International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

2D Mapping System using the Google Cartographer SLAM Algorithm and RPLIDAR

Shaik Ruhi Eram, Bestha Jashwanth, Banda Anjali, Paccha Shivani
Department of Electrical & Electronics Engineering, G. Pulla Reddy Engineering College, Kurnool, India.

https://doi.org/10.55041/ijsmt.v2i4.517

Cite this Article: Eram, S. R., Jashwanth, B., Anjali, B. & Shivani, P. (2026). 2D Mapping System using the Google Cartographer SLAM Algorithm and RPLIDAR. International Journal of Science, Strategic Management and Technology, *02*(04). https://doi.org/10.55041/ijsmt.v2i4.517

Abstract: This work presents the design and implementation of a 2D autonomous navigation mapping system which uses SLAM algorithm provided by Google Cartographer and RPLiDAR. The proposed system will allow us to have an estimation about the location of the robot along with the map of the environment at the same time. The system is able to solve some limitations of previous techniques like mapping inaccuracy, drift, and heavy computations. Real-time loop closing and pose graph optimization are some of the techniques used in this system to enhance the mapping ability of the system. Methodology includes simulation with the help of MATLAB in order to analyze sensor data obtained through MATLAB. In addition to that, scan matching and path estimation are two other techniques that are employed. After that, implementation of a real-time algorithm using RPLiDAR is done. The results obtained from the mapping procedure are highly accurate.

Key Words: SLAM, 2D Mapping, Google Cartographer, RPLiDAR, Autonomous Navigation, Robotics

I.Introduction

LiDAR (Light Detection and Ranging) is a remote sensing method that utilizes laser light to calculate the distances between a detector and the objects around it using the principle of time-of-flight. The method calculates the time it takes for pulses of light to bounce off objects and back to generate accurate two-dimensional or three-dimensional maps known as point clouds. LiDAR is considered very useful due to its high accuracy, quick operation, and independence of any lighting conditions for indoor and outdoor operations.

SLAM allows a robot to create a map of an unknown environment while localizing itself within the environment, particularly indoors where GPS cannot be utilized. The Google Cartographer framework can be used to generate real-time two-dimensional and three-dimensional maps of the environment by analyzing the data from the robot's sensors including LiDAR and IMU.

Distance information captured by the LiDAR sensor in two-dimensional mapping involves RPLiDAR that is then converted to occupancy grid map containing information about the obstacles and free spaces. Google Cartographer with the help of RPLiDAR is an affordable yet highly accurate method, with the drift being significantly minimized. The design of the system involves hardware integration and use of ROS2.

II. Material And Methods

Literature Review and Research Gap:

Several studies have been conducted on SLAM in the context of mobile robots. These studies gave rise to the creation of the most popular approaches applied in classical SLAM, which include extended Kalman filter SLAM approach, FastSLAM approach, and SLAM approaches based on graph theory. EKF SLAM approach is an algorithm that

estimates the pose of the robot and map features simultaneously using probabilistic models in conjunction with linearization processes. This approach is recognized for its computationally expensive calculations and linearization errors. FastSLAM approach employs particle filter approach in the estimation of the pose of the robot and separately in the mapping of the landmarks. The disadvantage with FastSLAM approach is particle depletion problem.

SLAM has been extensively analyzed by "LiDAR based SLAM for robotic mapping: state of the art and new frontiers," written by Xiangdi Yue et al. (2023) in which the approaches have been divided into filter-based SLAM, optimization-based SLAM, and learning-based SLAM. As per the authors, the graph-based SLAM possesses high accuracy and consistency owing to its loop closures and optimization. This same topic has also been elaborated on by Yang & Li (2019) and M. U. Khan et al. (2021), discussing the importance of LiDARs in current-day SLAM.

The early works on the topic are of "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," where an efficient mapping approach based on particle filters is proposed. Another work is that of "Hector SLAM" in which scan matching can be used without using any odometry measurements.

Graph-based approaches have become widely used quite recently. For instance, W. Hess et al. ("Real-Time Loop Closure in 2D LIDAR SLAM") show how Google Cartographer has been upgraded to remove the issue of drift with the help of submap creation, scan matching, and pose graph optimization. As previously stated, according to Cui Zhi & Shi Xiumin (2019), even inexpensive LiDARs could be efficiently utilized in Cartographer.

