ENPM 662 – INTRODUCTION TO ROBOT MODELLING

PROJECT 1 – REPORT

Kumara Ritvik Oruganti

UID: 117368963

Email: okritvik@umd.edu

INTRODUCTION:

This project gave a hands-on experience on modelling of a simple four-wheel robot and simulating the same in the Gazebo using the ROS. It helped me understand and create the launch files, xacro integration, generating the URDF from the exporter tool, changing the origin of the lidar and communication with the ROS topics.

INITIAL THOUGHT:

Since both of us in the team are new to the SolidWorks and ROS, we initially made a sample car to check how the ROS controllers, URDF and launch files work. We saw some YouTube videos along with the resources provided on Canvas to see how the config file, URDF and launch files are modified for the simulation.

Link to the YouTube Videos

While thinking of a model for the steering, we concluded that an axle at the front is not a feasible option. Hence decided to have an L-bend for each front wheel controlled with a single value from the ROS. We both worked together on the project but my major contribution is in the ROS part.

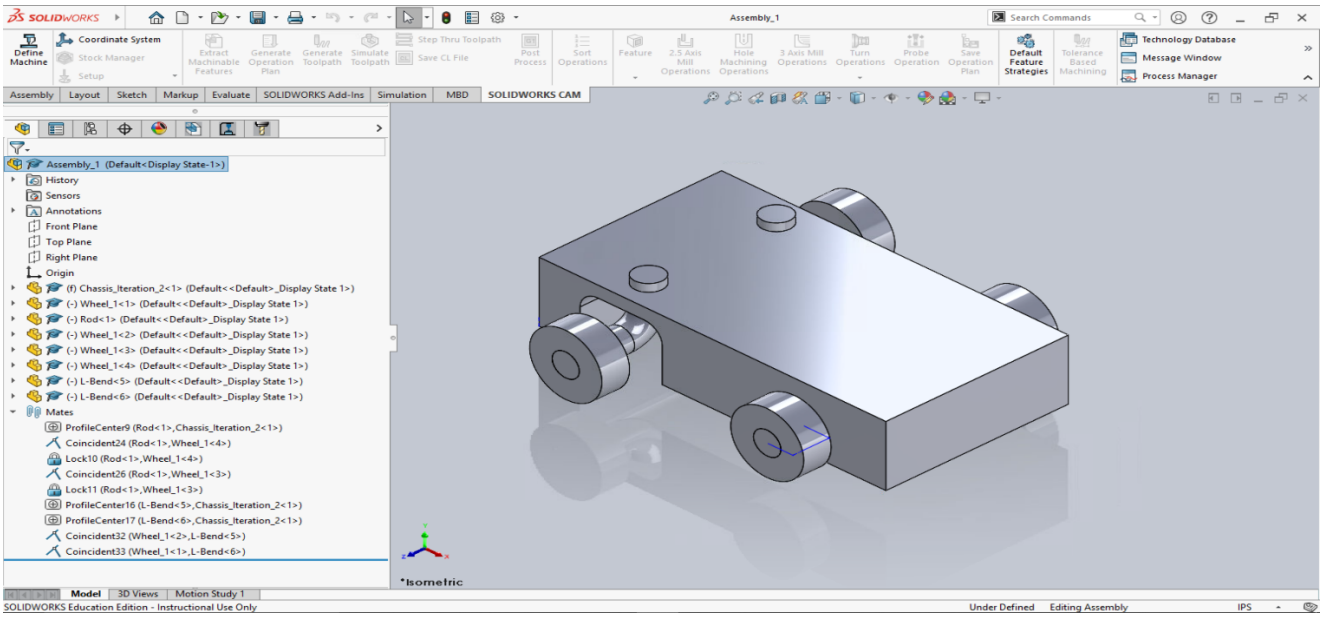

Figure: Sample Car – Initial Model

ERRORS ENCOUNTERED:

1) When the sample car URDF was generated, we selected "automatically generate axis and global origin" option in the tool which made the car's axis flipped by 180 degrees. The problem is because the gazebo axis reference is different from the solidworks axis. The problem is solved by putting the points on each part and adding a reference coordinate system in assembly. This link helped us to add separate axis and coordinate frames for each link.
2) After adding the gazebo plugin in the URDF file, the robot parts converged to origin. After a decent amount of time on figuring out the root cause, this link helped us to point towards the problem in inertia definitions. One of the link in the model was not properly defined by the URDF tool
3) We made the assembly again and followed the step-by-step process for generating the URDF. The same problem was encountered. Hence decided to make a new robot from scratch.
4) Every part was made from scratch but this time we decided to smart dimension each sketch till it becomes black (completely defined, which was not taken care in the sample model). URDF generator gave proper URDF file with all the inertias and mass values.
5) Added the gazebo links and generated a launch file with empty world in gazebo and used velocity controller from effort_controllers which didn't work. Hence looked up at the gazebo tutorial on the ROS control wiki and added velocity_controllers interface for the rear axle joint.
6) Initially the L-Bend was made as revolute joint and we didn't specify the limits for effort, velocity, minimum and maximum angles. So, I asked Ajinkya for help and he suggested to change the limits to some value as revolute joint with 0s in limits tag makes it fixed. Changed the limits accordingly, but the robot exploded after launching the model. Hence, changed the joints to continuous instead of revolute which made the model stable.
7) Checked the angle of rotation for the front wheels by publishing data to the raj_position_controller/command topic so that the wheel doesn't go inside the chassis. The limits were identified as 0.5 (trail and error process).
8) While integrating the lidar, the ydlidar_ros package was not installed in the system. sudo apt-get install <rospackage> was not working because it was showing security key error. The ROS community forgot to update the security key and the systems who had ROS installed before May 2021 had to update the GPG key. Resolved this issue using this link.
9) Moving the Lidar on top of the robot chassis is one of the confusing tasks that I have found so far. I changed the axis of the lidar in its URDF file initially and it was there on the top of the robot but the rviz is displaying the robot as well in visual scan. Then I replaced the lidar file with the original lidar file and changed the axis of the laser_frame between the lidar joint and the base link instead of axis in the lidar urdf. It fixed the issue.
10) The console shows the spawner error and pid gains not specified for the joints but they can be ignored as the model spawned correctly and simulation is working fine with the pid controls specified in the config file.

RESULT:

1) **Link** to Project_1 Teleop
2) **Link** to Project_1 Move in a straight path and circular path using publisher and subscriber nodes