Roadmap for Navigation with Robot Operating System (ROS)

Suraj Koundinya P[1], Dr. K. R. Prakash[2], Ananya S[3]
[1,2,3]*Dept. of Mechanical Engineering, The National Institute of Engineering,* Mysuru, India

Abstract—ROS (Robot Operating System) (Robot Operating System) A robust toolkit for path planning, simultaneous localization, and mapping is called Navigation Stack (SLAM). The goal is to provide a quick inside look at the navigation stack and show a straightforward procedure to implement SLAM in any robot. Focusing primarily on the requirements that will enable readers to set up perception sensors, tune navigation controllers, set odometry, reference frames, and their transformations, and plan the path for their own virtual or real robots.

Keywords— ROS, Localization, Navigation, Mapping.

I. INTRODUCTION

The crucial tool of simultaneous localization and mapping (SLAM) enables the robot to recognise the obstacles in its immediate environment and devise a way to avoid them [1]. This technique combines multiple strategies to enable robots to navigate in unfamiliar or imperfectly known settings.

A SLAM package is available for ROS. This article seeks to present the theory behind ROS and clearly explain how to implement SLAM in any robot. It will also detail how to apply SLAM to virtual robots in a virtual environment. These robots are created to execute a differential wheeled mobile robot's planned navigation by publishing and subscribing the same information in real and virtual environments, where all real-world sensors and actuators are functioning.

For a robot to navigate through a medium, ROS offers a collection of tools that are useful. In other words, the robot can plan and follow a path while deviating from obstacles that pop up along the way along the course. On the navigation stack, you can find these resources [2]. The localization systems, which enable a robot to locate itself regardless of whether a static map is provided, or simultaneous localization and mapping is required, are one of the several resources required for accomplishing this task and that are included on the navigation stack. A tool called AMCL enables a robot to locate itself in a space using a static map that was previously made. The drawback of this resource is that because it uses a static map, any changes to the robot's surroundings will cause it to suffer [3] because each change would necessitate the creation of a new map, which would take time and processing resources. Since the robots should be able to function in settings like factories and classrooms, where there is constant mobility, being able to navigate only in situations without modifications is insufficient. Two more localization systems, gmapping and hector mapping, are provided by the navigation to get over the static maps' lack of flexibility.

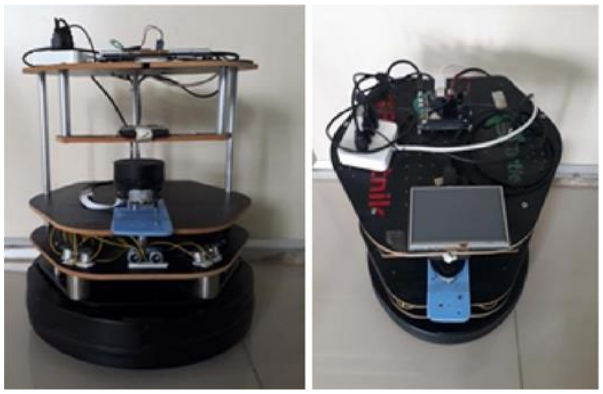
Fig. 1. Front and Top view of robots used to test the navigation.

Applying a speed command to regulate the robot locations in comparison to the trajectory taken and a Gazebosimulation environment, Turtlebot 2 attempted to accomplish our intended goal. Both gmapping and hector mapping are based on SLAM [4], a method that entailsmapping an area while a robot is moving. To put it another way, when a robot moves through an environment, it receives data from the surroundings with the help of its sensors and creates a map. By doing this, you may create a mobile base that can update an existing map as well ascreate a map of an uncharted environment [5], allowing you to utilize the device in a wider range of settings.

Gmapping and hector mapping are different in that the first one considers odometry data to generate, update the robot's pose, and the map but the robot needs encoders, which prevents some robots from utilizing it. Odometry datais intriguing because it can help create more accurate maps [6], and by comprehending the dynamics of the robot, we can determine its position.

Kinematics is another name for the robot's dynamic behavior. The way the components that ensure the robot's movement are put together has an impact on kinematics [7]. The type of wheel, the number of wheels, the placement of the wheels, and the angle are mechanical characteristics that affect kinematics. Although the odometry data might bequite helpful, it is not error-free. The defects are brought on by the lack of accuracy in the capitation, friction, slide, and drift, among other reasons. Over time, they may collect, producing inconsistent data and harming the generation of maps, which are prone to distortion in such situations. For the simultaneous localization and mapping in this work, a mobile robot is utilized, as shown in Fig. 1.