There are, however, a few drawbacks to be considered. These include the expensive cost of sensors, high computational requirements, and sensitivity to noise. Almost none of the studies make use of simulation or real-time hardware to achieve mapping. To overcome these problems, the goal of this paper is to construct a 2D SLAM system using Google Cartographer and RPLiDAR.

System Design Consideration: The system design is such that it uses sensors in real-time mapping and localization through LiDAR. Components have been chosen keeping accuracy, cost, and ease of integration with the embedded system in mind. This will help ensure that data collection and localization are done efficiently.

System Configuration Method

The system uses LiDAR, IMU, and odometry sensors connected to the Raspberry Pi 4 as the main computer. LiDAR performs continuous scanning of the surroundings, providing distance measurements. The IMU generates orientation and motion data. Odometry is computed by the displacement caused by the motor's movement.

The data gathered from the sensors is first processed and analyzed using MATLAB to examine scan matching, filtering, and trajectory calculations. The processed data is used for the SLAM algorithm through ROS2. Localization and mapping are conducted in real time using the SLAM algorithm (Google Cartographer).

System Configuration

- **System A:** Sensor unit (LiDAR + IMU + odometry) for environment perception and motion estimation
- **System B:** Processing unit (Raspberry Pi 4 with ROS2) for data processing, SLAM execution, and decision making
- **System C:** Mapping and output unit for visualization of occupancy grid maps and navigation

Procedure Methodology

The system continually gathers environmental and motion measurements via the sensors LiDAR, IMU, and odometry. After the gathering process, data analysis takes place so that distance, orientation, and motion can be determined. MATLAB is used at first for simulation and validation of data processing, scan matching, and trajectory determination. After data processing, the next step is the SLAM algorithm running in the ROS2 environment to carry out localization and mapping concurrently. Analysis of scans carried out here includes obstacle identification, robot positioning, and map updating.

Mapping occurs in form of the 2D occupancy grid where obstacles and free space in the environment are shown. This happens in real time and ensures accurate mapping and reliable navigation. Loop closure detection is also carried out to reduce drifting of the map.

MATLAB Simulation

The 2D map generation is implemented and tested in MATLAB R2024a using various toolboxes like Navigation Toolbox, Lidar Toolbox, and Computer Vision Toolbox. The simulation considers a scenario where a vehicle with a LiDAR sensor is used for navigating a vehicle through a 3D virtual environment (for instance, parking lot scenario). LiDAR point clouds are computed and analyzed in real-time to compute the trajectory of the vehicle and generate the map using SLAM algorithms.

A systematic simulation process involves initial pre-processing of the raw 3D point cloud followed by converting the point cloud to 2D LiDAR scan information. Relative motion or odometry is computed by employing scan matching techniques like phase correlation and scan matching. Finally, by using the function 'buildMap', the poses are computed and used to create the occupancy grid map.

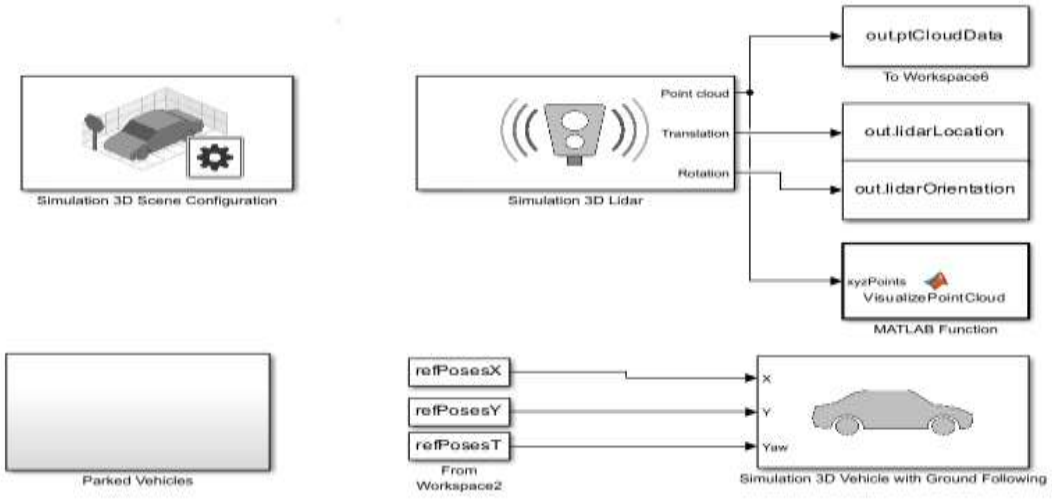


Fig 1. MATLAB Modelling

Features

- **3D Simulation Environment:**

A The software creates a simulated virtual environment such as an Unreal Engine parking lot to replicate the actual driving experience.

