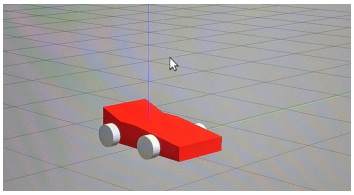
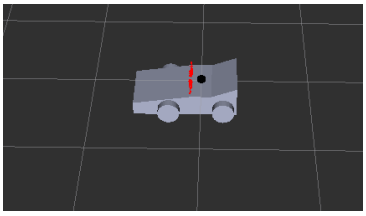
ENPM662 - Introduction to Robot Modelling

Project 1: CAD Modelling & Simulation using Gazebo

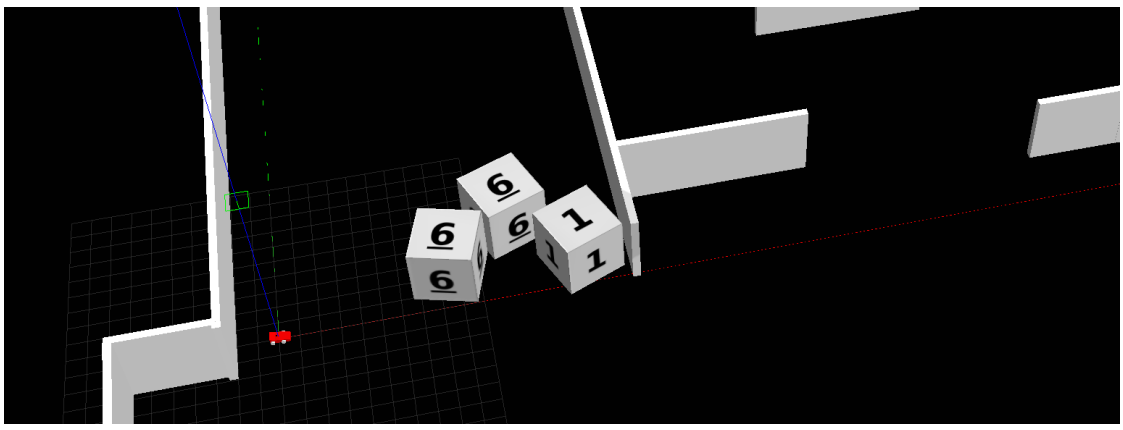
For the project we decided to implement a car simulation with a differential drive system. The project started with building the solid works part. First we created the Car chassis, followed by the front and back wheels. After assembling the parts together, the four wheeled red car was ready and the URDF file was generated. The model looked as seen in the figure alongside. In the first attempt, we tried to create a car with multiple Snap fit joints but later redid it and decided to stick with a single part chassis to reduce complexity and avoid any issues in simulation.



Once the model was ready, we checked the urdf for correctness and built the project in the catkin source folder. The next step was to add a LIDAR sensor on the car. In order to do this a dummy link was added to the main URDF file where the laser sensor can be attached. Next step was to add the ydlidar urdf file in the project and configure it's URDF. After finishing this step we were unable to visualize the laser on RVIZ. This issue was solved after inspecting the URDF and adding missing information about the dummy link name and robot model name which we missed. Once this error was fixed and running the check_urdf command on all URDFs, the next step was to consolidate it to a single xacro file.



Next we updated the launch file by reference to the final urdf with consolidation of car and laser data. After this step, the model sometimes showed anomalous behaviour, few parts were not visible in simulation and during some run the position of all parts was origin which showed a deformed model in rviz and gazebo. After researching and discussing the issue with TA we realized that we need to add constraints to the joints in the model. Doing so resolved the issue. By adding the path of competition_arena to the launch file, we could now visualize the model in the stated world.



To run the car model in gazebo, the next step was to add the controller. For our differentially driven model, we face difficulties implementing the controller. After adding the controller configuration I could see errors on the terminal stating missing controller plugins. This was because I missed the installation of controllers plugins. Once this was resolved, the project threw an error saying the gain values were not stated. This was because the publisher command had incorrect syntax. At this point, rostopic list displayed all the model topics on the terminal correctly. After correctly configuring the teleop file and adding commands for all four wheels, we could control the car using teleop. The video of this simulation can be found here.

The final step was to create a publisher subscriber to publish data on car model topics and automate moving them in Gazebo. This step was completed without any difficulties. First we created the publisher. This file includes codes to create a publisher node and commands to publish data on the simulated car model topics - in this case the wheels. On running this python file, the car moved in a straight line. To view data published, we created a subscriber node and subscribed to its publisher. On running the subscriber python file we could view the data on the terminal that was running the car model on Gazebo. The video for coded car control using this can be found here

A ReadMe file with commands to run the simulation can be found here

My contribution to this project was mainly working on the solidworks model. Creating the individual parts for the car model, exporting and configuring the URDF file for the model. All the errors that we faced during the development of this project were researched and dubugged together by both the members. I wrote the files to automate the simulation of the car model using the publisher and subscriber. I also helped in verifying the generated files, for example: checking errors in URDF, xacro files; checking issues with incorrect arguments and syntax in the launch file for resolving the errors occurred in adding the laser to the car model and configuring the controller.

Required links:
Simulation using Teleop
Simulation using publisher/subscriber
ReadMe file with commands to run project