Grenze International Journal of Engineering and Technology, Jan Issue

Hector SLAM Mapping and Localization System using ROS and LIDAR

Medha Patil[1], Aarti Thakur[2], Prof. Priyanka Bagul[3] and Prof. Yogita Ajgar[4]
[1-4]Department of Electronics and Telecommunication Engineering, Modern Education Society's College of Engineering, Pune, Maharashtra, India
Email: medha1022@gmail.com, taarti197@gmail.com, priyanka.awasare@mescoepune.org, yogita.ajgar@mescoepune.org

***Abstract*— This paper demonstrates the use of laser technology in environment scanning for performing indoor positioning and mapping. A 360-degree LiDAR forms the core of the project and the input device used for understanding its surrounding. Hector SLAM and ROS are the two technologies implemented in this project. Hector SLAM where SLAM stands for Simultaneous Mapping and Positioning, is the algorithm used to map the environment and improve navigation accuracy. The aim of this project is to allow an autonomous mobile robot to understand an unknown environment and map its surroundings to perform further tasks using the created map.**

***Index Terms*— SLAM, ROS, lidar, map, scan.**

I. Introduction

We are in the age of automation, where technology has become a lifestyle. From home applications to military operations, technology has advanced. And with advancement, come new features to make functions easier for humans to use. Features like safety, visualization, and navigation have become primary elements. One such feature used in most of these applications is environment mapping. This is the most crucial factor in robot vacuum cleaners as well as military robots. Environment Mapping is used by machines (robots) to visualize and understand their environment.

The proposed system involves the use of the Hector Slam Algorithm that uses LiDAR for Simultaneous Mapping and Positioning. The system prototype is demonstrated with the aid of a small 4-wheeled vehicle imitating an indoor warehouse robot. This robot can deliver parcels from sources to various destinations by mapping and finding the best path in a warehouse and is chosen for demonstration as it is one the major applications of the used technologies. There are various systems that are based on GPS and facilitate autonomous robots and vehicles for accurate positioning when used outdoors. However, these systems are unavailable to perform the same tasks indoors. For an instance, consider moving packets in a warehouse or floor-cleaning robots require real-time accurate mapping of the environment they are working in. The implementation process follows - Laser scanning, implementation of Hector SLAM, robot control, autonomous navigation.

The paper is further organized as follows. Section II describes the system design and architecture; Section III demonstrates the experimental setup and results. Section IV discusses the possibilities for future work and the paper is finally concluded.

II. System Design And Architecture

A. System Architecture

Grenze ID: 01.GIJET.10.1.4

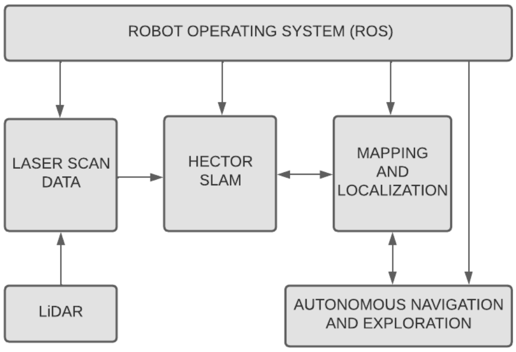


Figure 1. System Architecture

The system uses Robot Operating System for integrating and implementing all the different robotic operations requires for the project like Hector SLAM and LiDAR. The laser data from the LiDAR is used to generate a map of the environment and show the location of the robot on the map. Hector SLAM is responsible for localization and mapping. During the mapping process, the robot autonomously navigates around the area to create a more accurate map. This navigation is also done with respect to the obstacles that are identified using the LiDAR. Fig.1 demonstrates the complete system structure.

B. Software Development

ROS is the most important software used for controlling all the robotic operations. It is run on Raspbian OS for this demonstration but it can run on Ubuntu desktop or Linux OS as well. Hector SLAM is implemented on ROS and the RPLidar libraries are also integrated in ROS.

The recorded map is displayed on RViz application in real-time. Refer fig. 2.

