ROS based SLAM implementation for Autonomous navigation using Turtlebot

Sumegh Pramod Thale[1], Mihir Mangesh Prabhu[2], Pranjali Vinod Thakur[3], Pratik kadam[4]
Dept.of Electronics Engg,
Ramrao Adik Institute of Technology, Navi Mumbai, India
[1]*thalesumegh@gmail.com*

***Abstract*—This paper presents the autonomous navigation of a robot using SLAM algorithm.The proposed work uses Robot Operating system as a framework.The robot is simulated in gazebo and Rviz used for data visualization.Gmapping package is used for mapping by utilizing laser and odometry data from various sensors.The Turtlebot provides open source software to perform navigation.**

***Index Terms*—Robot Operating System (ROS),Simultaneous localization and mapping (SLAM), Adaptive Monte Carlo Localization (AMCL) Kinect, Rviz, Gazebo, Gmapping, Turtlebot, URDF, Odometry**

I. Introduction

Autonomous navigation is a research field that has seen incredible advances for the past few years.In this well researched domain of autonomous navigation our main area of interest is indoor office like environment. In industrial applications the use of autonomous robots is increasing, the main cost associated with manual material handling is labour.The plant's material handling cost can be reduced up to 30 per cent by the use of autonomous robotic vehicles.These autonomous robots can navigate in locations dangerous to workers with minimum human intervention.Various issues in autonomous navigation like mapping,localization and path planning can be solved using different approaches.The purpose of this paper is to discuss and demonstrate the concept of designing and simulating a mobile robot capable of visually detecting and avoiding static obstacles using SLAM and reaching the desired location autonomously.

A. Related work

In the research paper [1] the authors have discussed various methods to implement SLAM through the use of Extended Kalman filter and Rao-Blackwellized Filter.The authors of paper [1]have incorporated autonomous navigation for moving obstacles and outdoor vision based navigation.Various SLAM algorithms like CoreSLAM, Gmapping, and HectorSLAM have been evaluated in the paper [2].The authors of the paper [3] made use of a kinect sensor and rviz to implement autonomous navigation on turtlebot 2.The utilization of compact RF ID sensor for the object level mapping of indoor environment has has been briefly explained by the authors of [4].

B. Solution

For implementing autonomous navigation SLAM algorithm can be used. SLAM is used for estimating the position of the robot by moving it along the unknown area. More the number of attempts of exploring the unknown area, the more is the quality of the generated map. SLAM algorithm is implemented using GMapping Tool. GMapping Tool requires odometry data (encoder data from the wheels) and laser data (kinect data) . slam_gmapping is used to create 2-D occupancy grid map which is like a building floor plan.

The robot software development is done in ROS (Robot Operating System). ROS provides services like message passing between processes and package,low level device control, implementation of common functions. In ROS the process are shown in graph format and the processing happens in nodes.

All the complex 3D simulations are done in GAZEBO. Models, GZ Sever, GZ Client and GZ Web are the main components of it. GAZEBO has features like built in robot models, cloud simulation,dynamics simulation, advanced 3D graphics, command line tools and TCP IP transport. With the help of GAZEBO we can conceptualize Inertia, Forces and Sensor Information.

C. Objectives

- Manually mapping the environment.
- To locate the robot in the generated map.
- Autonomously moving the robot.

D. Organization

Section 2 gives a brief information about the simulation environment and how the system is implemented in it. Section 3 consists of a theoretical explanation of SLAM and Section 4 has map generation and SLAM implementation.

II. Simulation Environment

A. What is ROS?

Robot operating system is an open source meta operating system under BSD (Berkeley Software Distribution) license. Code can be written, built and tested across various computers using the inbuilt tools and libraries provided in ROS. Various services like controlling low level devices, implementing generally used functionality, exchanging messages between various processes and management of packages are provided in

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

ROS [5]. ROS enables us to use already created packages like Gmapping and teleop_key which reduces development time. In this paper ROS is used to publish messages in the form of topics between different nodes. ROS is also used to create a virtual environment, generate robot model, implement the algorithms and visualize it in the virtual world rather than implementing the whole system in the hardware itself. The creation of the virtual environment is done using Gazebo and rviz.

