Paper ID: 351

Autonomous Navigation and Localization with AMCL using Turtlebot4 and ROS2

Imnul Haque Ruman Talukder[1], Avishak Talukdar[2], S.M.Fahim Faisal[3], Mumit Hassan[4] & Shishir Chandra Das[5]
[1,2,3,4,5] Department of Mechatronics and Industrial Engineering,
Chittagong University of Engineering and Technology, Chattogram-4349, Bangladesh
E-mail: u1809014@student.cuet.ac.bd, u1809004@student.cuet.ac.bd, fahimfaisal@cuet.ac.bd, mumit0962@gmail.com, u1809019@student.cuet.ac.bd

Abstract

This paper aims to perform autonomous navigation and object localization in unstructured, unknown environments. The SLAM algorithm is a widely used approach for building a map of an environment and estimating the robot's position within it, which is especially useful in dynamic and unstructured environments. This research implements various SLAM algorithms, including the Extended Kalman Filter and GraphSLAM algorithms. ROS2 SLAM Toolbox is used for robot path planning and movement. The AMCL algorithm is used for robot localization and tracking the position of the robot in the known environment of the SLAM map. This research also explores different path-planning algorithms like Dijkstra's algorithm, A, D*, and the Dynamic Window Approach to plan collision-free paths in 2D navigation. It also uses the Nav2 plugin of the ROS2 framework to determine optimal path planning and navigate the robot from one place to another with appropriate speed. This research approach has been implemented and validated on the Clearpath Turtlebot4, demonstrating large-scale autonomous navigation to facilitate optimal, safe, and effective mobile robot navigation and localization in various unknown environments.*

Keywords: Robot Operating System (ROS), Adaptive Monte Carlo Localization (AMCL), SLAM Toolbox, Nav2, Simultaneous Localization and Mapping (SLAM).

1. Introduction

Autonomous robots are becoming increasingly incorporated into outdoor activities as new robotic solutions are created. Navigation technologies enable mobile and flying robots to safely explore their environment. To accomplish mobile robot autonomous navigation in an unfamiliar area, the robot must have a trustworthy map of the surroundings as well as its location. One of the primary tasks of an autonomous robot is to reach its destination in an unfamiliar area while avoiding obstacles.

Localization, mapping, and path-planning enable robots to recognize their surroundings and predict their location correctly. The Simultaneous Localization and Mapping (SLAM) technique is widely used for robot localization and mapping in unfamiliar situations [1]. For decades, robotics research has focused on SLAM-based robot localization and navigation algorithms, with several methodologies and approaches suggested and tested in different contexts. SLAM algorithms generally include two components: mapping and localization. The mapping component creates a map of the surroundings using sensor data from laser range finders, cameras, or sonar sensors. The mapping component creates a map of the surroundings using sensor data from laser range finders, cameras, or sonar sensors. Because of the noisy environment and imperfections in sensor map development, localization procedures include predicting the robot's location using AMCL and particle filter techniques.

Numerous applications, including autonomous driving, interior navigation, and exploratory missions, have seen success with the SLAM algorithm. Autonomous cars, for instance, employ the SLAM algorithm to map their surroundings and estimate their location and orientation. This enables the vehicle to travel safely and arrive at its destination. Similar to this, SLAM-based algorithms have been applied to interior settings for robot navigation, allowing the robot to navigate through dynamic, unstructured spaces while avoiding obstacles and accomplishing its objective.

In this work, the SLAM toolbox package will be used to create a map of an unknown area. A terrain map for a more sophisticated autonomous driving system is made using it. Talk about other SLAM methods as well, such as the GraphSLAM algorithms and the extended Kalman filter (EKF), and how they are used to estimate a robot's

