ENPM 662: Introduction to Robot Modeling

Project 1 - Group 2 Report

Group Members: Manas Desai, Grayson Gilbert, Kunj Golwala, Shreya Kalyanaraman

---

RVIZ Visualization and Pose Plot -

Figure 1 below shows the visualization of LiDAR sensor data points in RVIZ (left) collected by the robot in the competition gazebo world (right). This view helped us verify that the LiDAR was correctly configured and functioning. Figure 2 represents the pose plot obtained through the subscription method in autonomous mode, showing robot's path toward the target location.

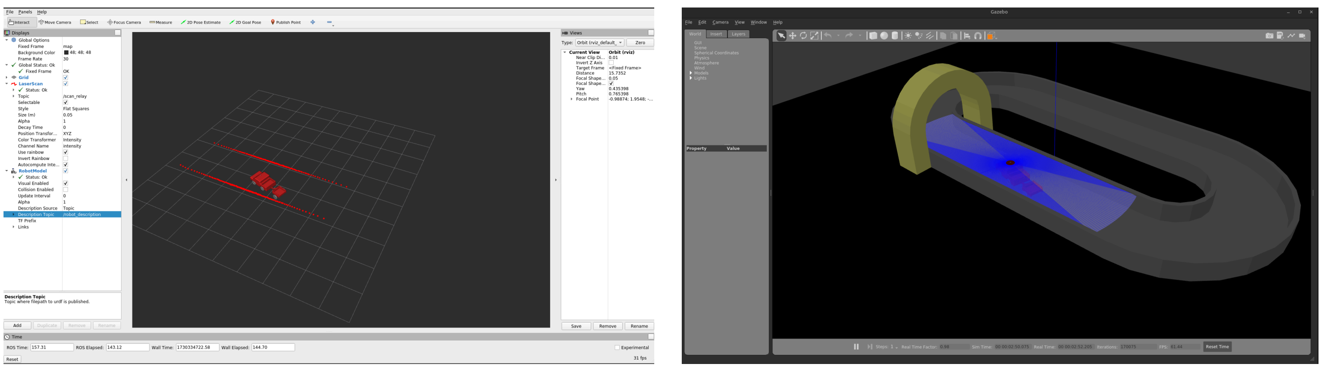

Figure 1: LiDAR Sensor Data Points in RVIZ

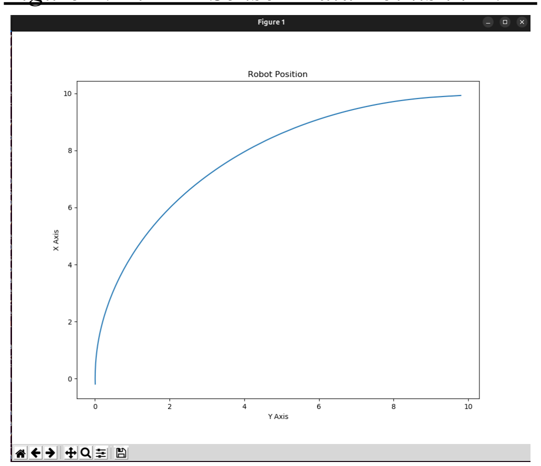

Figure 2: Robot Pose Plot in Autonomous Model

Steps Followed -

a) CAD Modeling and URDF Export - First, we designed a Truck and Trailer CAD assembly in Solidworks. The assembly included all the joints and mates necessary to properly simulate the basic vehicle function. Then, we used the URDF exporter tool to generate a ROS package of our robot model.

b) **ROS2 Compatibility -** After designing the Truck and Trailer assembly, we needed to modify the URDF package to properly load the models into Gazebo and be compatible with ROS2. This involved updating the URDF, CMAKE, and Package files.

c) Adding Controllers and LiDAR Sensor - Next, we added position and velocity controllers in order to control the Truck and Trailer assembly in Gazebo. A LiDAR sensor was also added to the vehicle to detect nearby objects, and it introduced us to RVIZ for visualizing sensor data.
d) Teleoperation Mode - At this point, the ROS2 package was set up, and we began to make the Truck and Trailer move in an empty Gazebo world. After debugging issues within the CAD model and URDF file, we successfully incorporated the Teleoperation script into our project.
e) Autonomous Mode with Proportional Controller - The final piece to the project was autonomous navigation via a proportional controller. In order to tackle this component, we leveraged knowledge from Project 0 and previous lectures to better understand how to implement the controller. Using the previous lectures to solve the kinematic equations, and a Turtlebot3 controller for guidance on how to design the node, we developed a proportional controller to regulate vehicle speed throughout a calculated trajectory.

Challenges Faced -

1. Vehicle Weights - Due to the truck style vehicle design, the center of gravity of the vehicle was shifted more towards the rear of the vehicle. This caused handling issues during teleoperation. It was finally resolved by reducing the weight of the truck body.
2. RVIZ LiDAR Data - The RVIZ default settings were generating an error when trying to visualize LiDAR data on the /scan topic. To resolve the issue and visualize the data, we created a relay node called /scan_relay that used settings compatible with RVIZ.
3. Frame assignment in solidworks vs URDF - We had to reverse the axis of rotation assigned for drive wheel joints and steering wheel joints
4. Crooked Wheels during Solidworks Assembly - The CAD assembly was originally exported with the wheels angled slightly. This resulted in the robot spawning into Gazebo with angled wheels even though the position controller thought they were straight forward.
5. Debug.launch file issue - The debug.launch file was not launching the joint state publisher resulting in incorrect joint transformations for the robot in RVIZ. Updating the joint_state_publisher node and launchDescription fixed the issue.

Results and Observations -

Teleoperation Mode: The robot successfully completed a lap in the Gazebo competition environment under teleoperation. The LiDAR sensor effectively mapped objects, providing real-time obstacle feedback, and RVIZ was able to depict the sensor data as the vehicle traveled around the track.

Autonomous Mode: The proportional controller worked effectively in guiding the robot from point A to point B. The robot's pose plot (Figure 2) illustrates a continuous trajectory toward the desired location, demonstrating the controller's success. Due to the Gazebo Physics engine and flaws within the robot design, minor drift was observed, but overall, the controller provided stable and reliable autonomous movement.

Team Contributions -

The team collaborated effectively across all aspects of the project while also leading specific tasks:

- **Grayson Gilbert**: Led CAD modeling and URDF export; ensured ROS 2 compatibility.
- **Kunj Golwala** : Added controllers and integrated the LiDAR sensor for visualization.

- **Manas Desai** : Implemented teleoperation mode, published odometry data, and integrated the IMU plugin.
- **Shreya Kalyanaraman** : Developed the proportional controller for autonomous navigation and visualizing the sensor data in RViz.

Links to Resources

- **GitHub Repository**: Github repository
- **Teleoperation Mode Video**: Teleoperation Mode Video
- **Autonomous Mode Video**: Proportional Controller Video (Autonomous Mode)