B. What is Gazebo?

Gazebo is a 3D dynamic simulator used for simulating complex robotic models in indoor and outdoor environments.It provides with the ability to test the robotic model in the simulated environment and incorporate the data from the sensors.Gazebo allows to test the performance of the robotic model in extreme conditions without damaging the hardware.It uses an physical engine for illumination,inertia,gravity,etc.An XML file called the (URDF) Universal Robotic Description Format is used to describe different elements of the robot. The "Fig. 1" shows the 3D model of turtlebot simulated in gazebo.The robotic model is tested in the environment shown in the "Fig. 2".

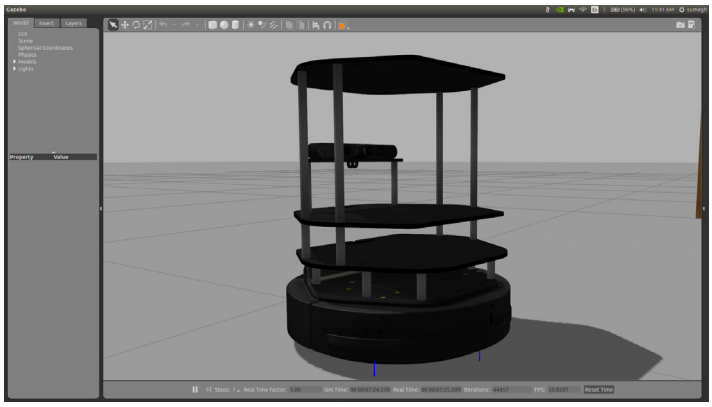


Fig. 1: Turtlebot model in gazebo

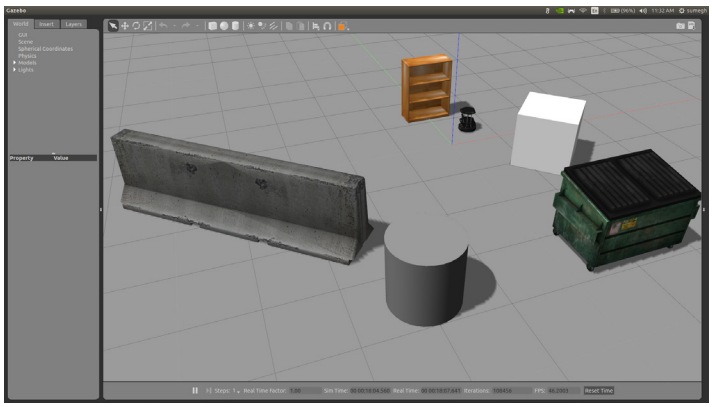


Fig. 2: Simulated environment in gazebo

C. What is Rviz?

Rviz is an acronym for ROS visualization, is a effective 3D visualization tool for ROS. It permits the user to see the simulated robot model, log sensor data from the robot's sensors, and replay the logged sensor data. By visualizing what the robot is seeing, thinking, and doing, the user can debug a robot application from sensor inputs to planned (or unplanned) actions.

Rviz exhibits 3D sensor data from stereo cameras, lasers, Kinects, and other devices in the form of point clouds. 2D sensor data from webcams, RGB cameras can be viewed in rviz as image data.

III. SLAM Algorithm

Autonomous robots are able to navigate indoors and outdoors without colliding with obstacles in its environment. Simultaneous localization and mapping is a technique used for the purpose of making a map, updating it and to estimate the position of the robot. Some of the algorithms that are used to solve it include particle filter, Extended Kalman filter, FastSLAM, Covariance intersection and Graph based SLAM. SLAM algorithm is a combination of Mapping, Sensing, Kinematic modeling, Multiple Objects, Multiple Cameras, Moving Objects, Loop closure, Exploration, Complexity. The main part of SLAM is a range measurement device which is used for observing the environment. The range measurement device depends on different variables. The robot uses landmarks to determine its location using measuring devices and sensors. When the robot has sensed a landmark it extracts the input and identifies the environment. Some types of SLAM algorithm that exist are CoreSLAM, Gmapping, KartoSLAM, Lago SLAM and HectorSLAM. In this paper the SLAM algorithm used is Gmapping which relies on Particle filter. The Objective of the particle filter is to compute the posterior distributions of the states of a Markov process, if there are some noisy and partial observations. Particle filter uses approximate prediction and updates. The samples from the distribution are given in the form of particles, each particle has a weight assigned to it which represents the probability of particles being sampled.

Gmapping is a profoundly proficient Rao-Blackwellized particle filter that creates grid maps from laser data [6]. This package contains a ROS wrapper for OpenSlam's Gmapping. The gmapping package provides laser-based SLAM, as a ROS node called slam_gmapping. Utilizing slam_gmapping, you can create a 2-D occupancy grid map from laser and pose data gathered by a mobile robot .