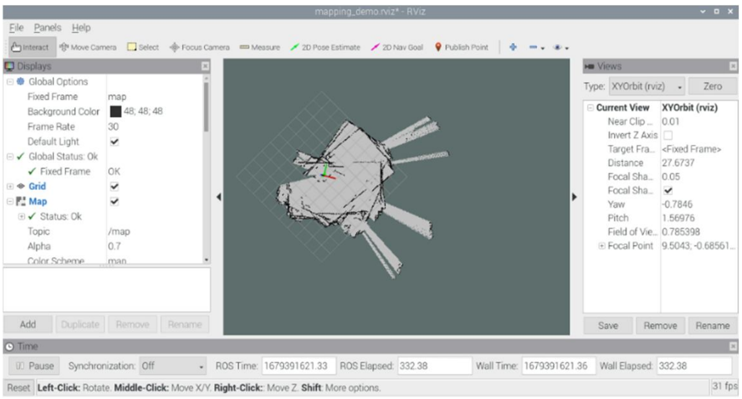


Figure 2. RViz

1. Hector SLAM

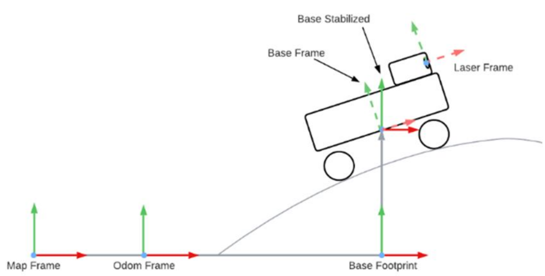


Figure 3. Hector SLAM

Hector SLAM algorithm is used since it gives reliable and accurate navigation.[1] Other grid-based SLAM algorithms like TinySLAM and GMapping require odometer information whereas Hector SLAM uses only laser scan data and previous map data (scan matching technique) to estimate the movement and location of the robot. This means that this algorithm reduces the need for extra sensors cameras and ultrasonic. Odometer information can be used to improve efficiency and reduce sensor error.

Fig. 3 demonstrates the system frames of interest for a robot with roll and pitch motion.

Hector SLAM uses scan matching technique. The map frame is the previous map data. Odometer frame is used to improve accuracy of 2D mapping. If there is no use of odometer, the map frame can be directly transformed to base frame. Base footprint and and base stabilized frames are solely used to represent robot location and height respectively. These frame values are equal with no roll and pitch. The laser frame is the current map data. All the previous frame data is evaluated and matched with the laser frame.[2]

The robot's position and velocity are estimated by this approach using the Gauss-Newton iteration formula and the robot's body transformation from previous map. The previous map data and scan data is matched.

$$RobotPose : Q = \left(P_x, P_y, \Psi\right)T \qquad (1)$$

$$Q^* = argmin_Q \sum_{i-1}^{n} \left[1 - M\left(S_i(Q)\right)\right]^2 \qquad (2)$$

where function S_i represents the coordinates of the i^{th} scan when the pose is i. M represents Map values at coordinates given by S.

With the initial value of Q, an updated value $(Q + \Delta Q)$ is evaluated from $M\left(S_i(Q)\right)$ using Taylor Expansion which gives,

$$\Delta Q^* = H^{-1} \sum_{i-1}^{n} [\nabla M(S_i(Q) \frac{\partial(S_i(Q))}{\partial Q}] \qquad (3)$$

where H is the Hessian matrix.[3]

2. Autonomous Navigation

The software development also involves autonomous navigation. This uses the Lidar data to avoid obstacles and allows efficient mapping of the area. Fig. 4 shows the architecture for autonomous navigation.

The robot uses hector_navigation package of autonomous navigation and exploration. Here the elevation data (height map) is taken along with the 2D map created by the LiDAR and combined to develop an combined 2D map or cost map of the surroundings. The cost map is then utilized by the exploration planner which analyses the map and uses exploration transform approach discussed in [4], to make decisions and plan a path for the robot. Finally, it computes the robot's linear and angular velocities required to follow the planned path and sends it to the base/exploration controller. The controller receives the robot's velocities and converts these velocities into individual wheel velocity commands.[5]

