ENPM 662 Introduction to Robot Modeling *Varun Lakshmanan - 120169595*

Project – 1 Report

Steps involved in accomplishing the results:

Step - 1: Creating a model of robot using SolidWorks Software.

Step - 2: Converting the designed Robot model to URDF file and exporting it.

Step - 3: Exporting the URDF file from Windows to Ubuntu operating system.

Step - 4: Setting up a workspace in ubuntu to create a ROS2 package.

Step - 5: URDF and Mesh folders are added to ROS2 package and CMakeLists are Updated.

Step - 6: Making the URDF file to be flexible to add color, a LiDar Sensor and IMU.

Step - 7: Providing Mobility to Robot Joints by Adding Controllers.

Step - 8: Sourcing Commands to Initialize the controllers and data is published to the joints.

Step - 9: The LiDar properties of the robot are evaluated in RviZ and Obstacles are introduced to robot in order to test it.

Step -10: Teleoperation code is introduced into the workspace and the robot is controlled using keyboard (Teleop Controller).

Step - 11: The Robot is tested in a competition track using teleoperation.

Step - 12: A Proportional controller is implemented to move the robot in a straight line.

Step - 13: An (Error vs Time and Control vs Time) graph is plotted to record performance of the proportional controller.

Errors and Troubleshooting done while doing the Project:

- **URDF Frame Assignment:** We came across difficulties while making the URDF conversion due to incorrect frame assignments. The initial assignment of frames was not accurate, resulting in problems.
- **LiDAR Visualization in RVIZ**: While visualizing LiDAR data using RVIZ, we initially faced a challenge with the visualization due to an oversight where we had forgotten to update the frame name in the URDF, which prevented us from noticing the expected red dots.
- **Joint State Publisher GUI**: At the outset, we came across an issue where the sliders of the Joint State Publisher GUI were not visible. We resolved this by installing the Joint State Publisher GUI, ensuring that we could control the robot's joints effectively.
- **Teleoperation Spin Issue:** During teleoperation, we came across an issue where the robot was spinning in one place rather than moving as it should. We Resolved this Problem by adjusting the sign of the velocity published, thus making us achieve the movement desired.
- **Closed-Loop Controller Position Calculation:** In the process of developing the closed-loop controller, we encountered a problem with finding the robot's location on the map. To solve this, we had to compute the yaw and the distance moved by using the orientation and linear acceleration data from /imu_plugin/out. This change allowed us to successfully regulate the robot's position within the environment.

Enhancements Made in Documentation:

We found the documentation very helpful for our project, as it provided us with useful instructions and assistance. But when we moved on to the closed-loop control stage, we faced a specific problem. Getting the exact location of the robot from the IMU data was very hard. Even though the documentation was helpful, this problem was very challenging and demanded more focus and creativity.

Individual Contributions:

- The Robot Model is Designed and Exported by Varun Lakshmanan (UID: 120169595).

- The Conversion of the package compatible with ROS2 is done by Nitish Ravisankar Raveendran (UID: 120385506).
- The Laser and controllers are added to the robot by Nitish Ravisankar Raveendran (UID: 120385506) and Varun Lakshmanan (UID: 120169595).
- The Teleoperation of the Robot in Competition Setup is done by Nitish Ravisankar Raveendran (UID: 120385506).
- The Proportional controller is implemented by Nitish Ravisankar Raveendran (UID: 120385506) and Varun Lakshmanan (UID: 120169595).

Images of Steps involved:

