Autonomous Navigation of Bot

Project Description

The Navigation Stack is fairly simple on a conceptual level. For the project we are going to do whole project using ROS (Robot Operating System)
The Robot Operating System (ROS) is an open-source framework that helps researchers and developers build and reuse code between robotics applications. It takes in information from odometry and sensor streams and outputs velocity commands to send to a mobile base. Use of the Navigation Stack on an arbitrary robot, however, is a bit more complicated.

Approach

- Design and build your customized physical robot with autonomous navigation capability
- Create a map of your house using the lidar scanner of the robot
- Localize the robot in the map
- Navigate the bot in the map between 2 points in rviz

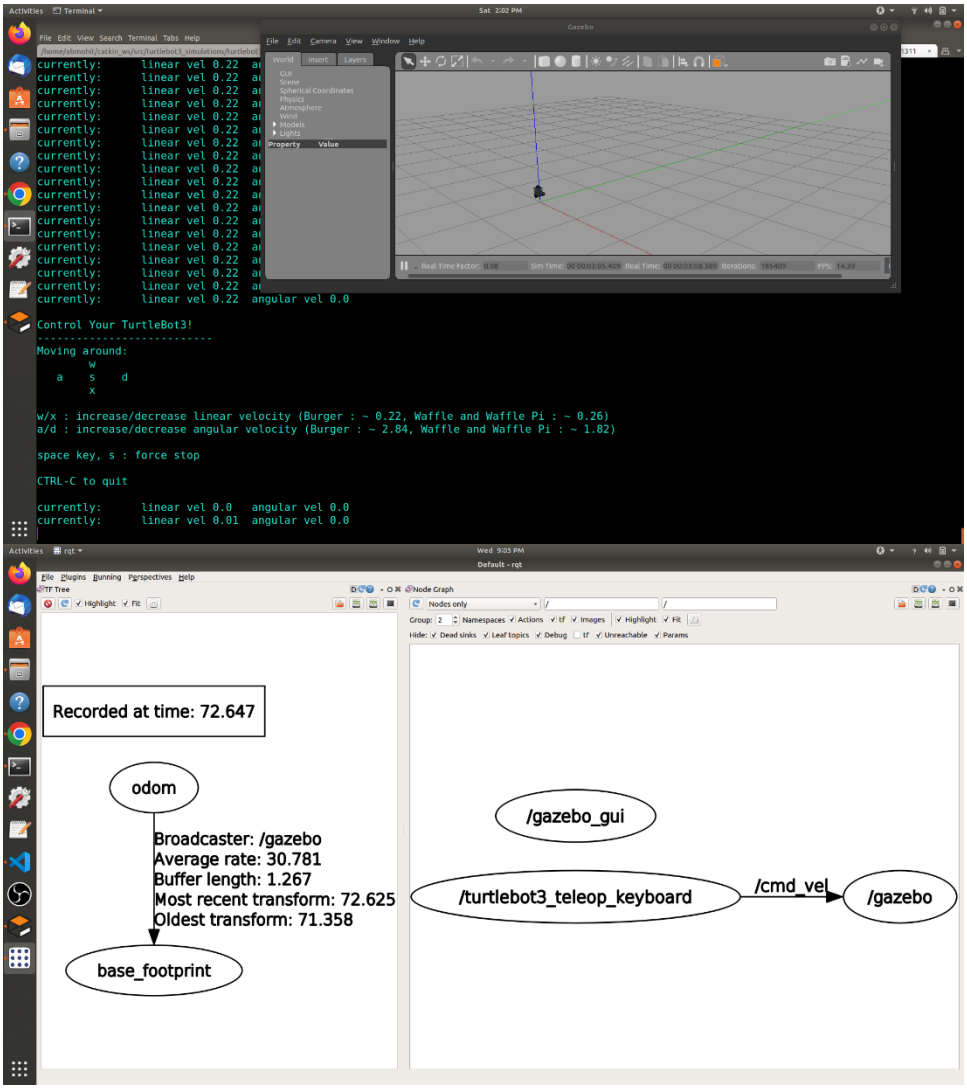
Tech Stack

- Google Colab
- Ros-melodic desktop full
- Git and Github
- VS-Code
- Python
- Discord
- Google doc

