Lab 2: Introduction to Linux & ROS

RSS Team 8
Kevin Chan, Alexander Galitskiy, Josh Hellerstein,
Trevor Henderson, Caroline Hope, Katy Muhlrad

6.141/16.405J Robotics: Science and Systems

Spring 2017

Submitted: February 22, 2017

1 Introduction

This lab project included two main sections. Lab 2A involved installation of the virtual machine workstation on team computers and an overview to practice running existing ROS modules. This lab required use of command line to run the TurtleSim node. As Lab 2A was completed individually by each team member, it is not discussed in this report. Then Lab 2B asked teams to set up their MIT-direct GitHub accounts, create a team repository, and a team website. Teams also became more familiar with RACECAR simulator by running Gazebo along with ROS to simulate the RACECAR model and take sensor (LIDAR and ZED Camera) data. **[Caroline]**

2 Methodology

2.1 Controlling the Simulation

The first task revolved around the simulation of a RACECAR with a LIDAR sensor in Gazebp. The team started by experimenting with the simulation alone. Messages were able to be directly published to the ROS topic `/vesc/Ackermann_md_mux/input/teleop` with information about the speed and angle of the car.

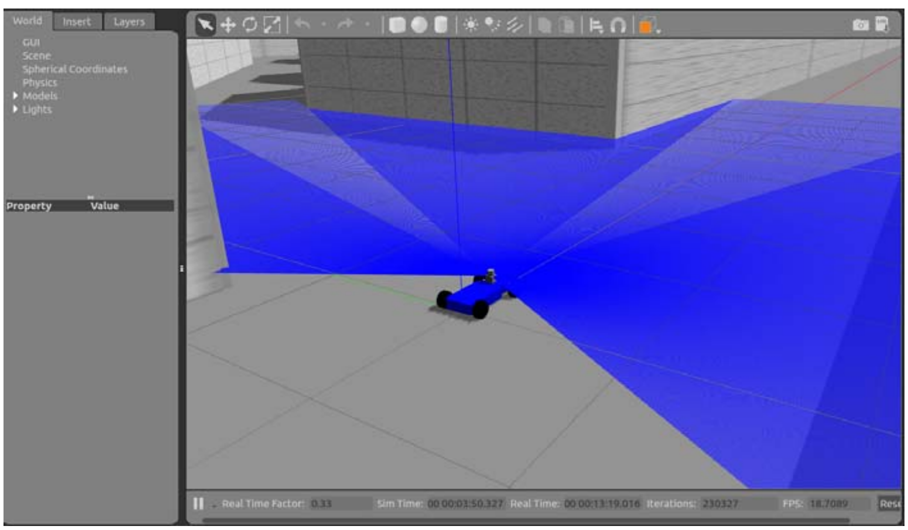

Figure 1. *Image of Gazebo simulator during joystick operation. The blue indicates the region of which the RACECAR is aware, i.e. collecting sensor data.*

After becoming familiar with the simulation, the team made it possible to control the simulated RACECAR via a USB controller. The code producing the output from the joystick originally did not communicate with the virtual car, so a new launch file was created to remap the output of the joystick to the input of the simulation. This new file also launched the teleop.launch file required to tell Gazebo to listen to commands from the joystick.

Finally, RViz was used to see the laser scan data while the RACECAR was controlled with the joystick. Below are is an image of what the car sees with the LIDAR scan when it is located at the same position as the simulation photo above in the map of the tunnels. **[Katy]**