- **LiDAR Scan Generation:**

Collecting a series of LiDAR scan data that contains the distances between objects within the surrounding environment.

- **Point Cloud Visualization:**

The point cloud created by scanning is visualized in the form of a bird's eye-view that allows environmental assessment.

- **Vehicle Trajectory Tracking:**

The software computes vehicle movement based on scan-matching and pose estimation algorithms.

Outputs

- **Point Cloud Data:**

High-fidelity 3D spatial data collected by LiDAR sensors.

- **LiDAR Position and Orientation:**

Computed LiDAR sensor positions (x,y) and orientations (θ) at specific times based on odometry.

- **Occupancy Grid Map:**

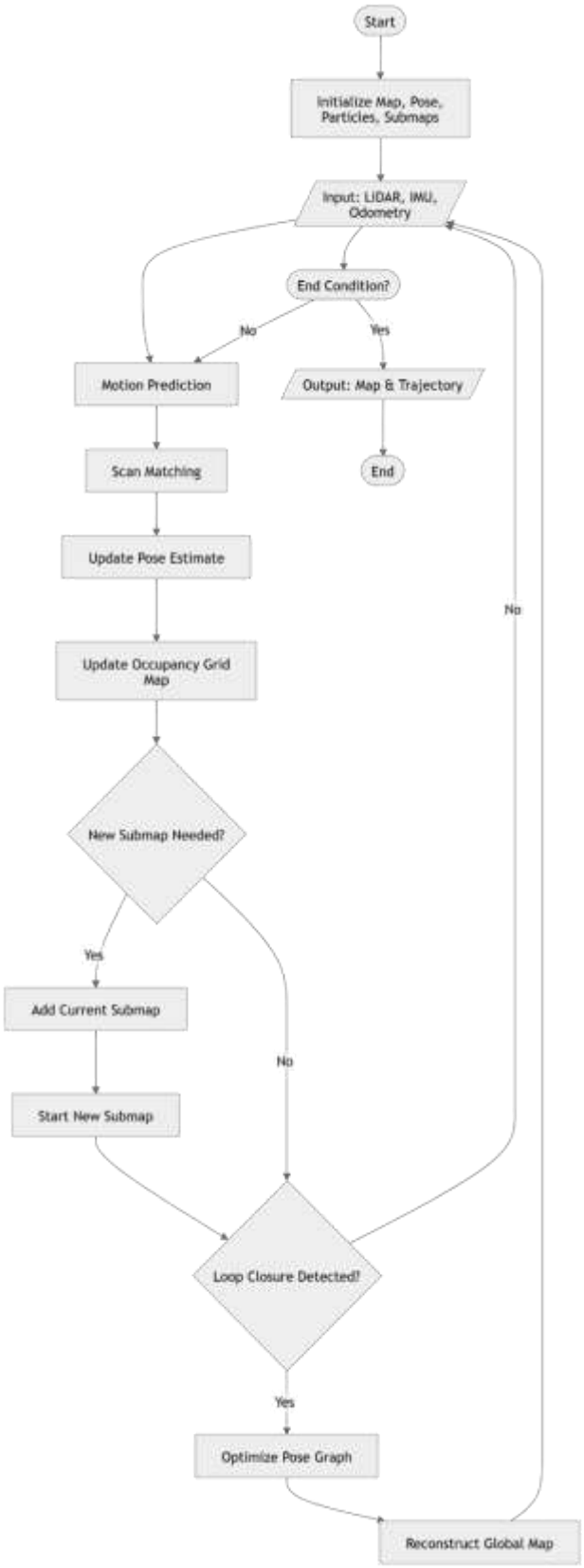
A map generated using LiDAR scan data and poses indicating occupied and unoccupied areas.