TASK

1) **Turtlebot3 simulation**

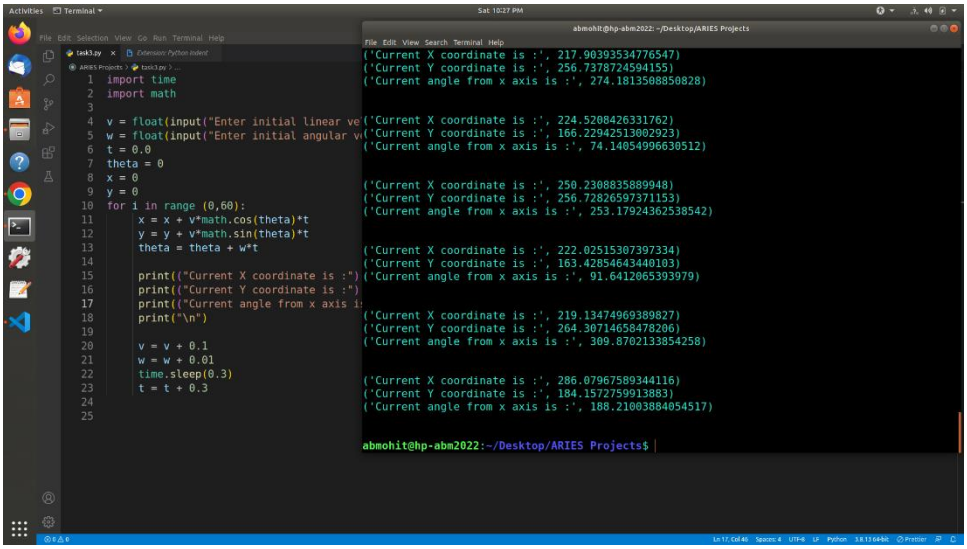
   Create a workspace. Install turtlebot3 packages. Run gazebo simulation. Move the turtlebot using teleop keys.



2) **Pseudocode for odometry calculation**

Read the provided pseudo code (git repo) to get an idea regarding pose calculation and application of sensor data.

3) **Writing python script for pose calculation by taking fake input of v and w**