First, a description of mobile robot navigation will be given, followed by a discussion of simultaneous localization and mapping and a discussion of transformations.

The major goal of the second section is to elaborate on everything required to configure a real or virtual robot. A thorough description of odometry and kinematics is provided, with particular attention paid to issues like the precision of odometry and the consequences of its absence inthe creation of navigational cost maps. A robot needs to understand the surroundings from its sensor readings. There are different types of perception sensors, such as depth, lightdetection, and ranging.

The navigation stack is set up to function with as many robots as feasible while attempting to structure in a way that both achieves a high level of specificity and abstraction, granting a way to make robot perform SLAM are also discussed.

The distance values from the sensor are used to create map because they are important that they oversee detecting theoutside world and acting as a guide for the robot. The information gathered by the sensor, Before the gadget may use it, it must be altered. Since the sensors measure the environment in proportion to themselves, a geometric conversion is needed. The tool offered by ROS allows users to change the sensors in reference to modify the robot and adapt the measurements to the robot navigation to make it easier to convert.

II. METHODOLOGY

A. *What is the talk about?*

A robot must "understand" the surroundings using its sensor readings in order to perform navigation.

Understanding entails acquiring knowledge at the appropriate degree of abstraction. Fig. 2. An autonomously moving robot should be able to recognise its surroundings and its location.

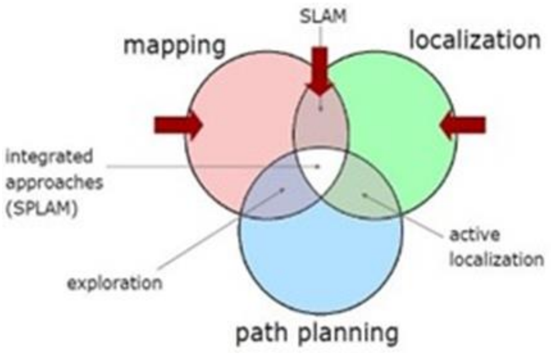

Fig. 2. Understanding the concept of SLAM

B. *Map*

- A map is a picture of the area in which the robot is working.
- It should have sufficient knowledge to conduct an interesting activity.
- The map server package must be loaded in order to use the map that your robot created.

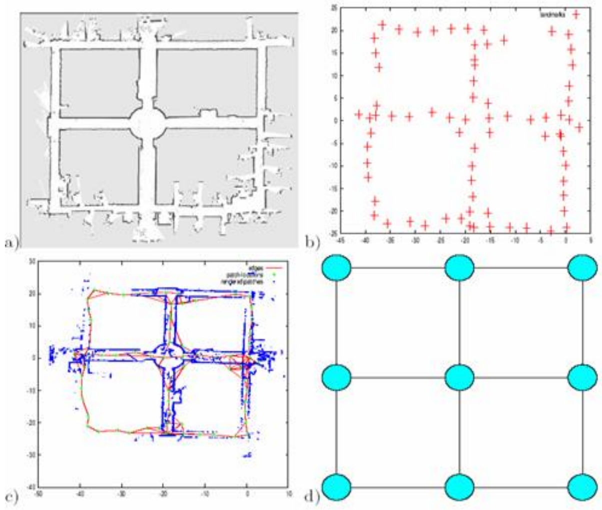

Fig. 3. Sample maps generated from the Turtlebot.

C. *Robot Pose and Path*

- A reference frame is specified by a metric map.
- A robot must be aware of its location within the map'sreference frame in order to function.
- A path is a set of waypoints or actions taken in orderto travel from one spot on the map to another.

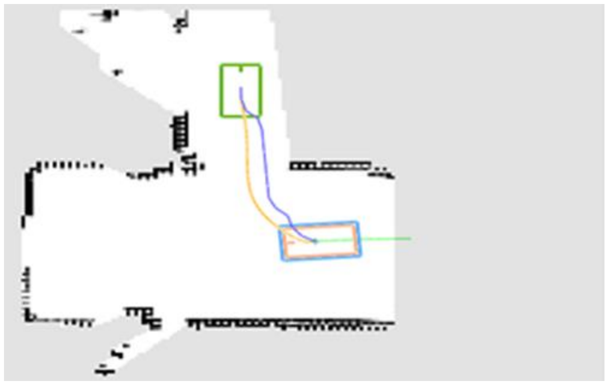
Fig. 4. Sample Robot Pose & Path generated from the Turtlebot.