location and orientation probabilistically. This research also investigates localization techniques like AMCL, which estimates a robot's location and orientation in real-time using a particle filter.
In order to move from the starting point to the destination while avoiding collisions with obstacles, the research robot uses the nav2 stack for robot navigation. It also explores various path-planning techniques that can be used with the map created by the SLAM algorithm to generate collision-free paths for the robot to navigate toward its goal. Robot navigation and localization are challenging problems, and one common robotics approach to solve them is simultaneous localization and mapping, or SLAM. By using this method, a robot may map its surroundings and locate itself inside them at the same time without depending on outside sensors or previous environmental information [2]. For the SLAM approach to construct an environment map and update robot location simultaneously, it needs a laser scanner and an odometry system.
In this study, TurtleBot 4 creates a 3D Lidar SLAM map using the GraphSLAM method, which represents the surroundings and the robots' posture estimations on a graph. The main goals of current SLAM research are to enhance data linking, accelerate processing, and identify data loops [3]. Despite delayed investigation, significant progress was achieved in mapping, location, and circulation. Some experiments integrate sonar and optical information in SLAM to improve the accuracy of locating and mapping [4]. The ROS2 Nav2 Plugin is used to execute robot navigation and establish the best route planning for the robot's goals. Navigation may be reduced to three major algorithms: costmap, planner, and controller. Planner uses Dijkstra's algorithm, A*, D*, and DWA algorithms to find the robot's best route in a given map. Path planning, positioning, and environmental perception are necessary for autonomous mapping and navigation [5]. The robustness of navigation was shown by the indoor mobile robot platform, which used the Gmapping algorithm to create indoor maps based on a Raspberry Pi [6]. The Gmapping technique was used for creating maps, while the A* and DWA (Dynamic Windows Approach) algorithms were chosen for route planning [7]. Although the Dijkstra algorithm for route planning lessened reliance on the odometer, issues remained, including a sluggish path search efficiency and high lidar frequency needs [8]. This work focuses on autonomous navigation in an unfamiliar interior environment. Turtlebot4, a mobile robot operating in a dynamic environment, is used to construct a SLAM map in order to carry out this navigation. Robots may make intelligent decisions about where to travel based on SLAM maps by utilizing the ROS navigation stack Nav2, which is used for optimum route finding using AMCL localization and path-planning techniques such as Dijkstra's algorithm, A*, D*, and the Dynamic Window Approach.

2. Methodology and Navigation Approach

The aim of this chapter is to provide a comprehensive overview of the whole process involved in autonomous navigation in an unfamiliar area. The research employed the NAV2 stack, which featured Nav2 aim and two-dimensional posture, for autonomous navigation. Robot location precision is achieved in autonomous navigation systems by the use of methods such as odometry and three-dimensional odometry. The whole process will be carried out with the aid of Rviz, Nav2 stack, Localization, Path-Planning, Path-tracking, and a SLAM toolbox.

2.1 ROS2 Architecture

The Robot Operating System (ROS) has been modified to become ROS2, which is a set of software libraries and tools to help create robot applications. It provides a framework for creating robot software and includes tools, drivers, and algorithms for developers [9]. ROS2 may interact with one another utilizing the Node, Topics, and Services ways.
Node: A node is a basic entity in ROS that is in charge of controlling devices or processes. Nodes exchange information via topics or services, and packages which often include one or more nodes to carry out certain tasks distribute ROS software.
Topic: Data streams for sharing information across nodes, including motor speed or sensor data, are called topics. With a distinct name and message type for every subject, nodes may publish or subscribe to them. A single node is not able to post or subscribe to a topic at the same time as other nodes.
Service: Services facilitate two-way communication between nodes by enabling a server node to offer a service and other nodes to request it and get a response.

2.2 Odometry

Odometry is a technique used in robotics to determine a robot's position from its first frame of motion. It entails determining the robot's present position by calculating rotation and linear distance using high-rate sensor data input. By using odometry as a means of localization, a robot may monitor its position while it navigates its surroundings. It is often used with localization strategies, such as visual-based approaches that assist the robot in determining its stance by using elements of its surroundings.

2.3 SLAM Mapping

The well-known ROS module SLAM toolbox is used in ROS2 to carry out simultaneous localization and mapping. It continuously records a representation of the ambient sensors and the robot's posture, or position and orientation. The robot's surroundings are represented by a map that is created using its posture and sensor data. Advanced features of the SLAM toolbox include the use of the ROS2 transform library (tf2), which allows a robot's posture to be transformed over time in relation to different "frames" of reference (positions and orientations) to create a representation. One such technique, called GMapping, enables SLAM to be performed utilizing a local orthogonal source and laser range data (LiDAR scans).