SLAM algorithm is implemented in various applications such as self driving cars, underwater reef monitoring, mine explorations and unmanned aircrafts. SLAM algorithms are implemented using an open source ROS.

IV. Adaptive Monte Carlo Localization

A key issue with particle filter is keeping up random distribution of particles all through the state space, which gets complicated if the problem is of high magnitude.Because of these reasons it is preferred to utilize an adaptive particle filter which merges a lot quicker and is computationally considerably more effective than a primitive particle filter.

AMCL is a probabilistic localization system, which is used in robots moving in 2D. It performs the adaptive Monte Carlo localization approach, which utilizes a particle filter to follow the position of a robot against a previously known map.

The first step to utilize adaptive particle filter for localization is generate the map of the environment. The robot can be set to a random location or it could start from no initial estimate of its position. After the robot moves forward the generation of new sample starts that will predict the robots position after the command. Random uniformly distributed samples can be added as the robot recovers if it losses track of its position.

V. SLAM implementation

A. Map Generation

The map generation is carried out by first generating the environment and importing turtlebot in Gazebo.The robot model consists of two wheels and a kinect sensor is mounted inside the frame.The mapping process in this paper is implemented using the gmapping package.To use gmapping the robot model must provide odometry data and should be equipped with a horizontally-mounted, fixed, laser range-finder.The kinect captures the depth image of the environment using Infrared sensors and acts as a range finder device.It takes 16 bits per pixel infrared data with a resolution of 640 x 480 as an color image format.This depth image is converted into a 3d point cloud using the depth_image_proc package in ROS.Further this 3d point cloud data is converted into 2d laserscan using pointcloud_to_laserscan package.By providing all the required parameters to the packages in gmapping a map is created in rviz.The “Fig. 3” shows the generated map.Initially the robot is moved in the simulated environment using the teleop_key package in ROS by sending velocity commands via keyboard.The “Fig. 5” shows the rqt graph of gmapping.The gmapping package provides /map topic which can be used by the map_server package to save the generated map.The map is now ready to be utilized for autonomous navigation.

B. Localization

Localization estimates the position and pose of the robot with respect to the environment.To localize the robot laser data, odometry data and the map of environment are essential.The turlebot uses AMCL (Adaptive Monte Carlo Localization) for localization.It follows an probabilistic localization approach for robots in 2 dimensions.In AMCL the pose of the robot is tracked against the known map of the environment using a particle filter.To reduce the computational requirement KLD (Kullback–Leibler divergence) sampling is utilized to automatically adapt for sample size [7].The “Fig. 4” shows the self localized robot rviz.

