.3% compared with only relying on wheel encoders and IMU. The repeatability tests of the constructed map showed a high degree of similarity between map samples which have an average SSIM value of 80,89%, indicating the strong and reliable performance of the SLAM implementation. The research also highlighted the capability of the proposed approach to identify objects and recognising minor interpretation errors that could be utilised to enhance obstacle detection in autonomous control scenarios.

Acknowledgements This research is supported by a research grant, P2MI-2022, Faculty of Mechanical and Aerospace Engineering, Bandung Institute of Technology, Indonesia.

Author contributions A. S. I. prepared the research, conducted the experiments, and wrote the main manuscript text. B. H. directed the research and deepened the analysis, also provided funding for the research. D.K.D directed the manuscript writing format.R. P. S deepened the analysis, reviewed and added state-of-the-art (introduction).Z. A. provided measurement equipment, supervised the implementation of the research, and tested/deepened the methods used.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

Andradi, H., Blumenthal, S., Prassler, E., Plöger, P.G.: Lidar-based indoor localization with optimal particle filters using surface normal constraints. In: 2023 IEEE International Conference on Robotics and Automation (ICRA 2023), pp. 1947–1953 (2023). https://doi.org/10.1109/icra48891.2023.10160274

Aqel, M.O., Marhaban, M.H., Saripan, M.I., Ismail, N.B.: Review of visual odometry: types, approaches, challenges, and applications. Springerplus **5**, 1–26 (2016)

Bhargava, M., Mehta, R., Adhikari, C.D., Sivanathan, K.: Towards development of performance metrics for benchmarking slam algorithms. J. Phys. Conf. Ser. **1964**(6), 062115 (2021)

Brossard M., Bonnabel, S.: Learning wheel odometry and IMU errors for localization. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 291–297 (2019). https://doi.org/10.1109/ICRA.2019.8794237

Chang, L., Niu, X., Liu, T.: GNSS/IMU/ODO/lidar-SLAM integrated navigation system using IMU/ODO pre-integration. Sensors **20**(17), 4702 (2020). https://doi.org/10.3390/s20174702

Dai, Z., et al.: An intensity-enhanced lidar SLAM for unstructured environments. Meas. Sci. Technol. **34**(12), 125120 (2023). https://doi.org/10.1088/1361-6501/acf38d

Fazekas, M., Gáspár, P., Németh, B.: Challenges of the application of front-wheel odometry for vehicle localization. In: 2021 29th Mediterranean Conference on Control and Automation (MED), IEEE, pp. 132–137 (2021)

Fazekas, M., Gáspár, P., Németh, B.: Calibration and improvement of an odometry model with dynamic wheel and lateral dynamics integration. Sensors **21**(2), 337 (2021b)

Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based SLAM. IEEE Intell. Transp. Syst. Mag. **2**(4), 31–43 (2010). https://doi.org/10.1109/mits.2010.939925

Groves, P.D.: Principles of GNSS, inertial, and multisensor integrated navigation systems, [Book review]. IEEE Aerosp. Electron. Syst. Mag. **30**(2), 26–27 (2015). https://doi.org/10.1109/MAES.2014.14110

Han, D., Li, Y., Song, T., Liu, Z.: Multi-objective optimization of loop closure detection parameters for indoor 2D simultaneous localization and mapping. Sensors **20**(7), 1906–1906 (2020). https://doi.org/10.3390/s20071906

He, K., Ding, H., Xu, N., Guo, K.: Wheel odometry with deep learning-based error prediction model for vehicle localization. Appl. Sci. **13**(9), 5588–5588 (2023). https://doi.org/10.3390/app13095588

Hoshi, M., Hara, Y., Nakamura, S.: Graph-based SLAM using architectural floor plans without loop closure. Adv. Robot. **36**(15), 715–723 (2022). https://doi.org/10.1080/01691864.2022.2081513

Huang, L.: Review on lidar-based SLAM techniques. In: 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML), IEEE, pp. 163–168 (2021)

Junior, G.P.C., et al.: EKF-LOAM: an adaptive fusion of lidar SLAM with wheel odometry and inertial data for confined spaces with few geometric features. IEEE Trans. Autom. Sci. Eng. **19**(3), 1458–1471 (2022). https://doi.org/10.1109/tase.2022.3169442