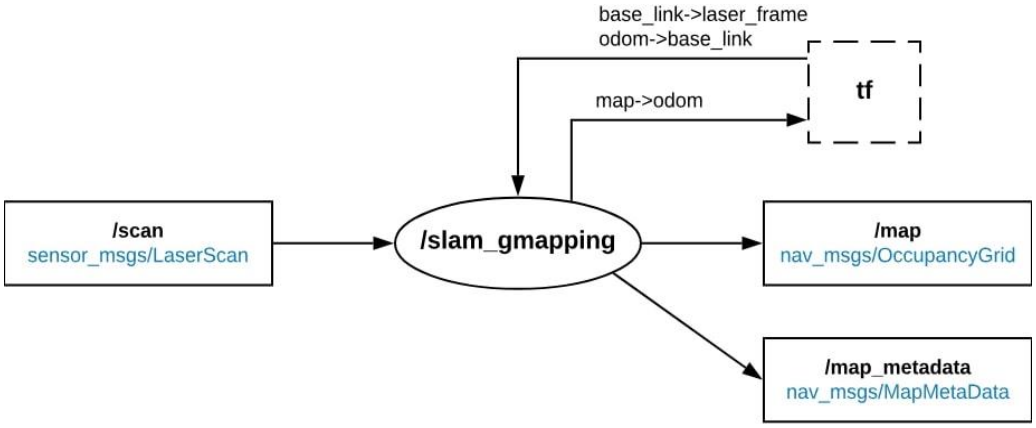


Fig. 1. SLAM topics and node

Using a laser scanner with an odometry system, a speed measuring device based on encoder data is one of the easiest ways to accomplish an accurate and exact SLAM. To prevent undesired changes, these algorithms use a variety of techniques, including the Monte Carlo approach. The AMCL algorithm updates the robot's potential posture distribution based on data from lidar or laser rangefinder sensors. To do this, a collection of particles from the distribution is sampled, each of which represents a potential robot stance. To increase the posture estimate's accuracy and update rate, the AMCL algorithm further takes the robot's odometry into account.