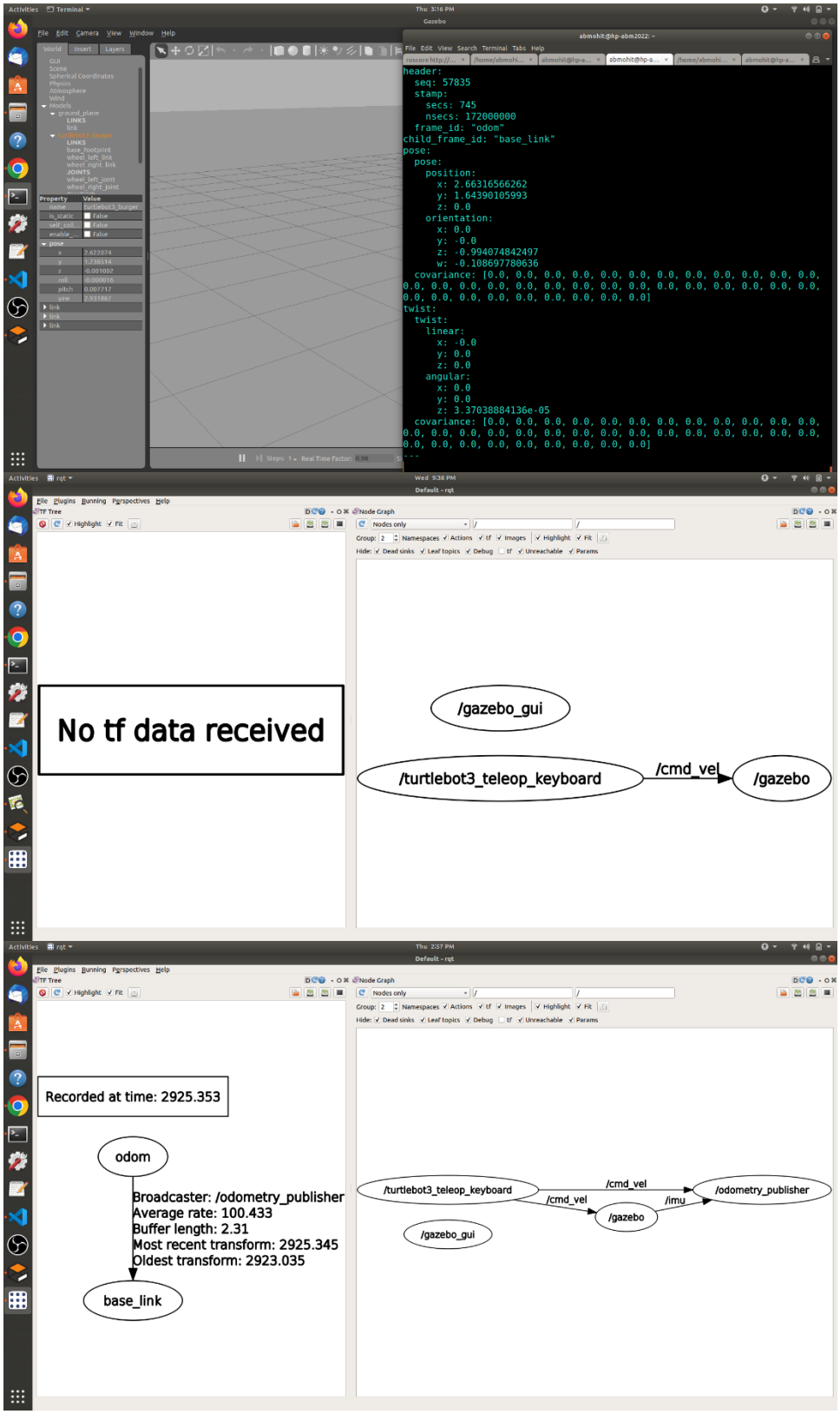
4) **Reading git repo file of odometry calculation**

Understanding the concepts involved in publishing of odometry information in ROS.