This MATLAB simulation provides an accurate testing ground for SLAM algorithms and validation of maps and sensor behaviors before hardware implementation.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Methodology (Cartography SLAM Algorithm)

The proposed Cartographer-based SLAM methodology follows a structured pipeline to achieve accurate localization and mapping by combining motion modeling, sensor correction, and map optimization.



1 Motion Prediction

The current pose of the robot is first estimated by a motion model using control inputs such as linear and angular velocities. Here, we try to predict the future state (pose) of the robot based on the past poses.

$x_t = x_{t-1} + v_t \cos(\theta_{t-1})\,\Delta t$

$y_t = y_{t-1} + v_t \sin(\theta_{t-1})\,\Delta t$

$\theta_t = \theta_{t-1} + \omega_t\,\Delta t$

where:

- $v_t \rightarrow$ linear velocity (from motor/ odometry)

- $\omega_t \rightarrow$ angular velocity (from IMU)

2 Scan Matching

This involves LiDAR scan matching to get a better estimation of pose. Here, the predicted pose from motion prediction is corrected for drift based on scan matching.

Equation wise:

$X_t = \arg\min_X \sum_k \|z_k - h(X, m)\|^2$

3 Pose Estimation

The pose of the robot can be estimated based on the scan-matching process corrected results. This process combines motion prediction with sensor input for more accurate localization and consistency in trajectories estimation. Each particle represents a pose: $X_t^{(i)} = (x_t^{(i)}, y_t^{(i)}, \theta_t^{(i)})$

Weight update: $w_t^{(i)} \propto p(z_t \mid X_t^{(i)}|, m)$

4 Occupancy Grid Mapping

The environment is modeled as an occupancy grid that shows the probability of the corresponding cells to be occupied. This model is constructed based on incremental observations.

Mathematically:

$l_t(m_i) = l_{t-1}(m_i) + \log\{p(m_i \mid z_t, x_t)\} / \{1 - p(m_i \mid z_t, x_t)\}$

where:

- $l_t(m_i) \rightarrow$ log-odds of occupancy

- $m_i \rightarrow$ grid cell

5 Submap Management

In order to increase the speed of the computation of the map the global map is split into submaps. Each submap is optimized independently and therefore provides faster computation time. This methodology will allow building a precise map and localization with SLAM algorithm by means of prediction, correction and optimization steps.

Mathematically: $\min_X \sum_{i,j} \|z_{i,j} - f(X_i, X_j)\|^2$

Fig 2. Cartography Algorithm Flow Chart

6 Loop Closure Detection

The visited places can be recognized by scanning similarities. After recognizing them, loop closure constraints can be applied for optimizing the whole trajectory. Through this methodical approach, real-time map building and localization can be done successfully with the help of prediction, correction, and optimization procedures.

Mathematically: $\min_X \sum_{i,j} \|z_{i,j} - f(X_i, X_j)\|^2$

This is Graph SLAM optimization

Noise Filtering

Noise filtering processes are employed to increase the quality and accuracy of the map generation by utilizing the unfiltered LiDAR data for SLAM calculations. The data obtained from LiDAR systems can be distorted due to various factors in the environment, such as the presence of reflective surfaces, noise, and moving objects.

- **Outliers Rejection:**

Aberrant data points differing greatly from other measurements are rejected. It helps avoid incorrect identification of obstacles and creates a more uniform scan result.

- **Signal Smoothing:**

Various filtering algorithms, including moving averages and Gaussian filters, are applied to minimize oscillations in the data from the sensor and produce more consistent scans.

- **Measurements Correction:**

Errors due to the imprecise calibration and operation of the sensors are addressed through correction mechanisms to improve the distance measurement accuracy.

As a result, noise filtering leads to higher levels of consistency when building a map.

Hardware Implementation

In terms of the hardware configuration, the whole system is built to have sensors, processors, and actuators work together in order to achieve real-time map building and navigation through SLAM.

Components

- **RPLiDAR A1 (8 Hz, 12 m range):**

Provides 360° laser scans of the environment for obstacle detection and mapping.

- **Raspberry Pi 4 (4GB RAM):**

Acts as the central controller, running ROS2 and the SLAM algorithm.

- **IMU Sensor:**

Serves as an input sensor to enhance localization precision.

- **DC Geared Motors (×4):**

Assist in moving the robot.