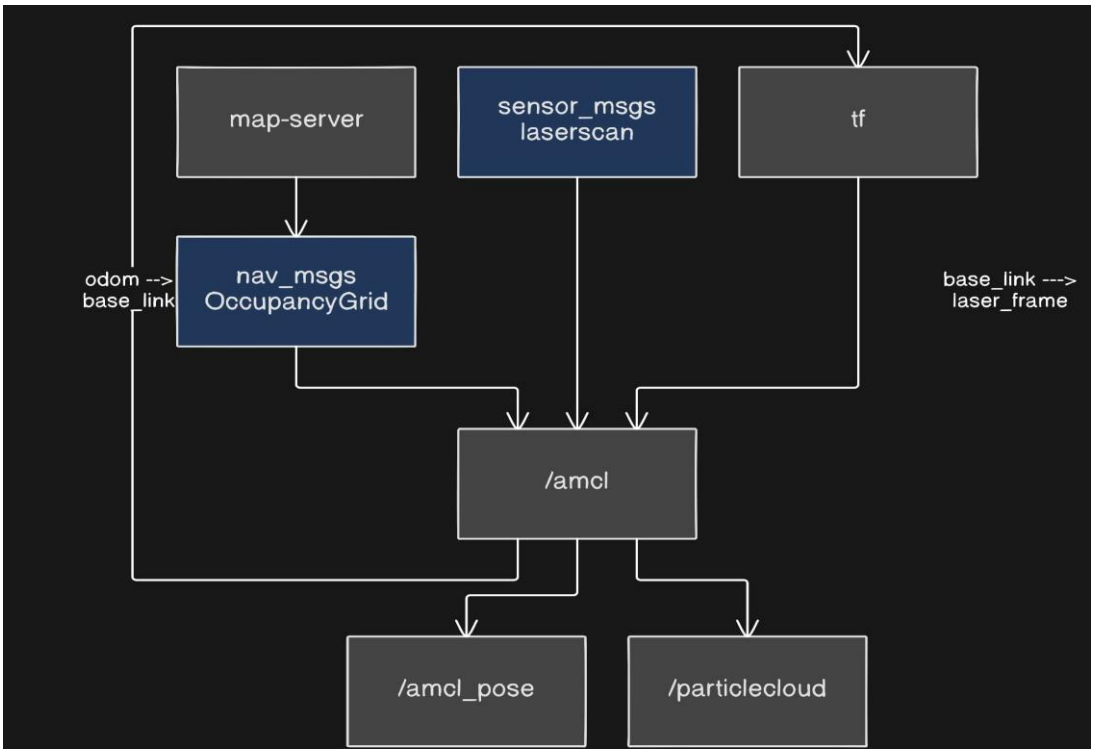


Fig. 2. Adaptive Monte Carlo Localization

2.4 Nav2 stack and Path-planning

Getting Around Robots can navigate the whole SLAM map securely thanks to the Nav2 module. Nav2 Path Planner's algorithm determines the best route to reach the goal. The planner employs a costmap to ensure your safe and efficient arrival at your location. It favors low-cost locations like wide spaces and avoids high-cost places like barriers. Finding a route that enables the robot to arrive at its destination safely and efficiently while avoiding obstacles and adhering to any mobility restrictions is the overall aim of path planning. Dijkstra's method uses A* and D* to design the course, choosing the one that minimizes the robot's destination.

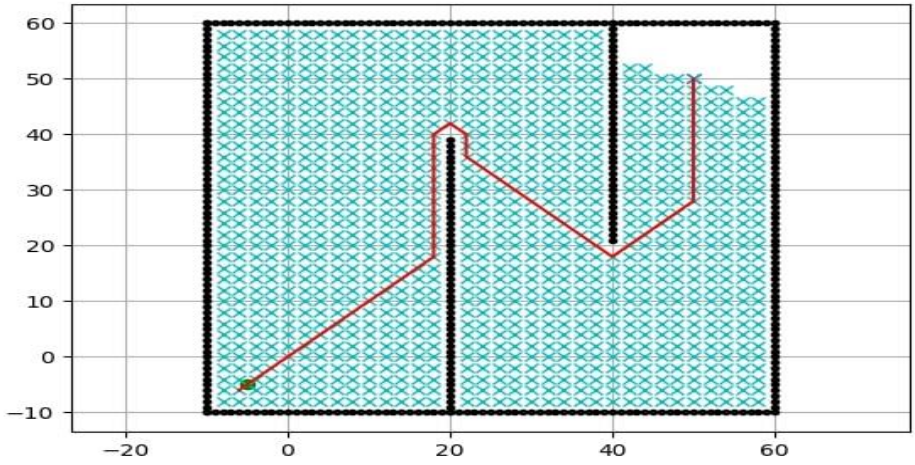


Fig. 3. Dijkstra Algorithm

3. Results

This study used Physical Turtlebot 4 to generate a SLAM map of the dynamic interior environment. Using Rviz software's Nav2 tool, specify a target for the robot, and the Nav2 stack will plot a route to the goal position and try to drive it.

3.1 Generating SLAM map with Turtlebot4

Start the SLAM and Rviz2 software to create a simultaneous localization and mapping (SLAM) system for an unknown indoor environment. This project uses a physical TurtleBot4 to generate an indoor SLAM map, leveraging a 2D Lidar scanner to create floor layouts for navigation.