Khan, M.U., et al.: A comparative survey of lidar-SLAM and lidar based sensor technologies. In: 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), IEEE, pp. 1–8 (2021)

Konolige, K., et al.: Efficient sparse pose adjustment for 2D mapping. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 22–29 (2010). https://doi.org/10.1109/iros.2010.5649043

Le, X.S., Fabresse, L., Bouraqadi, N., Lozenguez, G.: Evaluation of out-of-the-box ROS 2D SLAMs for autonomous exploration of unknown indoor environments. Lect. Notes Comput. Sci. (2018). https://doi.org/10.1007/978-3-319-97589-4_24

Li, Q., et al.: Multi-sensor fusion for navigation and mapping in autonomous vehicles: accurate localization in urban environments. Unmanned Syst. **8**(03), 229–237 (2020). https://doi.org/10.48550/arXiv.2103.13719

Liao, M., Wang, D., Yang, H.: Deploy indoor 2D laser SLAM on a raspberry pi-based mobile robot. In: 2019 11th Int. Conf. on Intelligent Human-Machine Systems and Cybernetics (IHMSC), pp. 7–10 (2019). https://doi.org/10.1109/ihmsc.2019.10097

Nimura, M., Kanai, K., Katto, J.: Accuracy evaluations of real-time lidar-based indoor localization system. In: 2023 IEEE Int. Conf. on Consumer Electronics (ICCE), pp. 1–5 (2023). https://doi.org/10.1109/ICCE56470.2023.10043573

Olalekan, A.F., Sagor, J.A., Hasan, M.H., Oluwatobi, A.S.: Comparison of two SLAM algorithms provided by ROS (robot operating system). In: 2021 2nd International Conference for Emerging Technology (INCET), pp. 1–5 (2021)

Pu, H., Luo, J., Wang, G., Huang, T., Liu, H.: Visual SLAM integration with semantic segmentation and deep learning: a review. IEEE Sens. J. **23**(19), 22119–22138 (2023). https://doi.org/10.1109/JSEN.2023.3306371

Qin, J., Liu, Z.: Multi-modal sensor fusion method based on kalman filter. In: 2021 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), IEEE, pp. 515–519 (2021). https://doi.org/10.1109/TOCS53301.2021.9688970

Quan, M., Piao, S., Tan, M., Huang, S.: Tightly-coupled monocular visual-odometric SLAM using wheels and a MEMS gyroscope. IEEE Access **7**, 97374–97389 (2019). https://doi.org/10.1109/access.2019.2930201

Quan, S., Chen, J.: AGV localization based on odometry and lidar. In: 2019 2nd World Conference on Mechanical Engineering and

Intelligent Manufacturing (WCMEIM), pp. 483–486 (2019). https://doi.org/10.1109/wcmeim48965.2019.00102

Sabattini, L., et al.: The pan-robots project: advanced automated guided vehicle systems for industrial logistics. IEEE Robot. Autom. Mag. **25**(1), 55–64 (2018). https://doi.org/10.1109/mra.2017.2700325

Santos, J.M., Portugal, D., Rocha, R.P.: An evaluation of 2D SLAM techniques available in Robot Operating System. In: 2013 IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR), pp. 1–6 (2013). https://doi.org/10.1109/SSRR.2013.6719348

Stimming, C., et al.: Multi-level on-board data fusion for 2D safety enhanced by 3D perception for AGVs. In: 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 239–244 (2015). https://doi.org/10.1109/iccp.2015.7312636

Van Nam, D., Gon-Woo, K.: Solid-state lidar based-SLAM: a concise review and application. In: 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 302–305 (2021). https://doi.org/10.1109/BigComp51126.2021.00064

Velas, M., Spanel, M., Hradis, M. Herout, A.: CNN for imu assisted odometry estimation using Velodyne lidar. In: IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 71–77 (2018). https://doi.org/10.1109/ICARSC.2018.8374163

Voronov, Y., Voronov, A., Makhambayev, D.: Current state and development prospects of autonomous haulage at surface mines. E3S Web Conf. (2020). https://doi.org/10.1051/e3sconf/202017401028

Woodman, O.: An Introduction to Inertial Navigation. Cambridge: University of Cambridge Computer Laboratory (2007). https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf. Accessed 29 Feb 2024