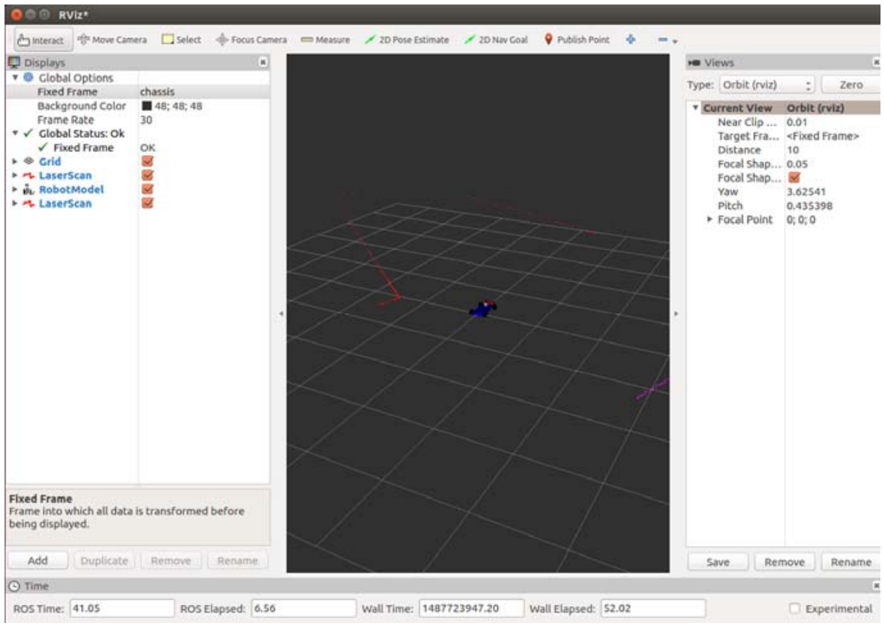

Figure 2. *Image of ground level LIDAR laser scan data in the simulated MIT Tunnels map.*

2.2 Collecting Sensor Data

After becoming familiar with the simulation, the team started working on the real robot. The team successfully view output streams from both the ZED camera and the LIDAR using the commands as described in the lab instructions. Originally it was difficult viewing these streams through the routers's wireless connection (a 60% packet loss rate was observed), but framerate

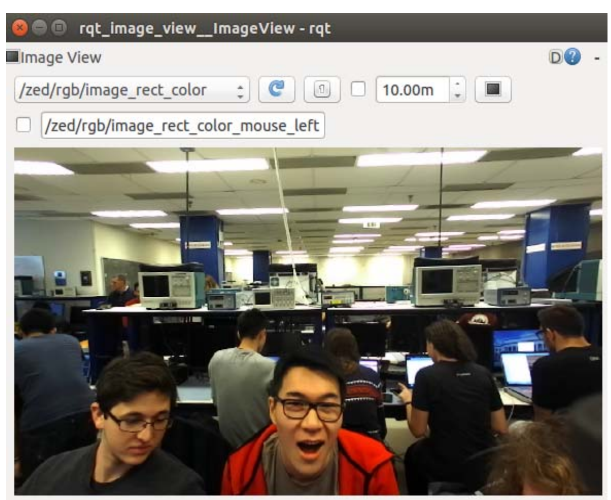

Figure 3. *ZED Camera streaming image of Kevin and Josh's faces.*

was improved when an Ethernet cable was used to connect to the router. Below are screenshots from the ZED camera and the LIDAR outputs.

Other output modes of the ZED were explored, including its depth map mode in order to get a sense of how accurately it estimates depth. Using the bag tool, approximately 2GB of sensor information was successfully recorded to a .bag file. **[Kevin & Josh]**