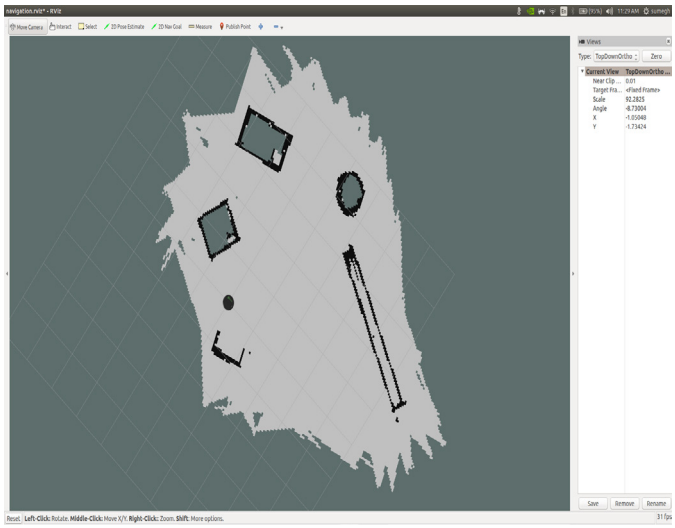


Fig. 3: Visualizing map in rviz

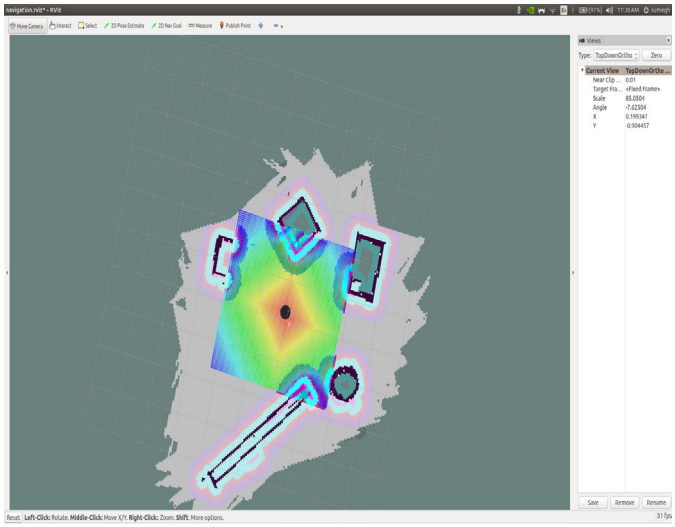


Fig. 4: Localized robot in rviz

C. Rqt graph

The Rqt tool suite in ROS provides a GUI plugon called Rqt graph.The rqt graph is a tool used to visualize all the running nodes and processes as well as the communication between them.The rqt graph can provide a global overview of the system.Rqt can be used as a debugging tool for the system.

D. Autonomous navigation

The autonomous navigation can be performed once the mapping and localization is completed.The ROS navigation stack packages are used to implement AMCL.The node to perform localization on a static map is provided by ROS AMCL package.This node subscribes to the TF data,2D laser