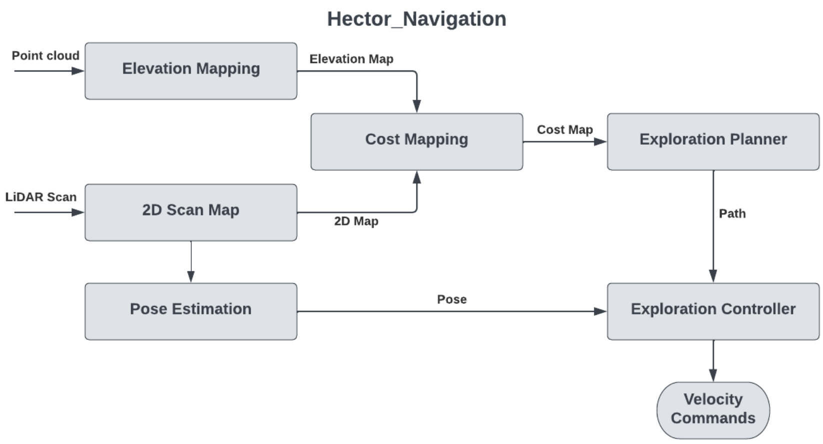


Figure 4. Robot Navigation Architecture

The kinematics of differential drive robot defines how the individual wheel velocities are calculated. Let the robot's linear velocity be V and angular velocity be ω. Then the velocities of left and right wheel, v_l and v_r are given by,

$$v_l = \frac{2v-wL}{2R} \quad (4)$$

$$v_r = \frac{2v+wL}{2R} \quad (5)$$

where L and R are the wheel separation length and wheel radius respectively.

3. Hardware Development

The whole system is run on 8GB Raspberry Pi 4 Model B microprocessor. The features and specifications of this board allows the smooth working of all the operations with efficiency. The LiDAR module used is RPLidar with a range of 6m radius. The drive mechanism used is a simple 4-by-4 differential wheel drive. Fig.4 shows all the hardware used for building the project.

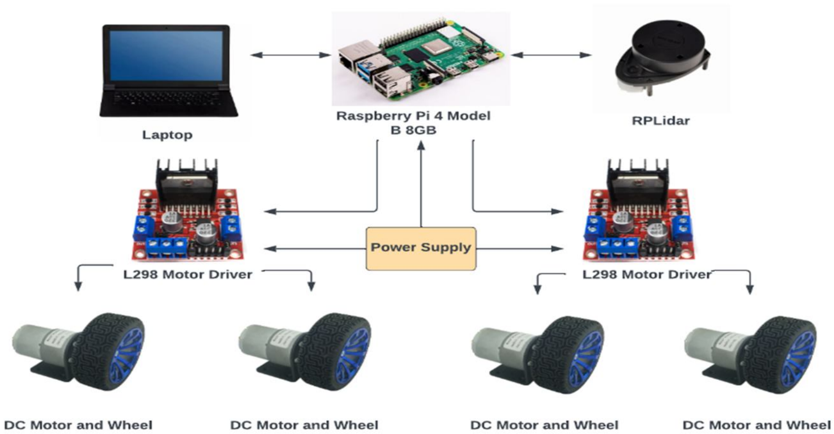
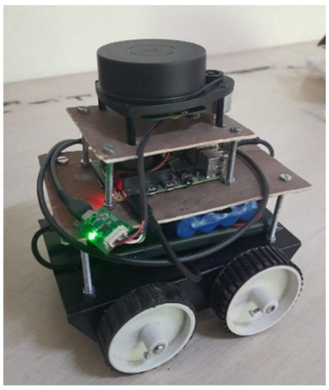


Figure 5. Hardware Components

III. Experimental Setup and Results



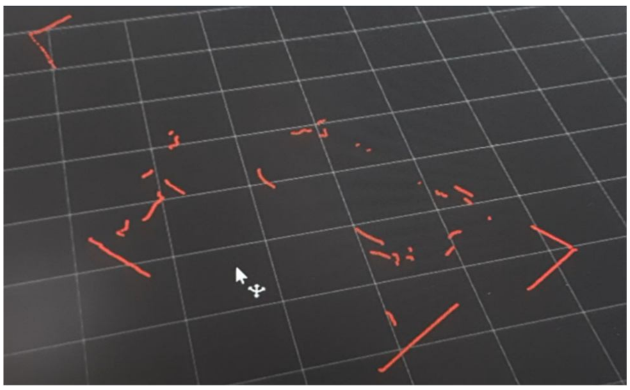


Figure 6. Robot Setup

Figure 7. Laser Scan

The project robot setup is demonstrated in fig. 6. The robot is self-designed and consists of an additional ultrasonic sensor for more efficient evaluation. The laser scan generated by lidar in a crowded room is shown in fig. 7. Generally Hector SLAM does not require an odometry device for localization but one can be used to generate more accurate maps. Hector SLAM is implemented after using the laser scan. The robot is tested in a small self-built area. The arena is shown in fig. 8 while the results on RViz i.e. map generated in the arena with and without the obstacles is demonstrated in fig. 9 and fig. 10.

