Autonomous Robot Control Using GMapping And Navigation On ROS

1. 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.Our aim was to build an autonomous navigation platform for indoor application.

a. Overview

Environment Mapping and Navigation is a technique to move an autonomous robot in an unknown environment. The whole project is developed with the ROS framework on ROS melodic as a distribution. The final robot is tested on the Gazebo simulator and visualized in rviz.

Build a robot application that communicates with the robot so that the control of the robot will be done using robot application. Here we are going to use a Gmapping package for SLAM techniques in Gazebo and Rviz. The static environment map will be created with the Laser distance sensor input from the simulated environment. Build a Simulation application that communicates with the Robot application, Gazebo simulator, and Rviz. Develop a simulation robot using URDF and Xacro modeling techniques, that modeled robot will load inside the Gazebo simulator. The Gazebo world will be created with SDF techniques in the ROS.

With the help of Gazebo plugins. We will control the robot and also we will get the Laser Distance Sensor data So that the robot will use SLAM on the Gazebo

simulation environment. Test the Gmapping algorithm with SLAM by using an autonomous robot, control the Robot inside the mapped environment.

b. Purpose

Robot Operating System (ROS) provides us with the architecture to achieve working successfully with sensors and actuators positions and everything every single second. The robotic innovation has quickly paced up since last decade with the advent of ROS wherein the engineers can build robotic apps and programs. Robot navigation is a very wide topic which most of the researchers are concentrating in the field of robotics. For a mobile robot system to be autonomous, it has to analyze data from different sensors and perform decision making in order to navigate in an unknown environment. ROS helps us in solving different problems related to the navigation of the mobile robot and also the techniques are not restricted to a particular robot but are reusable in different development projects in the field of robotics.

The purpose of this project 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.

2. Literature Survey

In the research paper [1], the Authors use ROS with a gmapping algorithm to localize and navigate. Gmapping algorithm uses laser scan data from the LIDAR sensor to form the map. The map is continuously monitored by OpenCV face detection and corobot to spot humans and navigate through the working environment.

The authors of the research paper [2] explain about 2 cooperative robots which work supporting ROS, mapping, and localization. These robots are self-driving and dealing in unknown areas. For this project also the algorithm used is SLAM. Here the most tasks of the robots are to select up three block pieces and to rearrange them during a predetermined manner. With the assistance of the ROS, they made robots for this purpose.

In the research paper [3], the Authors created a simulation of the manipulator and illustrated the methods to implement robot control in a brief time. Using the ROS

and gazebo package, they built a model of a pick and place robot with 7 DOF. They managed to find a robot control which takes less time.

A research paper [4] compares 3 SLAM algorithms core SLAM, Gmapping, and Hector SLAM using simulation. The simplest algorithm is employed to check unmanned ground vehicles(UGV) in several terrains for defense missions. Using simulation experiments they compared the performance of various algorithms and made a robotic platform which performs localization and mapping. The authors of the research paper [6], made a navigation platform with the use of automated vision and navigation framework. With the utilization of ROS, the open source GMapping bundle was used for Simultaneous Localization and Mapping (SLAM). Using this setup with rviz, the turtlebot 2 is implemented. employing a Kinect sensor in situ of a laser range finder, the value is reduced.

Journal [5] explains why the indoor environment is difficult for an autonomous quadcopter. Since the experiment is done indoors they couldn't use GPS, they used a mixture of a laser range finder, XSens IMU, and laser mirror to make a 3-D map and locate itself inside it. The quadcopter is navigating using the SLAM algorithm.

a. Existing Problem

There has been significant progress made when it involves robots perceiving and navigating their environments. Just check out self-driving cars, for instance . Mapping and navigation techniques will still evolve, but future robots have to be ready to operate in environments that are unmapped and poorly understood. For navigation, the grand challenge is to handle failures and having the ability to adapt, learn, and recover. For exploration, it's developing the innate abilities to form and recognize new discoveries. From a system perspective, this needs the physical robustness to face up to harsh, changeable environments, rough handling, and sophisticated manipulation. The robots have to have significant levels of autonomy resulting in complex self-monitoring, self-reconfiguration, and repair such that there's no single point of complete failure but rather graceful system degradation. When possible, solutions were to involve control of multiple heterogeneous robots; adaptively coordinate, interface, and use multiple assets; and share information from multiple data sources of variable reliability and accuracy.