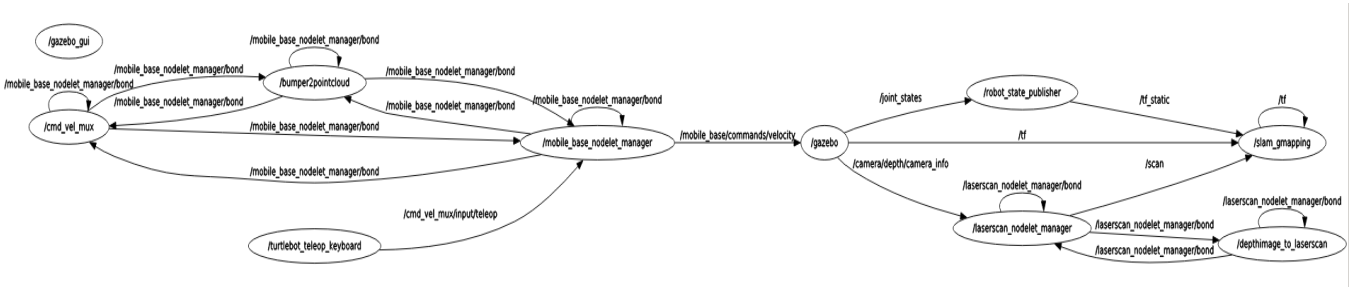


Fig. 5: Rqt graph gmapping

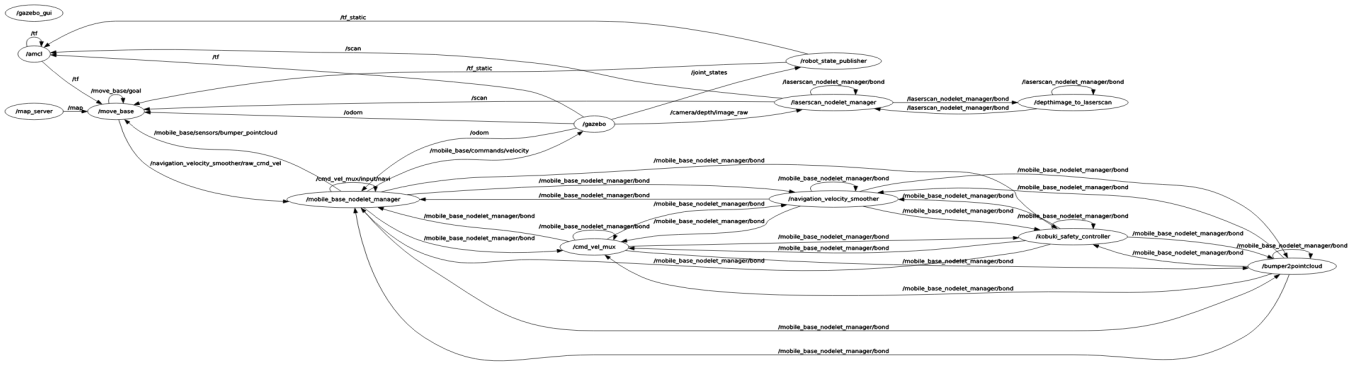
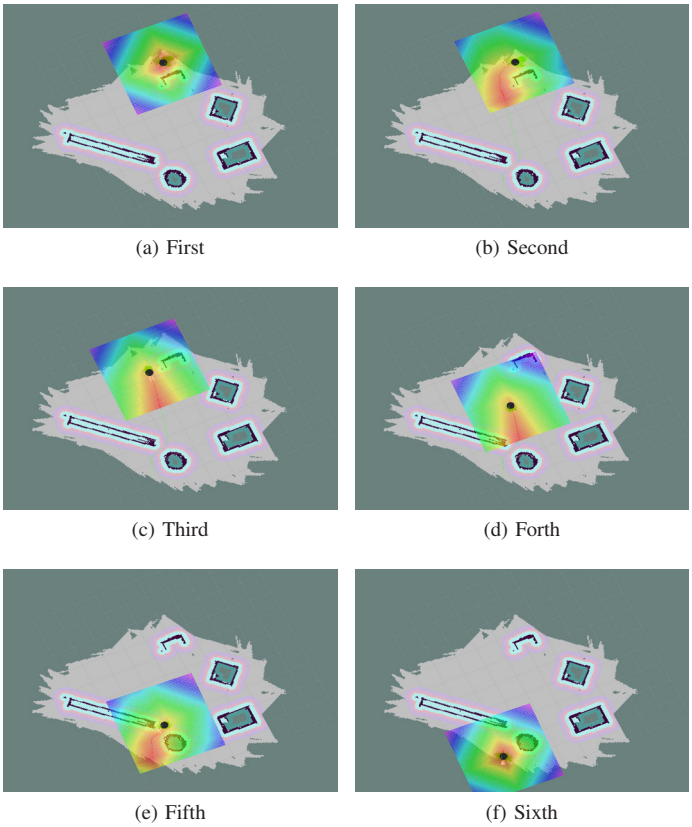


Fig. 6: Rqt graph AMCL

scan data provided by the robot and the previously generated static map.The pose of the robot and its estimated position with respect to the map is published by the AMCL node.The "Fig. 6" shows the rqt_graph of the AMCL.The information about the obstacles in the simulated world is stored in two costmaps.Global costmap is used for global path planning,it makes a long term path plan over the map.The local costmap is used for obstacle avoidance and local path planning.The Rviz visualizes the global and local costmap. The robot can be now moved autonomously .The 2D nav goal in Rviz is used to assign a destination goal to the robot in the map along with its orientation.The path to the desired destination is planned by the robot and velocity commands are given to the robot controller. The "Fig. 7" shows the path followed by the robot to navigate from its initial position to the desired destination goal assigned.



(a) First (b) Second

(c) Third (d) Forth

(e) Fifth (f) Sixth

Fig. 7: Autonomous navigation

VI. Conclusion

In this paper, we have analyzed and successfully executed an autonomous navigation system using ROS. The software and hardware of the system are described in the above section. Turtle bot was used in a simulated environment created with the help of Gazebo for experimenting and implementing autonomous navigation. Methods used for navigation are SLAM and path planning algorithms. As shown in the above section of the map was successfully generated by robot and managed to reach the assigned destination without colliding.

The current work can achieve more precision in navigation through more software development and analysis. The full hardware implementation and calibration for achieving more precision remains for future work.

References

[1] Cherubini, A., Spindler, F. and Chaumette, F., 2014. Autonomous visual navigation and laser-based moving obstacle avoidance. IEEE Transactions on Intelligent Transportation Systems, 15(5), pp.2101-2110.

[2] Turnage, D.M., 2016, December. Simulation results for localization and mapping algorithms. In 2016 Winter Simulation Conference (WSC) (pp. 3040-3051). IEEE.

[3] Omara, H.I.M.A. and Sahari, K.S.M., 2015, August. Indoor mapping using kinect and ROS. In 2015 International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR) (pp. 110-116). IEEE.

[4] Malla, H., Purushothaman, P., Rajan, S.V. and Balasubramanian, V., 2014, November. Object level mapping of an indoor environment using RFID. In 2014 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS) (pp. 203-212). IEEE.

[5] Amanda Dattalo, "Introduction," ros.wiki.org.

[6] Grisetti, G., Stachniss, C. and Burgard, W., 2007. Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE transactions on Robotics, 23(1), pp.34-46.

[7] Zaman, S., Slany, W. and Steinbauer, G., 2011, April. ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues. In 2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC) (pp. 1-5). IEEE.