Wu, K.J., Guo, C.X., Georgiou, G. Roumeliotis, S.I.: VINS on wheels. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5155–5162 (2017). https://doi.org/10.1109/icra.2017.7989603

Xing, Z., Zhu, X., Dong, D.: DE-SLAM: SLAM for highly dynamic environment. J. Field Robot. **39**, 528–542 (2022). https://doi.org/10.1002/rob.22062

Xu, X., et al.: A review of multi-sensor fusion slam systems based on 3D lidar. Remote Sens. **14**(12), 2835 (2022). https://doi.org/10.3390/rs14122835

Yan, Y., Zhang, B., Zhou, J., Zhang, Y., Liu, X.A.: Real-time localization and mapping utilizing multi-sensor fusion and visual–IMU–wheel odometry for agricultural robots in unstructured, dynamic and GPS-denied greenhouse environments. Agronomy **12**(8), 1740 (2022)

Yang, M., et al.: Sensors and sensor fusion methodologies for indoor odometry: a review. Polymers **14**(10), 2019 (2022). https://doi.org/10.3390/polym14102019

Zhang, S., Guo, Y., Zhu, Q., Liu, Z.: Lidar-IMU and wheel odometer based autonomous vehicle localization system. In: 2019 Chinese control and decision conference (CCDC), IEEE, pp. 4950–4955 (2019). https://doi.org/10.1109/CCDC.2019.8832695

Zhang, X., Zhang, H., Qian, C., Cao, Y.: A lidar-intensity SLAM and loop closure detection method using an intensity cylindrical-projection shape context descriptor. Int. J. Appl. Earth Obs. Geoinf. (2023). https://doi.org/10.1016/j.jag.2023.103419

Zhao, J., Liu, S., Li, J.: Research and implementation of autonomous navigation for mobile robots based on slam algorithm under ROS. Sensors **22**(11), 4172 (2022). https://doi.org/10.3390/s22114172

Zou, Q., et al.: A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles. IEEE Trans. Intell. Transp. Syst. **23**(7), 6907–6921 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Achmad Syahrul Irwansyah earned a bachelor's degree in mechanical engineering from Bandung Institute of Technology (ITB) – 2023. He currently works as a Control System Engineer in a Hydrofoil Boat Company. His research interests include control theory and autonomous robot navigation. Now, his research is control theory applied to hydrofoil boat's dynamics.

Budi Heryadi obtained master's degree (2013) and Ph.D. (2020) from Bandung Institute of Technology (ITB), in mechanical signal processing research area. He currently works in Bandung Institute of Technology as an Assistant Professor. His current research interests are dynamics control and mechatronics.

Dyah Kusuma Dewi is researcher in Research Center for Smart Mechatronics, National Research and Innovation Agency (BRIN), Indonesia. She obtained her doctoral degree in mechanical engineering major from Bandung Institute of Technology (ITB) – 2023. Her topic's interests are in dynamics analysis of rotating equipment. Now, her research interest widens to mechatronics topic such as implementation of digital twin on machine monitoring system.

Roni Permana Saputra is a Robotics Researcher at the Research Centre for Smart Mechatronics, National Research and Innovation Agency (BRIN). He completed his Ph.D. in Robotics at Imperial College London, specializing in autonomous mobile robot systems for search and rescue operations. He holds a Master of Engineering Science from the University of New South Wales (UNSW), Australia, and a Bachelor of Engineering from Gadjah Mada University (UGM), Indonesia. Currently, he actively conducts R&D in robotics and autonomous systems. His research interests include robotics system integration, intelligent unmanned systems, model predictive control, and robot perception. He is dedicated to enhancing robotic systems' functionality and its applicability in real world scenarios.

Zainal Abidin is a senior lecturer (Professor) in Mechanical Engineering, Faculty of Mechanical and Aerospace Engineering (FTMD), Bandung Institute of Technology (ITB). He received his doctoral degree in 1992 from the University of Salford, UK. Now, he actively conducts research in Dynamics and Control Research Group. His current research interests are in mechatronics, vibration-based predictive maintenance, vibration prediction of scaled rotor bearing system, rotor dynamic balancing method, roller brake tester and automation control driving system.