D. *Localization*

- Identify the location of the robot at the moment, themeasurements made up to that point, and a map
- When the robot's address is unknown or when the pose is not known in advance, it uses a particle filter to manage the global localization concerns.

E. *Path Planning*

- Find a route to a specified goal point using a localized robot and a map of traversable regions if one exists.
- To keep the robot moving from the start state to the end state, a suitable trajectory is constructed as a series of action to maintain.
- The path to be considered appropriate, it must be both robot navigable and best in terms of at least one variable.

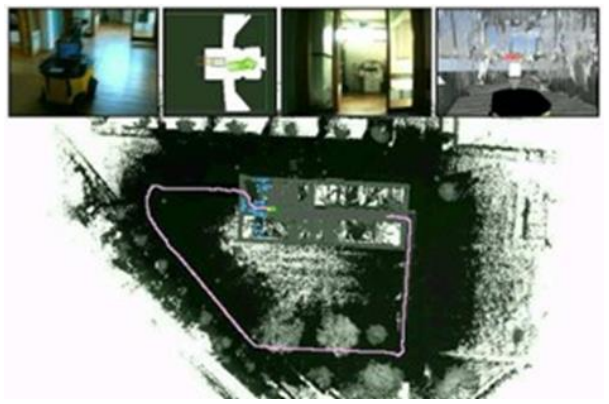
Fig. 5. Map generated when doing Path Planning.

F. *Mapping*

- Determine the map of the environment using a robot with a perfect ego-estimate of its position anda series of measurements.
- Typically, a precise estimation of the robot pose isnot available.
- Instead, we address Simultaneous Localization andMapping (SLAM) [11], a more challenging issue.

III. SLAM

SLAM (simultaneous localization and mapping) is a method of autonomously mapping using robots that allows for simultaneous map creation, robot localization. The robot can map out in an unknowing area thanks to SLAM algorithms. [12].

Estimate:

- The environment's map
- A moving object's route using a series of sensor measurements.
- A flexible mapping technique is Simulation Localization and Mapping (SLAM), which can be used for mapping.

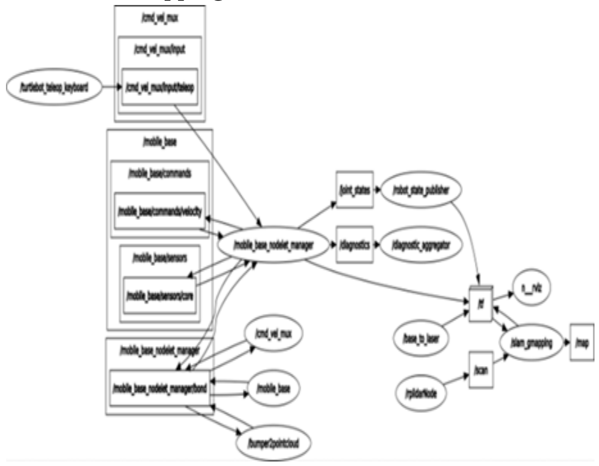

Fig. 6. RQT graph of Turtlebot navigation

IV. ROS NAVIGATION

Once the localization issue has been resolved, the motions necessary for the robot to assume the desired stance must becalculated. Choosing the best course of action and establishing the speed of implementation are the two halves of this question.

The Turtlebot fits the criteria for the ROS Navigation Stack, which is designed for 2D maps, square or circular robots with holonomic drives, and planar laser scanners. In order tooffer safe velocity commands, it makes use of odometry, sensor data, and a desired pose. seen in Fig. 7. The ROS Navigation Stack's "move base" node is where all the magic happens [13]. It controls communication across the navigation stack and carries out the navigation task using a global and local planner.

The robot's position is calculated using sensor data that is gathered, analyzed, and integrated (sensor sources node, sensor transformations node, sensor transformations node) [14]. (Odometry source node). To enable "move base" to compute the trajectory and communicate velocity commands, this data is made available through the base controller node.