Need for ROS in This Work

The proposed system utilizes several pieces of software which need to constantly communicate and work together in real time – from data obtained from LiDAR devices to SLAM processes to creating visualizations of maps. This task is made possible with ROS due to the distributed nature of this framework which utilizes interaction between autonomous nodes and allows for the use of data streams between these nodes. Therefore, the use of ROS simplifies many processes which otherwise should have been managed manually, increasing complexity of the system.

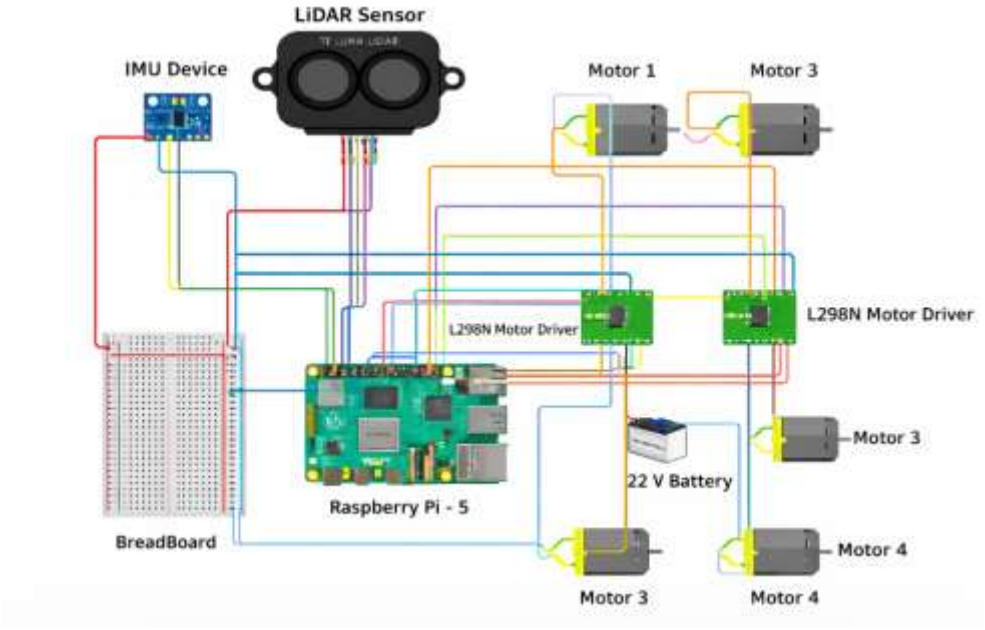


Fig 3. Circuit Diagram

`

III. Result

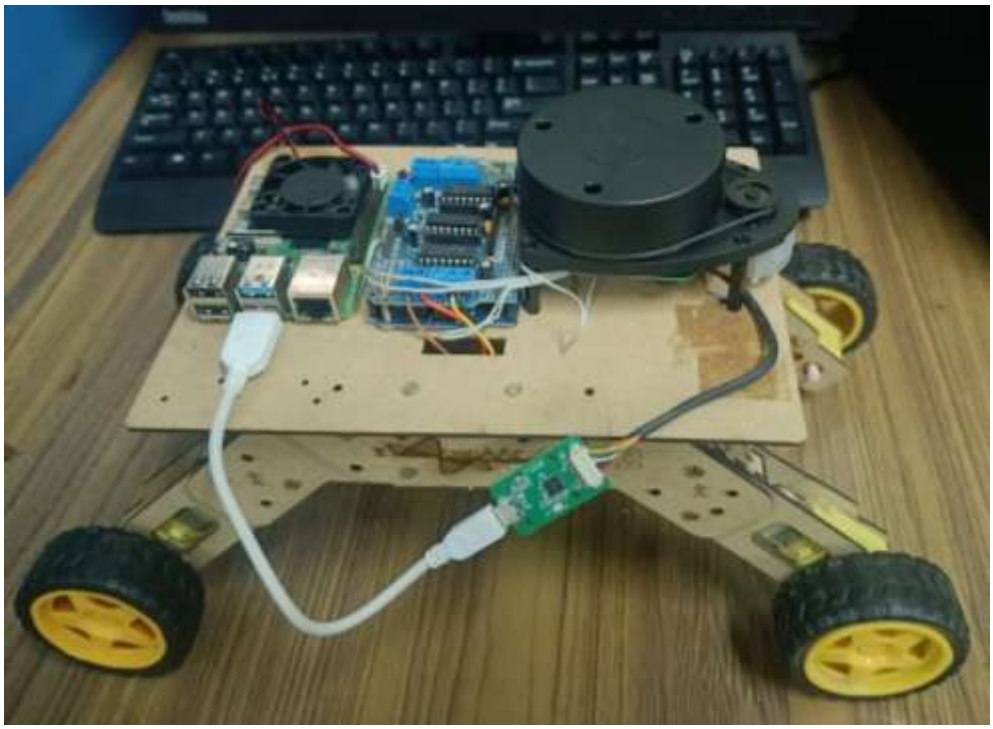


Fig 4. RPLiDAR and Raspberry Pi mounted on a 4 Wheel Robot

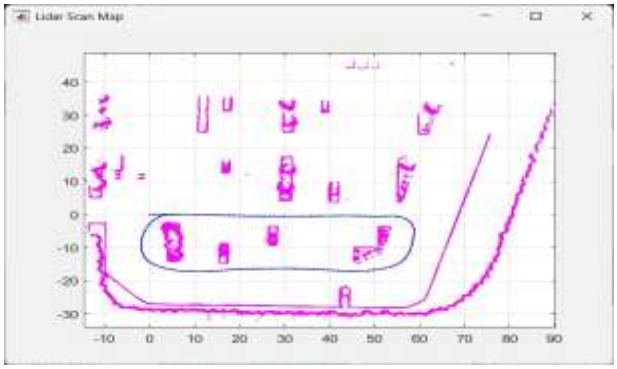


Fig 5. MATLAB Simulation Results

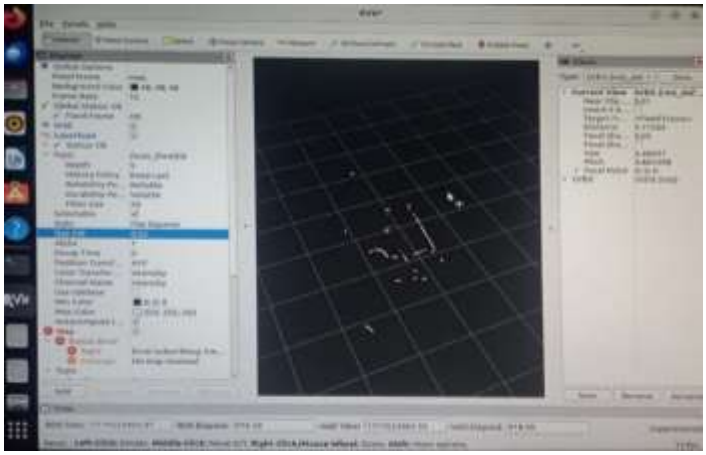


Fig 6. Laser Scan Map using ROS2 and Rviz

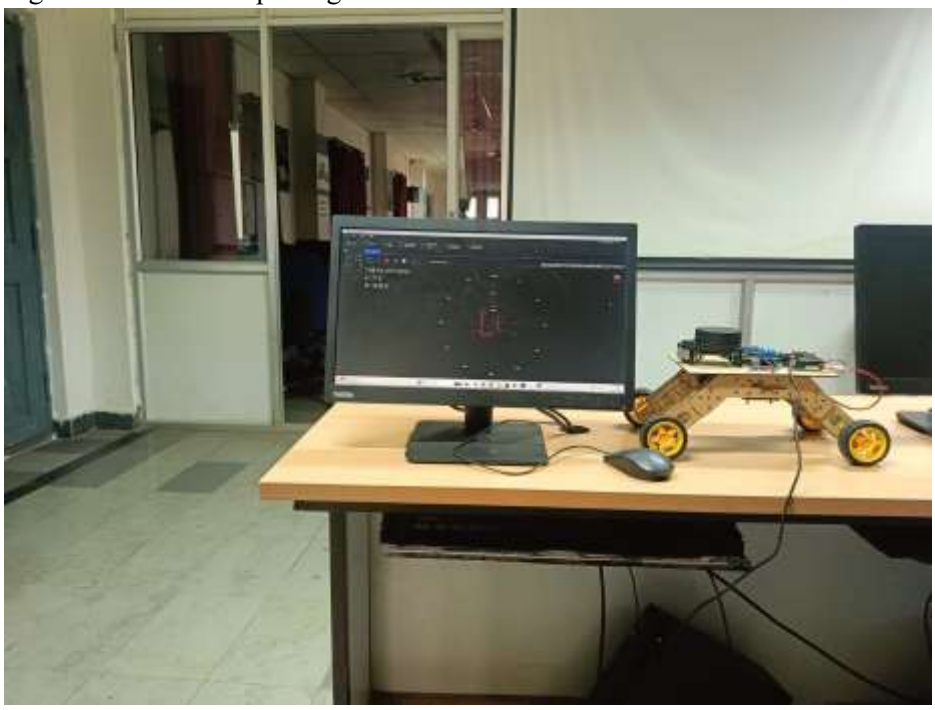


Fig 7: Experimental Setup of LiDAR

IV. Conclusion

The development of a 2D mapping system by incorporating Google Cartographer SLAM technology and RPLiDAR, enabling efficient real-time localization and mapping for autonomous navigation of robots. Through the combination of LiDAR sensing technology, inertial measurement unit information, and odometry information in a single SLAM process, the 2D mapping system efficiently creates occupancy grid maps while simultaneously tracking the robot's path.