Figure 8. Arena

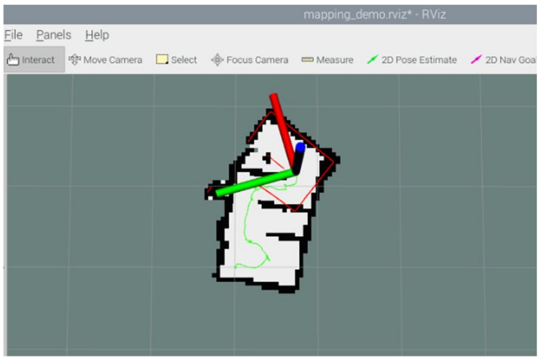


Figure 9. With Obstacles

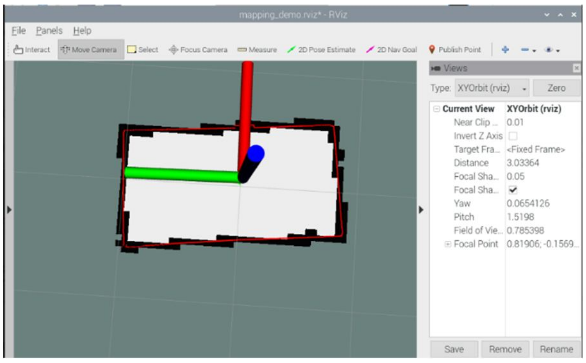


Figure 10. Without Obstacles

The map is demonstrated in RViz. The terminals green, red and blue refer to the front, right and top direction of the robot respectively. The curved green line on the map gives the path that the robot has taken from the start. On getting a closer look, red lines can be seen along the edges of the arena. These lines are the laser scan generated by the Lidar.

From the images, one can easily identify the arena. The arena used for testing is small (nearly 8x4 feet) and since the range of the lidar is 12m, the map is generated quickly without the obstacle. Hence, the robot did not move from it's original position.

IV. Conclusion And Future Work

A. Conclusion

While different GPS-based maps enable autonomous robots and vehicles for precise localization when they are outside, they cannot do the same when they are inside. Without an accurate mapping of rooms and buildings that is updated in real-time, robots that are responsible for tasks like moving packages in a warehouse or cleaning floors cannot do their jobs effectively.

This technology addresses that issue by coming up with a low-cost method for making indoor environment maps in real-time. This enables a robot to move without running into anything and to navigate by dodging obstacles. Additionally, this enhances performance and cuts down on navigation time.

B. Future Work

Path optimization algorithms play a crucial role in finding optimal paths in various domains, including robotics, logistics, transportation, and more. These algorithms aim to determine the most efficient or optimal path between a given start and goal location, considering factors such as distance, time, cost, or other specified objectives.

Path optimization algorithms refer to a class of computational algorithms used to find the best possible path or route between two or more points in a given environment, subject to certain constraints or objectives. The goal of path optimization algorithms is to minimize or maximize a given objective function, such as distance, time, cost, or energy, while taking into account the constraints of the environment, such as obstacles, terrain, or traffic. Path optimization algorithms can be used in various applications, including indoor and outdoor navigation, logistics and transportation, and robotics and automation. These algorithms can be based on a variety of mathematical techniques, including graph theory, optimization theory, and machine learning, and can be implemented using various programming languages and platforms.

REFERENCES

[1] P. Vanicek*, L. Beran, ``Navigation of Robotics Platform in Unknown Spaces using LiDAR, Raspberry Pi and Hector SLAM'', J Fundam Appl Sci. 2018, 10(3S), 494-506, February 2018.

[2] Unknown, "How to set up hector_slam for your robot"[Online].

[3] Xuexi Zhang , Jiajun Lai, Dongliang Xu, Huaijun Li , and Minyue Fu, ``2D Lidar-Based SLAM and Path Planning for Indoor Rescue Using Mobile Robots'', Journal of Advanced Transportation, Volume 2020, Article ID 8867937, November 2020.

[4] Stephan Wirth, Johannes Pellenz, ``Exploration Transform: A stable exploring algorithm for robots in rescue environments'', unpublished.

[5] Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen,Uwe Klingauf, Oskar von Stryk, ``Hector Open Source Modules for AutonomousMapping and Navigation with Rescue Robots'', Robot Soccer World Cup, DOI:10.1007/978-3-662-44468-9_58, January 2014.