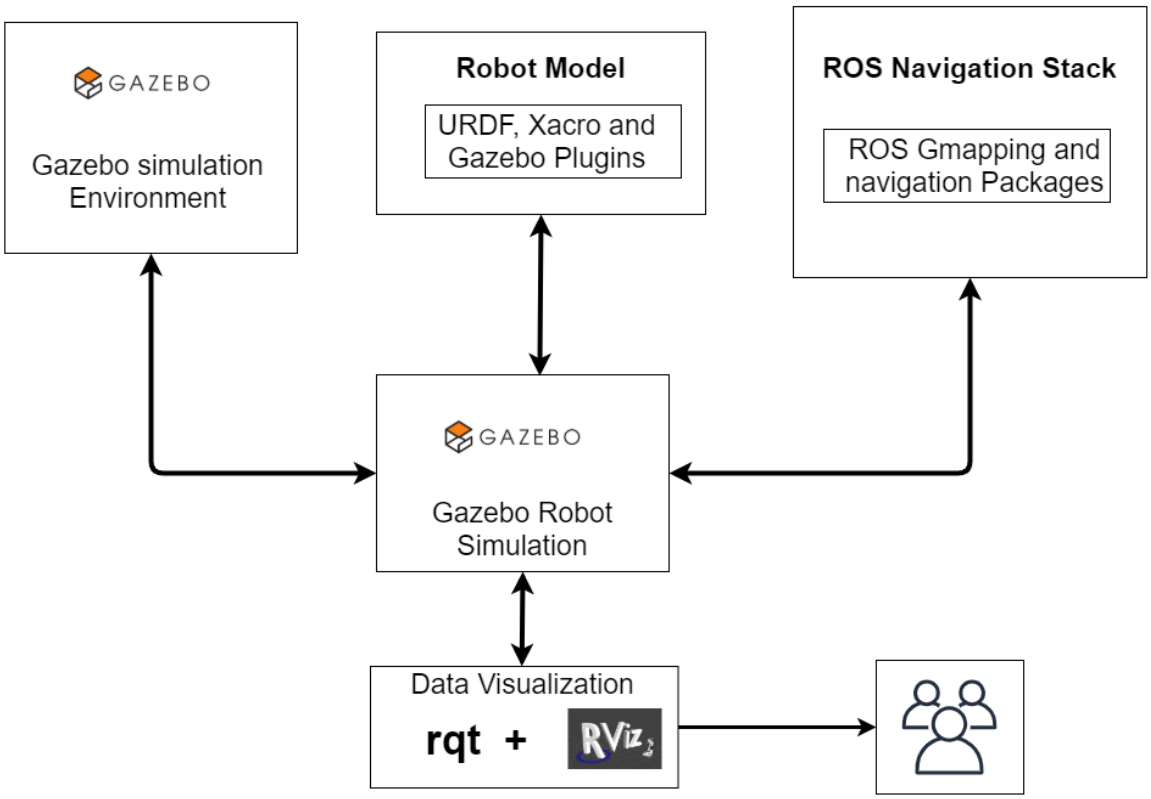
b. Proposed Solution

We have proposed to Build a robot application that communicates with the robot so that the control of the robot will be done using robot application. Here we are going to use a Gmapping package for SLAM techniques in Gazebo and Rviz. The static environment map will be created with the Laser distance sensor input from the simulated environment. Build a Simulation application that communicates with the Robot application, Gazebo simulator, and Rviz. Develop a simulation robot using URDF and Xacro modeling techniques, that modeled robot will load inside the Gazebo simulator. The Gazebo world will be created with SDF techniques in the ROS.

With the help of Gazebo plugins. We will control the robot and also we will get the Laser Distance Sensor data So that the robot will use SLAM on the Gazebo simulation environment. Test the Gmapping algorithm with SLAM by using an autonomous robot, control the Robot inside the mapped environment.

3. Theoretical Analysis

a. Block Diagram



b. Hardware/Software Designing

Virtual Box - Oracle VM **VirtualBox** is cross-platform virtualization software. It allows users to extend their existing computer to run multiple operating systems including Microsoft Windows, Mac OS X, Linux, and Oracle Solaris, at the same time.

Ubuntu 18.04 - Ubuntu is a complete Linux operating system, freely available with both community and professional support.

Robot Operating System (ROS) - It is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex
and robust robot behaviour across a wide variety of robotic platforms. ROS is a framework that sits on top of an existing operating system such as GNU/Linux. Packages are provided for Ubuntu Linux to help get your robot up and rolling. We are using this framework to create an automation gmapping and navigation system.

Python - It is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.