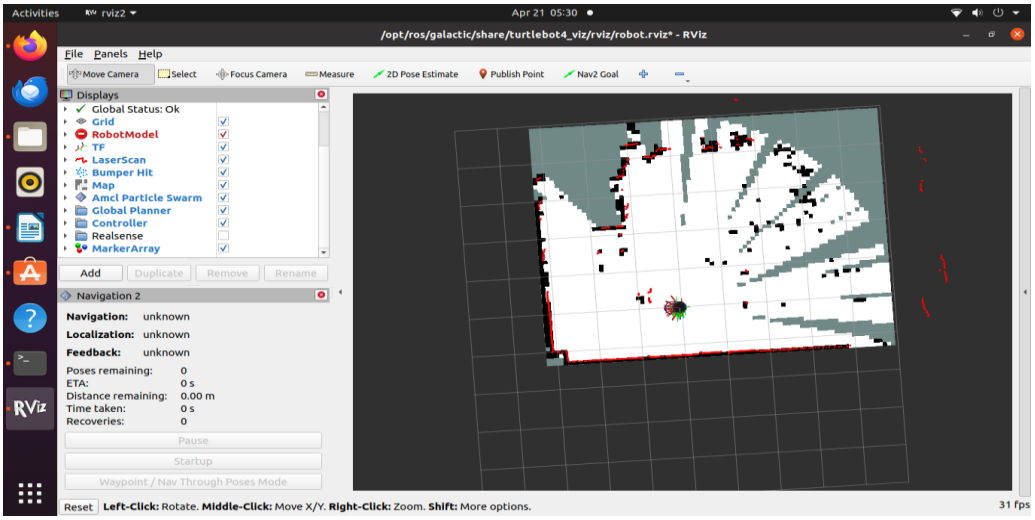


Fig. 4. Rviz and SLAM

3.2. Localization and Navigation

For autonomous navigation, TurtleBot4 interacts with the navigation stack nav2, which uses the robot to go from the starting location to the destination. The 2D posture estimate tool is used in localization to put the robot's estimated beginning posture on the map, while AMCL determines the robot's location and orientation in its surroundings. Additionally, a particle filter is used to visualize the probability distribution of the robot's pose. Particles from this distribution are used to represent possible robot posture hypotheses**.** Costmap determines the robot's safe route and helps the robot learn which places are simple to walk and which are more challenging.

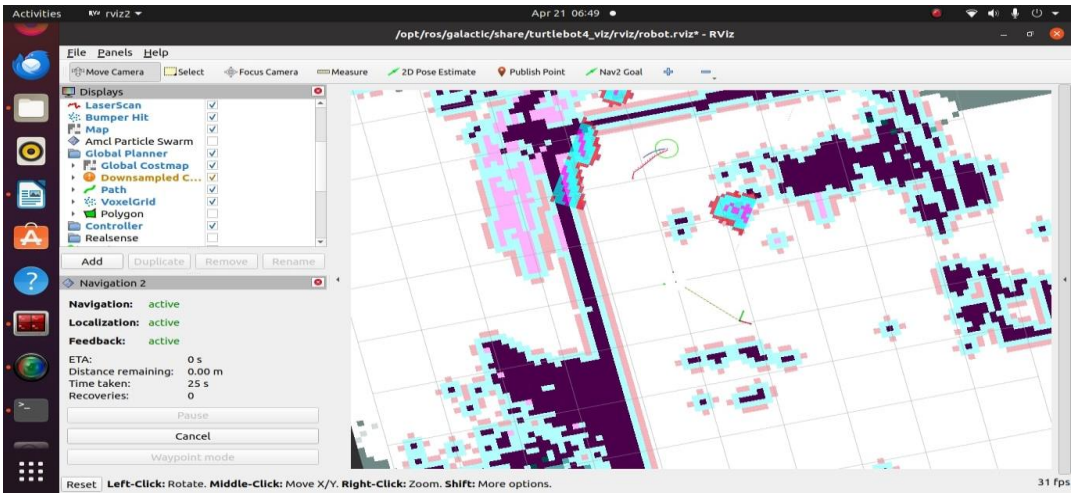


Fig. 5. Nav2 goal locate robot in desire position

Nav2 global planner algorithm try to find safe path for robot using costmap. It avoids high-cost areas such as obstacles and prefers low-cost areas such as open spaces. In the global cost map, robots use the produced SLAM map and the most recent AMCL localization routes. Nav2 Goal tool gives the goal pose of the robot and its plan collision-free route for the robot and updates a real-time global cost map of surroundings. In figure 5,the blue colors on the map show where the obstacles are, the low pink colors show the free way between the obstacles, and AMCL draws a pink dot line in figure 6 to help the robot figure out how to get to its destination.