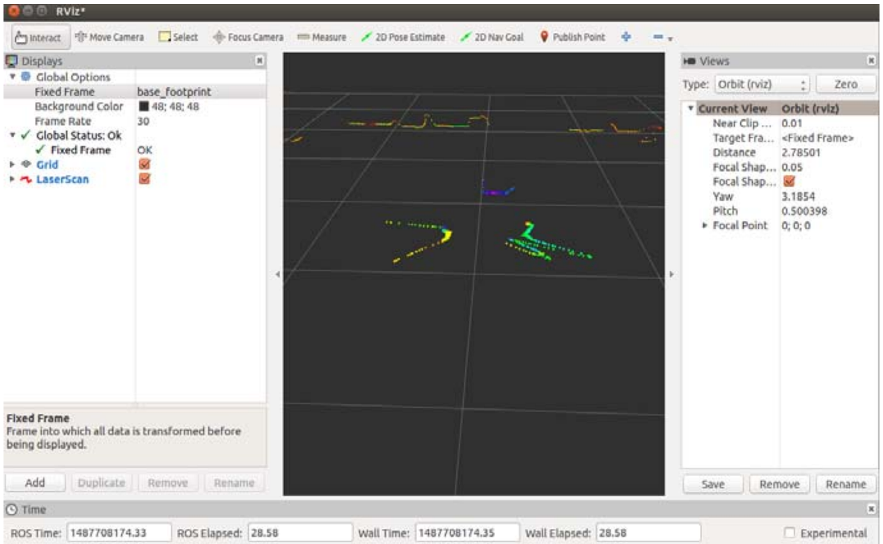

Figure 4. *LIDAR laser scan image of RC car real position, instead of the previously shown image in the simulator. This lines seen in the image correspond to close-proximity and mid-range (1-1.5 meters) instrumentation in the electronics lab as well as team member heads/hands.*

3 Troubleshooting

Since the lab was intended to be relatively straightforward for the purpose of Linux/ROS familiarity, it allowed opportunities to troubleshoot directly and effectively on specific parts of the lab implementation.

Initially the joystick input commands were not registered by the RACECAR simulator, and the first error identified was that the dongle to connect the joystick to the computer running the Gazebo simulator did not match.

The next problem identified and addressed was a spelling mistake in the launch file. Because the launch file created was so short – only four lines of code, two of which are essentially just to indicate that it was a launch file – it was difficult to understand why the joystick inputs were not remapping to the simulator inputs. Node conversation was checked on the rqt computational graph to understand which topics were active and connecting various nodes. An issue that arises with this method is that a spelling mistake in remapping did not raise an error, and the rqt instead just creates another topic with the misspelled identifier that does not connect messages between the intended nodes. Once this spelling error was found, the computational graph was again checked for accuracy, and the joystick was able to send commands to the RACECAR simulation.

Once the team was able to use the car, an error was thrown saying that the config file was missing. In the end it was found that the config file on the car was outdated and an updated file from the internet had to be manually fetched for the current software version. The car's router was hooked to the Ethernet so it would receive an internet connection, and then the team wirelessly SSHed into the car to force it to download a new file. After successfully downloading the file the car's sensors were able to be read.

Another issue that came up when taking data was that the wireless internet connection between the RC car and the VM was slow. It was unable to stream data from the ZED Camera to the computer, and instead displayed a grayscale blur. This issue was mitigated for the time being by utilizing a direct Ethernet connection instead of wireless connection. **[Trevor & Caroline]**

4 Teamwork

On the first communication forum the teams met together to discuss individual and team goals for this class. Goals range from simply becoming more familiar with programming processes to more complex technical skills such as implementation of inference algorithms, exploring machine learning capabilities, or making flame decals for the RC car. This class period also allowed for members to briefly introduce themselves to the team and to discuss what experience they had with the topics of the class. The team has a very wide range of familiarity with robotics because of class/project/work experience, as well as a wide range in programming because there are team members of both Course 6 and Course 16.

Since the beginning of the course, the team has created various ways to organize the work for the coming semester, including a shared Dropbox, Google Drive, and GitHub repository. The team has put together a website in order to display the team biographies, information about the class and equipment, and the lab reports.

Already there has been a lot of teamwork communication displayed by the members of the team by scheduling of meetings and work times, discussing lab processes, helping teammates with technical content as necessary, and troubleshooting. **[Sasha & Caroline]**

5 Conclusion

All objectives of Lab 2 were completed by the team for increased familiarity with ROS modules and the RACECAR simulator. The program deliverables for this lab are the RACECAR simulator launch file and README.md file, and have both been uploaded to the lab2 repository. The current status of the website is that the "About" and "Biography" pages have been completed, with a plan to continue to add information regarding the class and work being done by the team so that it can be useful during the course for administrative staff as well as after completion of the course as a portfolio. **[Sasha]**