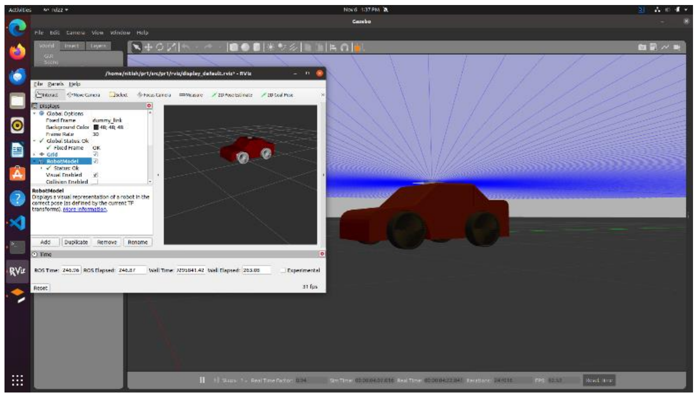


Fig. RviZ

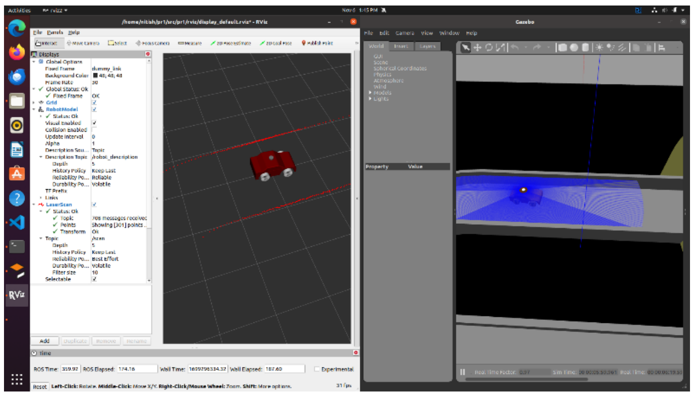


Fig. LiDAR Visualisation

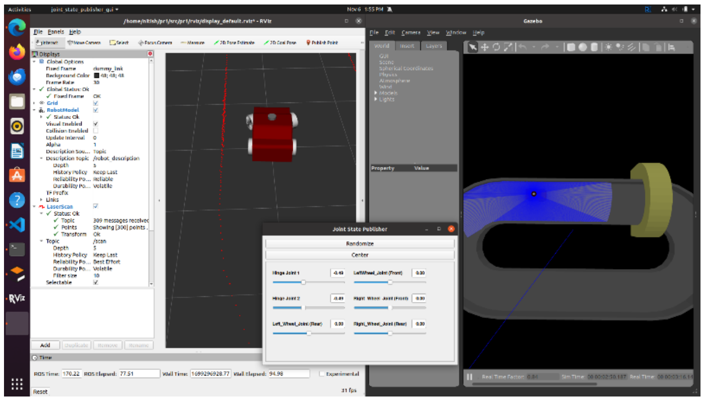
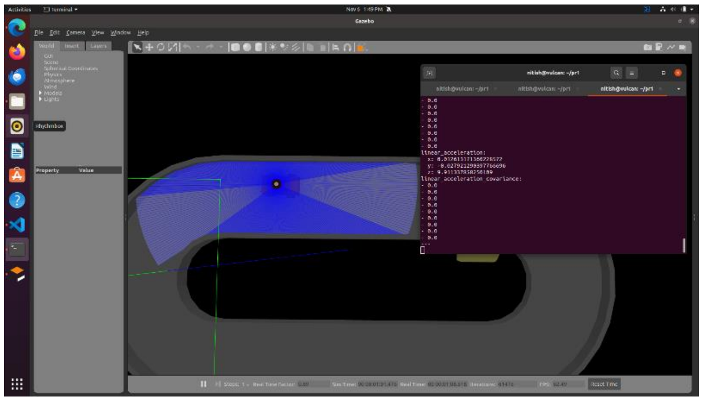


Fig. Join State Publisher Sliders



```
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
linear_acceleration:
  x: 0.012611171386228572
  y: -0.027921299397766696
  z: 9.011337858258109
linear_acceleration_covariance:
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
---
```

Fig. IMU Visualization

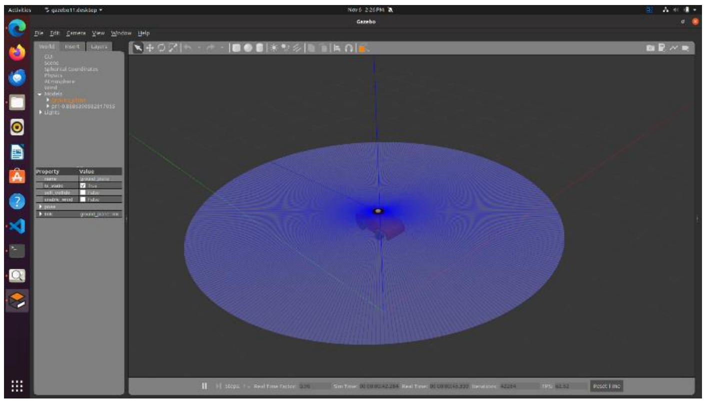


Fig: Car in Gazebo

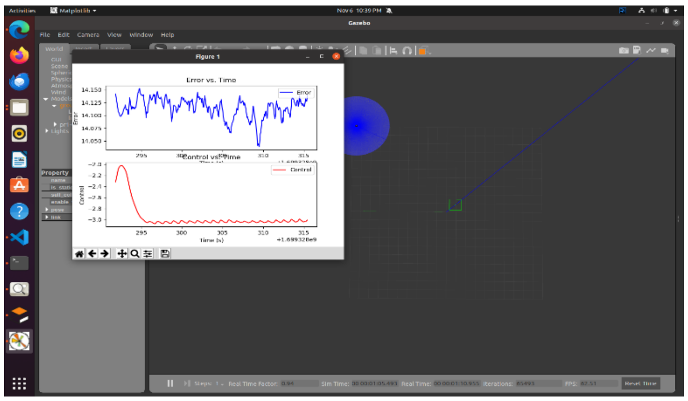


Fig: Error vs Time and Control vs Time Graph

Teleop => [Click Here](#)   OpenLoop Controller => [Click Here](#)