VINEET KUMAR SINGH (Dir Id: vsingh03) PROJECT 1 : ENPM662 FALL2022

28th Oct 2022

1. Team Members

1. Shantanu Suhas Parab (Directory Id: sparab)

2. Vineet Kumar Singh (Directory id: vsingh03)

2. Introduction

This project focuses on design of a rear wheel drive mobile robot with independent steerable front wheels. The vehicle is mounted with LiDAR to scan the surrounding environment. The model can be controlled either by using teleop keys or using a simple publisher and subscriber and we have implemented both these approaches. This report discusses the steps of the project from model design in Solidworks to spawing the robot and controlling it in a simulated environment using ROS and Gazebo.

3. Building a CAD model in Solidworks

The robot model was first conceptualized for the given constraints of dimensions of wheel base and model width and length. Then we designed a simple L angle wheel mounting to make both the front wheels independent steerable. The front joints between the L angle and body is revolute and so are all the joints between the wheels and L angles. The rear wheels are also designed the same as the front wheels but with a fixed joint with the robot body.

Once the design was final and assembled with proper mates, the final axes and the global origin was assigned in the plane having the lower ends of the wheels. This origin assignment was done in 5th iteration after we faced issues with robot spawning in Gazebo and other solutions did not work out. After the origin and axis assignments the model was exported as URDF using the URDF exporter plugin which generate the toy_car2 package having the associated launch files, urdf files and meshes.

4. Adding Controllers and Lidar to the model.

The URDF file generated from Solidworks was then changed to add dummy links to work properly while spawning in Gazebo. Further to this we added the LiDAR files as per the project Instructions. We faced issues after LiDAR mounting, where our model was not spawning correctly in Gazebo. After multiple tries, we finally solved this by changing the Global origin as described in Solidworks section.

The process of locating the LiDAR on the top of the vehicle was iterative. We tweaked the location of the LiDAR in the URDF file for proper location on the top of the vehicle. Then we added all the transmission blocks at the proper joints and assigned appropriate controllers. After this we generated a new URDF using the XACRO file given as part of the project description.

Using the new URDF from XACRO file we updated the .yaml file for controllers. Coding the **.yaml** files was a challenge as this was something we never worked on, and this led to multiple errors while trying to control the robot in Gazebo. After multiple changes to the URDF transmission section we achieved the correct configuration for the controllers and its interface and the model was spawning correctly in Gazebo.

After this we spawned the robot in Gazebo Competition world provided in Project Description and visualized the robot in RVIZ for the LiDAR outputs. We configured the LiDAR in RVIZ and we were able to see the objects being detected by the LiDAR. The screencast video for the same is present in the Video links section below.

Once all the functions were working perfectly, we made the final launch file (named final_launch.launch) based on the template_launch.launch file and did all the necessary changes to properly spawn our robot in Gazebo and its visualization in RVIZ.

5. Running the bot using teleop

Once our robot package was ready, we downloaded the template teleop file and did changes accordingly to publish the commands to each of the robot controllers. Then we launched the robot model in Gazebo and moved the robot from the center to one of the corner in the Competition arena world. We changed the speed by a factor of 50 in the publishing command so as to speed up the movement of our bot. The screencast video link for the teleop control of the robot is given in the Video links section at the end.

6. Subscriber and Publisher for control

As the last part of this project, we made two nodes, one subscriber and one publisher, which makes the robot traverse in circles. We use two different nodes for the velocity and steering angle as using this approach saves us from writing a custom rostopic for this project. The data being published by the publisher and then subscribed by the subscriber can be checked in the terminal. Details of running the code is provided in the readme file.

7. Conclusion

In this project we made mobile robot and controlled the robot using ROS and visualized our model in Gazebo and RVIZ. We learned how to use external modeling software for modeling our robot and the process of controlling these models using ROS. We explored the URDF files and how it is the core of every model definition and its importance in visualizing a robot. Choice of proper transmissions and controllers was very crucial for this project and we also got used to the yaml and xacro files. Finally, controlling the robot using teleop and its visualization in Gazebo helped us to analyze the gaps in our modeling and we did an iterative process between design in Solidworks and ROS models to optimize our robot for the task to be done. Overall from this project we got a good intuition of the process pipeline for modeling a robot and its control using ROS.

8. Video Links and Running the Code

Videos are uploaded on YouTube and made public. Link for the video is below:

https://youtu.be/C-zojYn1Fj8 - Lidar output

https://youtu.be/p3Kq7AhRVoQ - Teleop control of robot

https://youtu.be/RfGgYC5VbuE - Publisher and Subscriber for robot control

The **readme** file is present in the zip folder and package which defines the step by step process and terminal codes to load the robot package in catkin_ws and running the code. The commands are given for all the conditions which needs to be checked as per the Project Description document.