The application of Google Cartographer algorithm was very successful since it includes advanced functions like scan matching, submaps-based mapping, loop closure, and pose graph optimization, which greatly minimize drift issues that occur with conventional SLAM algorithms. Furthermore, motion model and probabilistic map helped increase the accuracy of localization in SLAM processes.

The simulation carried out with MATLAB laid a solid groundwork for verification of sensor data processing, scan matching, and trajectory estimation algorithms before implementing the hardware setup. The combination of both approaches facilitated not only theoretical validation but also practical testing of the algorithms. The developed hardware system using RPLiDAR A1 and Raspberry Pi 4 proved that even an inexpensive system can provide good results.

Noise reduction techniques applied, including filtering outliers and data smoothing, contributed to reliable and consistent processing of LiDAR data. As a result, scan matching accuracy was significantly increased, improving the map construction. Furthermore, incorporation of the ROS2 platform made it possible to communicate efficiently between all system components.

Overall, the system achieved:

- Precise 2D occupancy grid map creation
- Elimination of localization errors and drift
- Reliability in highly dynamic and noisy situations
- An affordable and scalable solution

This paper underscores the practicality of implementing low-cost LiDAR SLAM systems in a variety of real-world applications, including indoor navigation, obstacle detection, and service robotics.

References

1. LiDAR-based SLAM for robotic mapping: state of the art and new frontiers" – Xiangdi Yue, Miaolei He, Yihuan Zhang, 2023 (arXiv:2311.00276)
2. Cartographer – "Real-time loop closure in 2D LIDAR SLAM" – W. Hess et al., ICRA 2016
3. F-LOAM – "F-LOAM : Fast LiDAR Odometry and Mapping" – H. Wang et al., IROS 2021
4. "Research on Cartographer Algorithm Based on Low-Cost LiDAR" – Cui Zhi, Shi Xiumin, IJERT, Vol. 8 Issue 10, 2019
5. "Evaluation of Lidar-based 3D SLAM algorithms in SubT environment" – A. Koval, C. Kanellakis, G. Nikolakopoulos, 2023 (IFAC)
6. "LiDAR-based SLAM for robotic mapping: state of the art and new frontiers" – Xiangdi Yue et al., 2023
7. LeGO-LOAM – "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain" – T. Shan, B. Englot, IROS 2018
8. "A Comparative Survey of LiDAR-SLAM and LiDAR-based Sensor Technologies" – M. U. Khan et al., IEEE MAJICC 2021
9. "A Survey of SLAM Research based on LiDAR Sensors" – Yang & Li, 2019
10. "Evaluation of Lidar-based 3D SLAM algorithms in SubT environment" – A. Koval, C. Kanellakis, G. Nikolakopoulos, 2023 (IFAC)
11. "Hector SLAM: A Real-Time SLAM Approach for UAVs" – Kohlbrecher et al. (2011)
12. "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters" – Grisetti et al. (2007)
13. "Real-Time Loop Closure in 2D LIDAR SLAM" – Hess et al. (2016)
14. "RTAB-Map as an Open-Source Lidar and Visual SLAM Library" – Labbé & Michaud (2019)
15. "Benchmarking the Localization Accuracy of 2D SLAM Algorithms" (2020)