https://gist.github.com/atotto/f2754f75bedb6ea56e3e0264ec405dcf

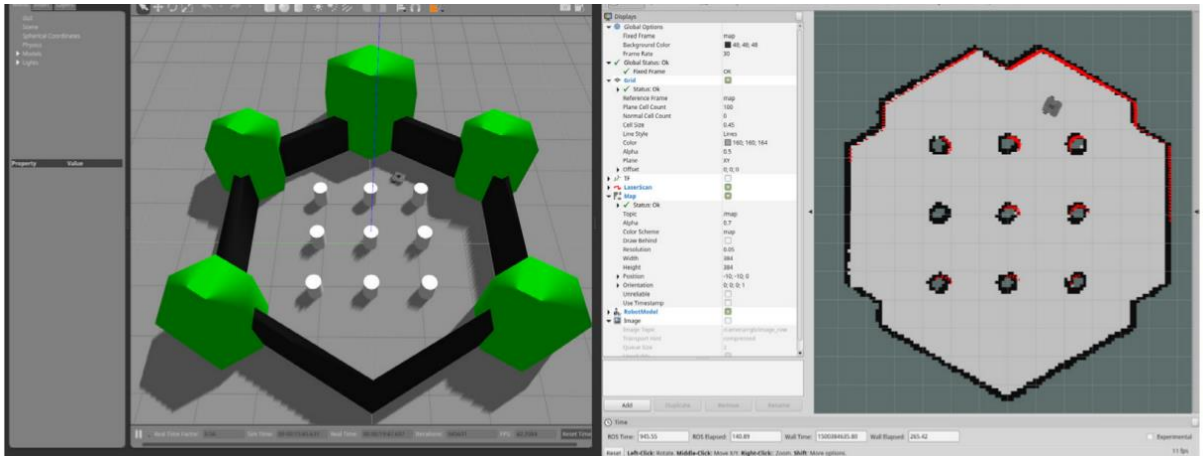
5) **Writing custom odometry node**

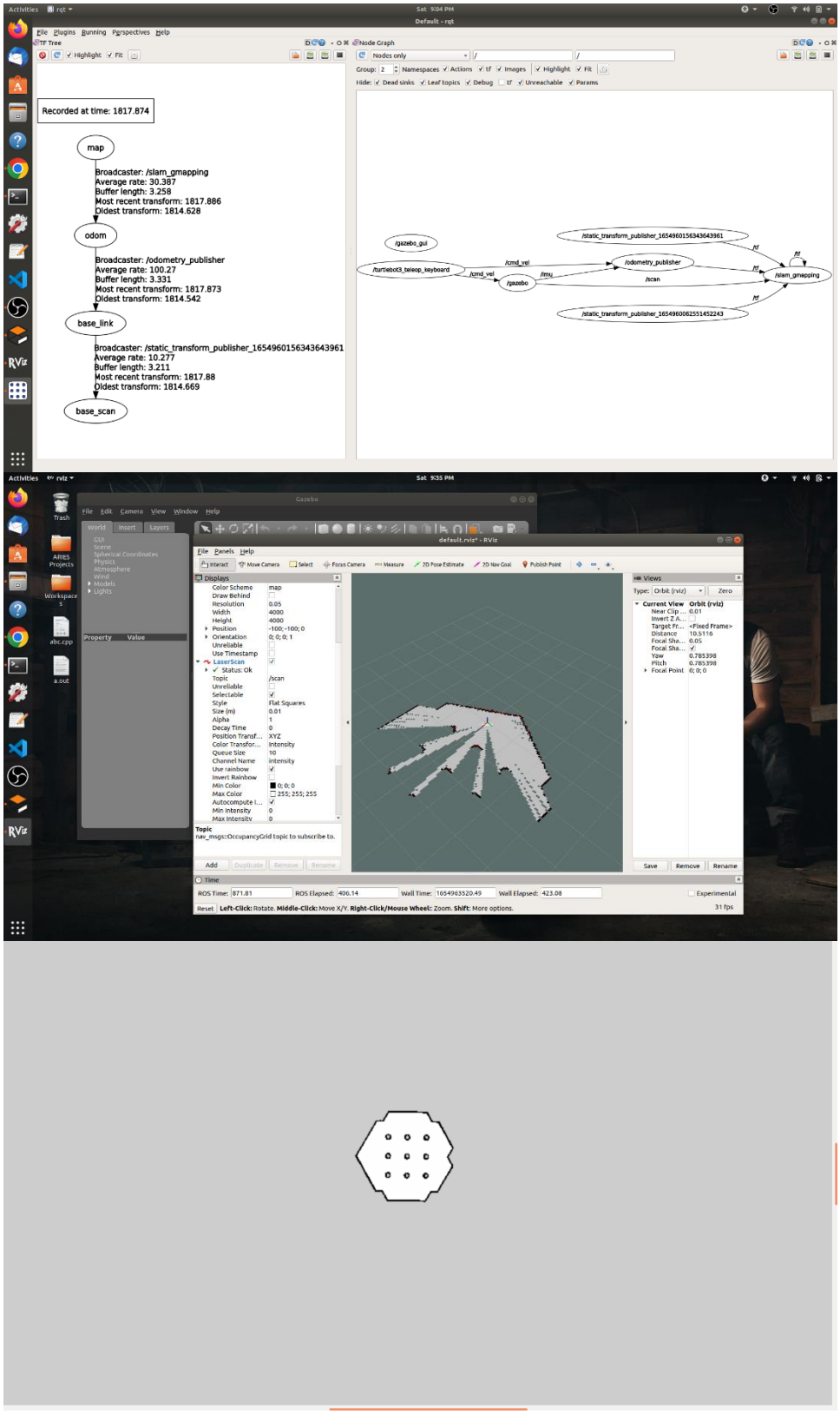
An incomplete python file was to be edited and run. File was written using class. Input is taken from subscribing to relevant topics(cmd_vel, imu).