We are choosing python because it:

1. Readily available and Open source
2. Huge Libraries
3. Easy to Understand and learn
4. Big developer Communities
5. Fewer code lines and larger Functionalities

Gazebo - Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. With the help of this we are able to simulate our robot in a virtual world.

RViz - It is the 3D visualization tool of ROS. The main purpose is to show ROS messages in 3D, allowing us to visually verify data. We use this tool to visualize our robot, laser scan and maps.

4. Experimental Investigation

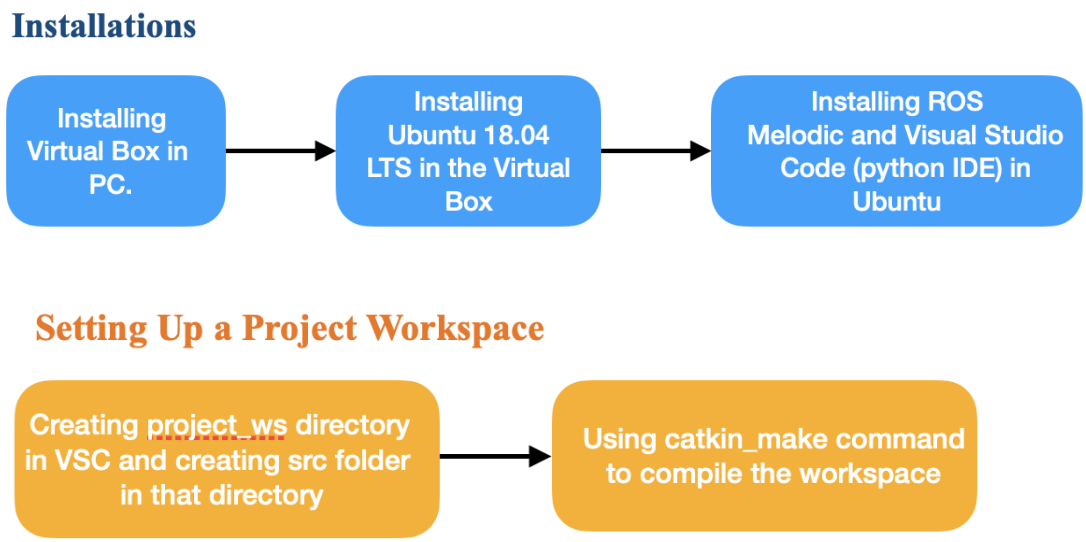
With technological advances in sensing devices and computing the problem known as Simultaneous Localization and Mapping (SLAM) has been one of the most worked problems by the Robotics research community in the last decade, which has produced several contributions To solve the SLAM problem, a map of the surroundings of a robot should be incrementally constructed at the same time that the robot pose (localization and orientation) is continuously estimated with respect to this map. Since the navigation of the robot in an environment involves knowing its own location and those of its goals, determining the best solutions for SLAM is considered an important step towards enabling mobile robots to operate without human intervention.

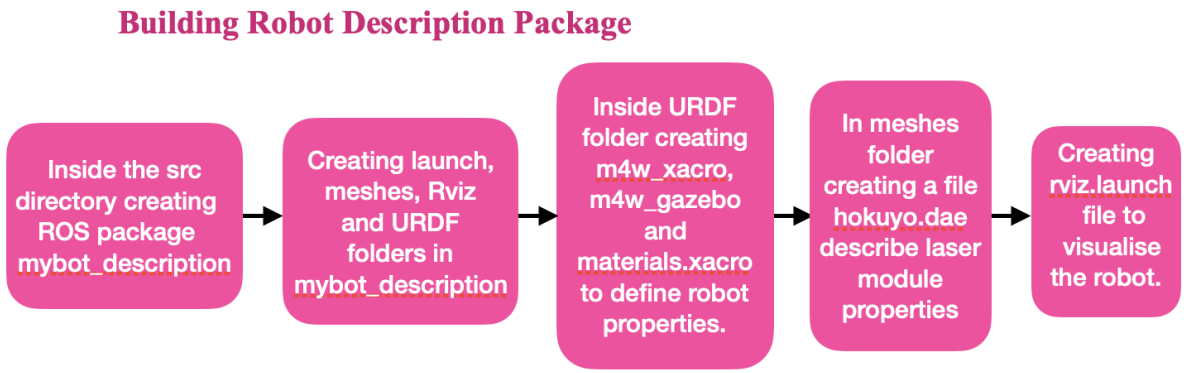
The interest for SLAM has increased due in large part to the Robot Operating System (ROS) , which has been proposed to facilitate the development of software for robotics and has become the de facto standard robotics middleware. Facilities like offering inter process/inter machine message passing mechanisms