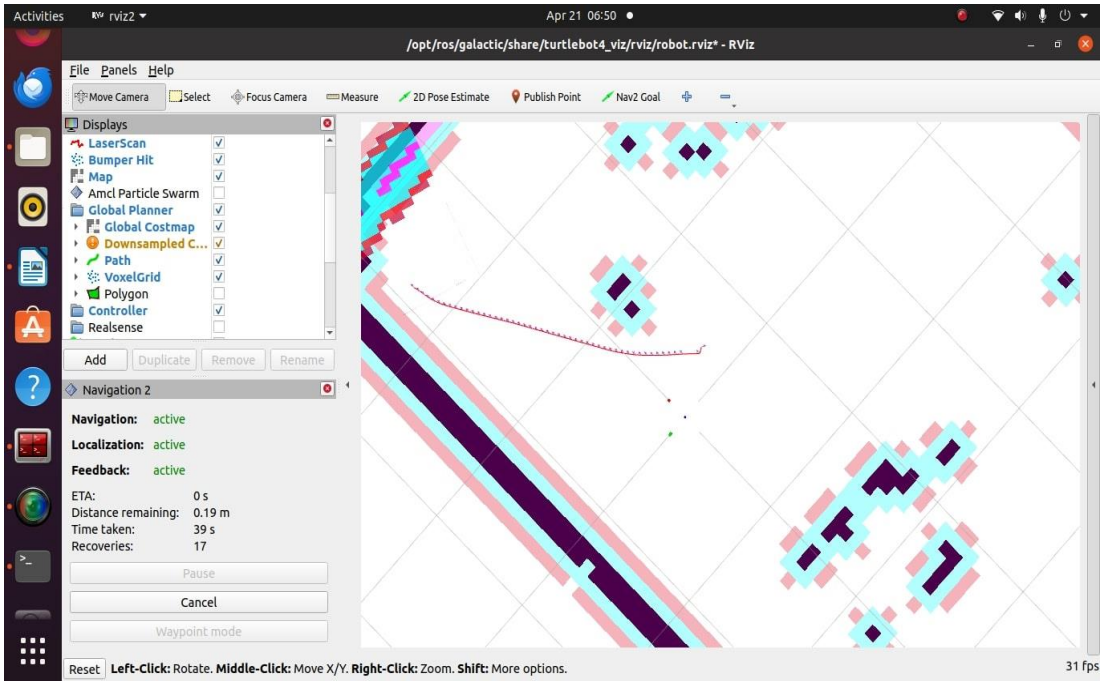


Fig. 6. AMCL path-planning route

4. Conclusion

This study proposes employing the TurtleBot4 for autonomous navigation in unfamiliar interior environments. All of the robot's systems were implemented using the ROS2 framework. SLAM-generated maps were utilized to interact with the user and plan the journey. This showed significant promise for future improvement and has the ability to function in mapless environments such as underground mines, tunnels, or buildings. The most important systems, such as localization, navigation, and Nav2 interaction with the turtlebot, were tested to ensure correctness. Autonomous navigation aims to reduce time, prevent human intervention in hazardous areas, and traverse complicated patterns with greater precision. The robot successfully completed tasks by localizing itself in the surroundings, dynamically replanning its course to avoid obstacles, and following that path to achieve its objectives.

5. References

[1] Wu H, Chen Y, Qin H (2021) MPC based trajectory tracking for an automonous deep-sea tracked mining vehicle. ASP Trans Internet Things 1(2):1–13

[2] Qiao, Junfu, Jinqin Guo, and Yongwei Li. "Simultaneous localization and mapping (SLAM)-based robot localization and navigation algorithm." Applied Water Science 14, no. 7 (2024): 1-8.

[3] Li, Jianlong, Zhijun Li, Ying Feng, Yiliang Liu, and Guangming Shi. "Development of a human–robot hybrid intelligent system based on brain teleoperation and deep learning SLAM." IEEE Transactions on Automation Science and Engineering 16, no. 4 (2019): 1664-1674.

[4] Dai, Yue. "Research on robot positioning and navigation algorithm based on SLAM." Wireless Communications and Mobile Computing 2022, no. 1 (2022): 3340529.

[5] Maosheng, L., Di, W., and Hao, Y. "Deploy Indoor 2D Laser SLAM on a Raspberry Pi-Based Mobile Robot." In Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 7-10. Hangzhou, China: IEEE, 2019

[6] Peng, W. Research on SLAM Technology and Path Planning Algorithms for Unmanned Vehicle. Master's Thesis, Harbin Institute of Technology, Harbin, China, 2021

[7] Maosheng, L.; Di, W.; Hao, Y. Deploy Indoor 2D Laser SLAM on a Raspberry Pi-Based Mobile Robot. In Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 24–25 August 2019; pp. 7–10

[8] Jiaxin, S.; Zhiming, Z.; Yongqing, S.; Zheng, Z. Design and Implementation of Indoor Positioning and Navigation System of Mobile Robot Based on ROS and Lidar. *Mach. Electron.* 2018, *36*, 76–80

[9] [Online]. "ROS Documentation," ROS.org, Available: https://docs.ros.org/en/rolling/index.html.