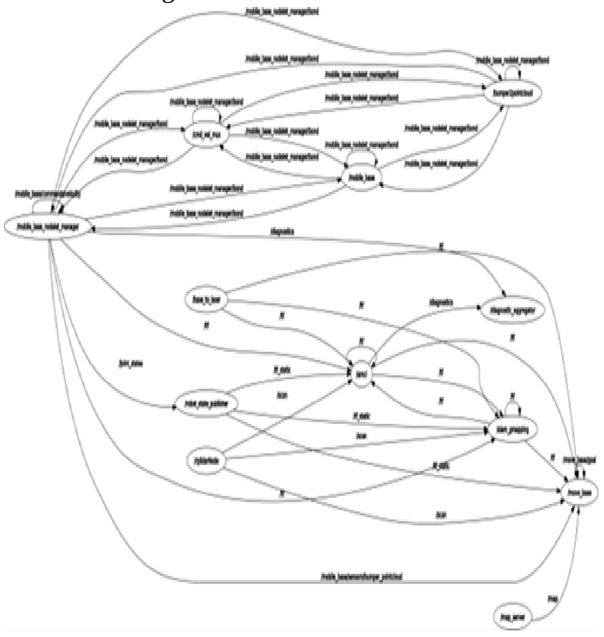

Fig. 7. ROS system structure generated by RQT graph

A. Building a Map

- There are countless SLAM algorithms available. Robot trajectories are tracked by ROS using Gmapping, which employs a particle filter.
- While driving the robot about in the environment it will operate in, record a bag using /Odom, /scan/, and /tf in order to create a map.
- The map is an occupancy map and is displayed as a picture of the environment's blueprint and a configuration file (yaml) that provides meta data about the environment.

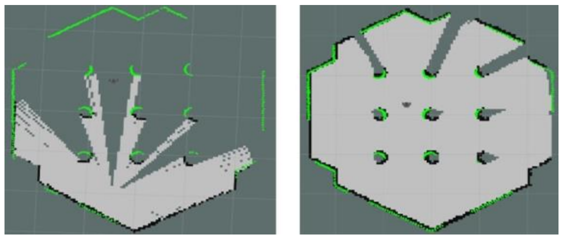

Fig. 8. Map generation using Turtlebot.

B. Localizing a Robot

- The Adaptive Monte Carlo Localization technique is implemented by ROS.
- The position of the robot is determined by AMCL using a particle filter.
- A particle is used to represent each posture. Particles move in accordance with the (relative) movement determined by odometry.
- ROS integrates the localization by generating a transform that "corrects" the odometry from a map-frame to the Odom frame.
- Ask the transform of base footprint in the map frame to get the robot's position based on localization.

V. ROS ENVIRONMENT

The implementation of these techniques was conducted with the help of the ROS development environment. There is a development environment that enables robotic applications with a variety of libraries and tools. Hardware,peripheral drivers, data viewers, messaging, and packagemanagement are all in one place. The architecture allows forthe possibility of a software program with the possibility to publishing and subscribing to different topics, a class of data bus used to transport information via messages with preset data formats.

VI. CONCLUSION

Common tasks were carried out with the help of a navigation system. The system maintains an accurate estimate of the robot's pose when no information is given prior. This map is updated when the pose estimation is high enough. The ambient lighting could cause false detections that could endanger the location and themap of the surroundings. The robot can find its way back and forth because of the continuous motion along with the global and Kalman filters.

Investigating new ways to match camera images with a depiction of the lights, fixture, and other celling components to speed localization might be fascinating. Significant movement without the identification of landmarks can cause the Kalman filter and mapping to operate with more accuracy. False alarms brought on by reflections, foreign materials, or unusual lighting circumstances are anticipated to be more resistant to it. The system responds to unexpected barriers like people or empty areas.

VII. FUTURE SCOPE

Finally, one employed the virtual forces technique of navigation to avoid these impediments, resuming the path whenever it was possible. The produced work might be used as a foundation for specialized tasks, like autonomousmapping and navigation, as a follow-up to the current effort.

REFERENCE

[1] L. Zhi and M. Xuesong. “Navigation and Control System of Mobile Robot Based on ROS,” *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018, pp. 368-372,* doi: 10.1109/IAEAC.2018.8577901

[2] Y. Nitta, S. Tamura, H. Yugen and H. Takase. “ZytleBot: FPGA Integrated Development Platform for ROS Based Autonomous Mobile Robot,” *2019 International Conference on Field-Programmable Technology (ICFPT), 2019, pp. 445-448*, doi: 10.1109/ICFPT47387.2019.00089.