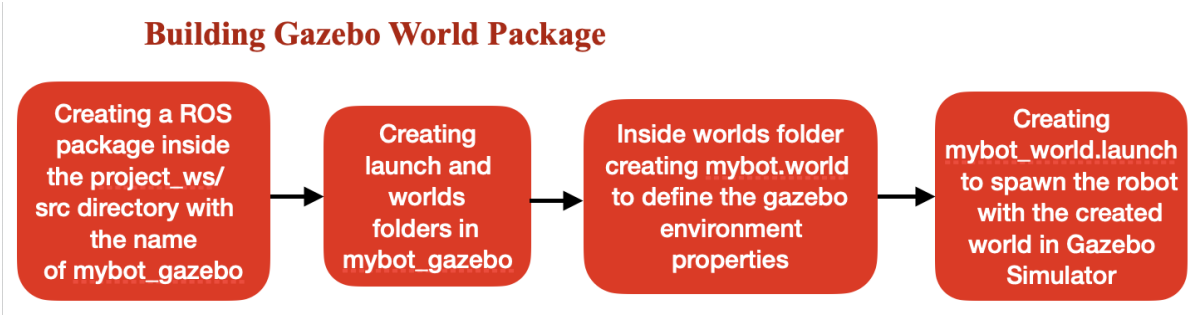
and hardware abstraction for handling data from robot sensors and controllers explains why ROS has been widely adopted in robot research and industry .

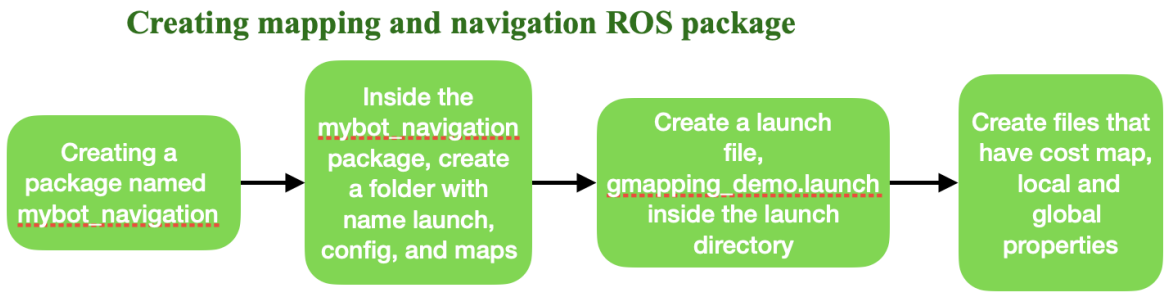
ROS is open source as well, which makes its development reach more visibility. Nevertheless, estimates for the pose of a robot and the mapping of its environment of operation, alone, are not sufficient for allowing the robot navigation. Besides offering SLAM estimates with the lowest possible errors, a robot must plan a trajectory from its location to goal locations using a suitable map representation, such as occupancy grid and then execute the necessary commands to traverse the planned route. The broad adoption of ROS by the community has contributed substantially to the development of novel approaches on both SLAM and robot navigation . The solutions are available as ROS packages, a high level of abstraction software ready to be used on any ROS compatible robot. It is often the case that roboticists and mainly non roboticists are posed with the choice of which SLAM algorithm should be employed on a given application in view of their requirements.

5. Flowchart









6. Results

We see that the robot moves autonomously by considering the laser scan values inside the Gazebo simulator. Users can visualize a robot model moving in the Gazebo environment in ROS.

7. Advantages and Disadvantages

1. Increase efficiency and productivity
2. Reduce error, re-work, and risk rates
3. Improve safety for employees in high-risk work environments
4. Perform lower value, mundane tasks so humans can work collaboratively to focus on more strategic efforts that cannot be automated

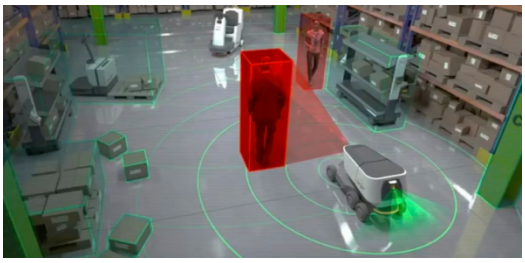
5. Enhance revenue by improving perfect order fulfillment rates, delivery speed, and ultimately, customer satisfaction