6) Mapping of an environment using gmapping package

Create a map using gmapping package in gazebo environment. Save the map





Video link :
https://drive.google.com/file/d/1wilrebbXFjb1g707sQ6bZWtSqFjk7_uS/view?usp=drivesdk

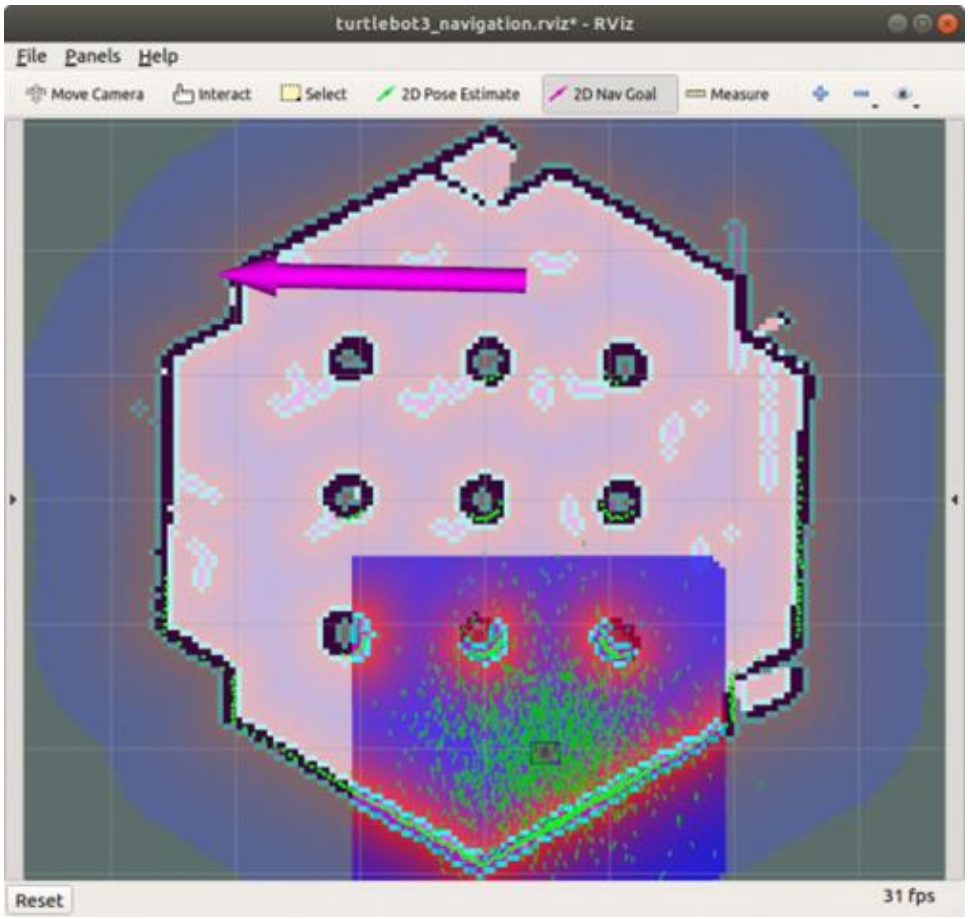
7) **Localization of bot in map using amcl package**

Clone navigation package form git repo in workspace. Move the bot using teleop and compare the coordinates calculated and in gazebo(real world).

Videl Link :
https://drive.google.com/file/d/1O0l1ZoAuxByhk6PjZm1r8EU9cTNCinP4/view?usp=drivesdk

8) **Navigation in map using move base**

Set initial and final pose in map in Rviz. The bot will now navigate from initial and final pose choosing the shortest path. Instead of calculating Odometry on our own, odometry estimate is taken from turtlebot3 package.



Video Link : https://drive.google.com/file/d/12U-ZFfdfukMCP64Ul8YYy5Yr2LTFvG-g/view?usp=sharing

References/Resources

- https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/
- https://gist.github.com/atotto/f2754f75bedb6ea56e3e0264ec405dcf