[3] Doris M. Turnage, “SIMULATION RESULTS FOR LOCALIZATION AND MAPPING ALGORITHMS,” in 2016 Winter Simulation Conference, 2016.

[4] Dissanayake, M. W., Newman, P., Clark, S., Whyte, H., & Csorba, M. (2001). A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on robotics andautomation, Vol. 17, NO. 3, June 2001.*

[5] J. Borenstein, H. Everett, L. Feng and D. Wehe, 'Mobile robot positioning: *Sensors and techniques', J. Robotic Syst., vol. 14, no. 4, pp. 231-249,1997*.

[6] H. Chang and T. Jin, “Adaptive tracking controller based on the pid for mobile robot path tracking,” in *Intelligent Robotics and Applications,Springer Berlin Heidelberg, 2013,* pp. 540–549.

[7] J. Leonard and H. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Transactions on Robotics and Automation,* vol.7, no. 3, pp. 376–382, 1991.

[8] Sumanarathna, D.M.G.A.I.; Senevirathna, I.A.S.R.; Sirisena, K.L.U.; Sandamali, H.G.N.; Pillai, M.B.; Abeykoon, A.M.H.S., "Simulation of mobile robot navigation with sensor fusion on an uneven path," in *Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on , vol., no., pp.388-393*, 20-21 March 2014.

[9] P. Avdullahu, “Turtlebot control, mapping and localization via ROS”,*Master Thesis, University of Prishtina, Kosovo, 2014.*

[10] Jeong Woo; Young-Joong Kim; Jeong-on Lee; Myo-Taeg Lim, "Localization of Mobile Robot using Particle Filter," in *SICE-ICASE, 2006. International Joint Conference , Oct. 2006.*

[11] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I, *IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110*, June 2006, doi: 10.1109/MRA.2006.1638022.

[12] Sumanarathna, D.M.G.A.I.; Senevirathna, I.A.S.R.; Sirisena, K.L.U.; Sandamali, H.G.N.; Pillai, M.B.; Abeykoon, A.M.H.S., "Simulation of mobile robot navigation with sensor fusion on an uneven path," in *Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on , 20-21 March 2014.*

[13] Zingaretti, P.; Frontoni, E., "Vision and sonar sensor fusion for mobile robot localization in aliased environments," in Mechatronic and Embedded Systems and Applications, Proceedings of the 2nd IEEE/ASME International Conference on , vol., no., pp.1-6, Aug.2006.

[14] Kyriakopoulos KJ, Saridis GN (1988) Minimum jerk path generation. In: Proceedings of the 1988 IEEE international conference on robotics and automation, pp 364–369.

[15] Lombai F, Szederkenyi G (2008) Trajectory tracking control of a 6-degree-of-freedom robot arm using nonlinear optimization. In: Proceedings

of the 10th IEEE international workshop on advanced motion control, pp 655–660.

[16] Lombai F, Szederkenyi G (2009) Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator. In: Proceedings of the 2009 IEEE international conference on mechatronics, pp 1–6.

[17] Ge SS, Cui YJ (2000) New potential functions for mobile robot path planning. IEEE Trans Robot Autom 16(5):615–620.

[18] Fiorini P, Shiller Z (1996) Time optimal trajectory planning in dynamic environments. In: Proceedings of the 1996 IEEE international conference on robotics and automation, pp 1553–1558.

[19] Caselli S, Reggiani M, Rocchi R (2001) Heuristic methods for randomized path planning in potential fields. In: Proceedings of the 2001 IEEE international symposium on computational intelligence in robotics and automation, pp 426–431.

[20] Amato NM, Wu Y (1996) A randomized roadmap method for path and manipulation planning. In: Proceedings of the 1996 IEEE international conference on robotics and automation, pp 113–120.

[21] Kunchev V, Jain L, Ivancevic V, Finn A (2006) Path planning and obstacle avoidance for autonomous mobile robots: a review. Knowledge-based intelligent information and engineering systems. Springer, Berlin, pp 537–544.

[22] Khatib O (1985) Real-time obstacle avoidance for manipulators and mobile robots. In: Pro ceedings of the 1985 IEEE international conference on robotics and automation, pp 500.