8. Applications

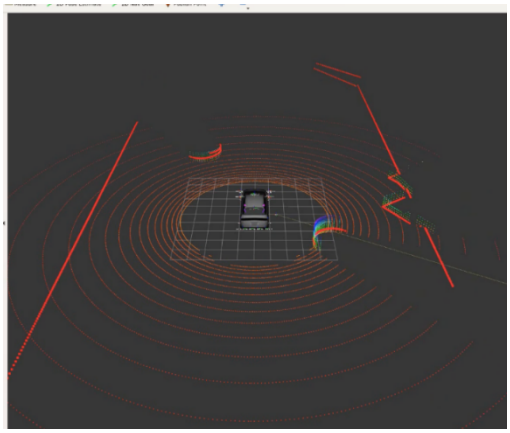
Autonomous Robot navigation is the sole purpose here.

- Object detection, avoidance, maps and navigation is everywhere and required for every moving electronic device .From a small mobile robot used indoors to a big industrial robot we want our robot to navigate autonomously and fulfil its purpose.

- supply chain innovation:improve the speed and accuracy of routine operations, particularly in warehousing and manufacturing spaces.[6]



- self-driving cars[7]



9. Conclusion

This work proposed an experimental analysis of two widely used ROS compatible SLAM packages: GMapping and RTAB-Map. The experiments were carried out by executing SLAM and navigation sessions on datasets from Gazebo simulation, TUM RGB-D benchmarks and using those collected in our university Lab. A discussion is drawn relating practical aspects of the algorithms Preprints with the inner workings and main parameters of the respective ROS implementation. Specifically, we found that RTAB-Map is a more complete solution for ground robots equipped with a RGB-D sensor, mainly because of capabilities like accuracy, quality of produced maps and configuration flexibility, which allows it to reuse maps in localization only mode.

Besides the detailed analysis performed, our dataset is also another contribution to the community that can use it in further works. So we believe that the analysis reported on this work should shed light for researchers involved with ROS application design in choosing a SLAM package that computes estimates suitable for ground robots equipped with a RGB-D sensor to autonomously navigate an environment.

10. Future Scope

The concept of localization and mapping was being scaled at various operations but the work here focuses on using SLAM on ROS to autonomously navigate and simultaneously avoid any collisions. Furthermore, the objective is to improve this method by embedding it with an arm to lift objects using image recognition which can be of great help to the blind.

The concept of localization and mapping was being scaled at various operations but the work here focuses on using SLAM on ROS to autonomously navigate and simultaneously avoid any collisions. Furthermore, the objective is to improve this

method by embedding it with an arm to lift objects using image recognition which can be of great help to the blind[8].

11. Bibliography

[1] Emil-Ioan Voisan, Bogdan Paulis, Radu-Emil Precup and Florin Dragan, "ROS-Based Robot Navigation and Human Interaction in Indoor Environment," in 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics, May 21-23

[2] SangYoung Park and GuiHyung Lee, "Mapping and Localization of Cooperative Robots by ROS and SLAM in Unknown Working Area," in Proceedings of the SICE Annual Conference 2017 September 19-22, 2017.

[3] Wei Qian, Zeyang Xia, Jing Xiong, Yangzhou Gan, Yangchao Guo, Shaokui Weng, Hao Deng, Ying Hu, Jianwei Zhang, "Manipulation Task Simulation using ROS and Gazebo," in 2014 IEEE International Conference on Robotics and Biomimetics December 5-10, 2014.

[4] Doris M. Turnage, "SIMULATION RESULTS FOR LOCALIZATION AND MAPPING ALGORITHMS," in 2016 Winter Simulation Conference, 2016.

[5] Sławomir Grzonka, Giorgio Grisetti, and Wolfram Burgard, "A Fully Autonomous Indoor Quadrotor," IEEE TRANSACTIONS ON ROBOTICS, VOL. 28, NO. 1, FEBRUARY 2012.

[6] (2021).Retrieved 31 July 2021,from https://www2.deloitte.com/content/dam/Deloitte/us/Documents/manufacturing/us-supply-chain-of-the-autonomous-robots.pdf

[7] How to Start with Self-Driving Cars Using ROS - The Construct. (2021). Retrieved 31 July 2021, from https://www.theconstructsim.com/start-self-driving-cars-using-ros/

[8] Nupur Choudhury; Rupesh Mandal "Human Robot Interaction Using Android and Point Bug Algorithm". Published in 2016 under IEEE.

12. Appendix

a. Source Code

Github Link:
https://github.com/smartinternz02/SI-GuidedProject-4798-1627035523/tree/main/Source%20codes

b. Robot Output Screenshot