ECE4804AMR Lab Report 1

Car and simulator setup

Hong Ray Teo, hteo6@gatech.edu, Fan Han Hoon, fhoon3@gatech.edu, Ethan Tse, etse8@gatech.edu

***Abstract*—This is our report for lab 1 of ECE 4804 Autonomous Mobile Robotics. In this lab, we completed the construction of our autonomous car, worked through a RViz simulation of the car, and recorded and played back a ROS bag of us driving the car in the real world. By the end of this lab, we all felt well introduced to the world of ROS 2 and mobile robotics as a whole.**

I. Introduction

THIS lab introduced us to the basics of robotics with regards to both hardware and software. The lab consists of three parts: assembling the car, setting up a Robot Operating System (ROS) simulator and setting up ROS on the car for teleoperation. It took our team about 4 hours to complete the lab over a span of 2 weeks.

The technologies involved in this lab included screwdrivers for mechanical assembly of the car using screws, wires for electrical connectivity between car components and a laptop for running the simulation as well as communicating with the finished car.

By the end of the lab, we had a working car and simulator along with a better understanding of ROS. The work done in this lab forms the foundation of all future labs, where we will make extensive use of ROS for both simulation and control of the physical robot.

II. Background

Our team worked on this lab in Klaus 2446. We were provided with a ROSMASTER R2 kit along with a build guide. We also had the kind assistance of the TAs, Sid and Matthew, who helped to address any problems we faced.

III. Procedure

Initially, the team was assigned with a Jetson Orin Nano Developer Kit with additional hardware such as 4ROS Lidar, RGB camera and Depth Sensor, Ackerman Steering Structure, LCD display and Power and Driver. Reference assembly guides were also provided by the courtesy of the TAs for ECE 4804AMR and the manufacturer of ROSMASTER R2.

Fig. 1. Photo of Starter Kit for ECE 4804 AMR

Upon completing the assembly, our team had the TAs to verify that the robot could run the prebuilt software and detect all the relevant hardware and expansion board to obtain the checkoffs for the robot assembly. Then, the team had to set up the F1Tenth Simulator software on our machines and demonstrate teleop keyboard control in the simulator. Software such as Docker was used to run the simulator and the Linux platform for ROS 2 packages [1]. After that, the team had to set up SSH connections from their personal machine to the robot for further remote control set-up via Docker to allow containers to gain access to the hardware through the first terminal. In the second terminal, the team will access the active Docker to establish connections to the joystick controller. Eventually, the team also demonstrated the robot driving itself in figure eight by replaying the bag file where it was recorded when the team was manually controlling the robot via the joystick interface in the third terminal.

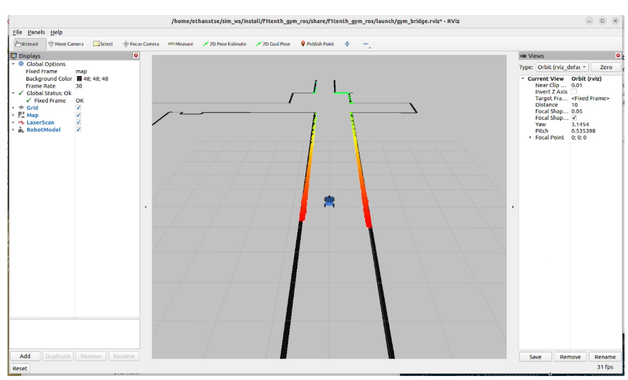

Fig. 2. Photo of Docker Software

Since our team consists of 3 members, Han, Ray and Ethan, each member was allocated a different task. Ray and Han were assigned to work on most of the robot assembly as Ethan was more experienced with the ROS2 platform and assigned to work on the simulation task.

Ray had been assisting in the assembly of the robot and reviewed the entire assembly instructions that were provided by the manufacturer of ROSMASTERS2. He also sorted out the labelled part bags for different sections and worked on cable management. Besides that, he also assisted in the joystick functionality test by moving the robot in Figure eight.

Han was the main assembler of the robot and screwed most of the hardware together on the given chassis. He also reviewed the instructions that were provided by Sid and Matthew to ensure that each hardware was aligned to the designated spot as the cable length provided was limited. Moreover, he also assisted in the initial joystick control testing when Ethan established the connections via SSH.

Ethan was the key member on the software simulations and software setups for the remote control of the robot. Initially, he also assisted in the assembly process and some screws and washers need to be identified for the correct hardware. He also ran the entire robot simulation through his laptop and demonstrated keyboard controls on the robot. Eventually, he also established the SSH connection to the robot and replayed the recorded bin file for Figure 8 movements.

IV. SOFTWARE

While this lab did was primarily hardware focused and did not involve creating our own software packages, we did have to interface with various ROS 2 packages via Linux.

We utilized a modified version of the F1Tenth ROS 2 communication bridge that turns it into a simulation that runs in RViz.

To drive the car, we connected into the car via ssh and started up the yahboom bringup launch file remotely. We then utilized the joystick package to control the car via the controller. Finally, we used ROS bags to record the topic ”/cmd_vel” which is the commanded velocity for the car. By playing back this bag, we were able to get the car to drive a path we previously drove.

V. RESULTS

We were able to achieve the three goals of assembling our car, installing and using the F1Tenth simulator, and driving the car using ROS 2 from both commanded input from a joystick, and from a ROS bag.

Figure 3 shows us in the midsts of assembling the car itself. This is the only place we ran into an issue. When trying to get our car to work, we found that the Jetson could not connect to the Yahboom ROS extension board. After extensive debugging, we determined the issue to be with the physical extension board itself. Since this was not something we could not fix ourselves, the TA let use use their vehicle to finish out the lab.

Figure 4 and figure 5 show an screen shot of the F1Tenth simulator and a photo of the car being driven via ROS 2 respectively. We ran into no issues with these parts of the lab.

Fig. 3. Photo during assembly of our car.

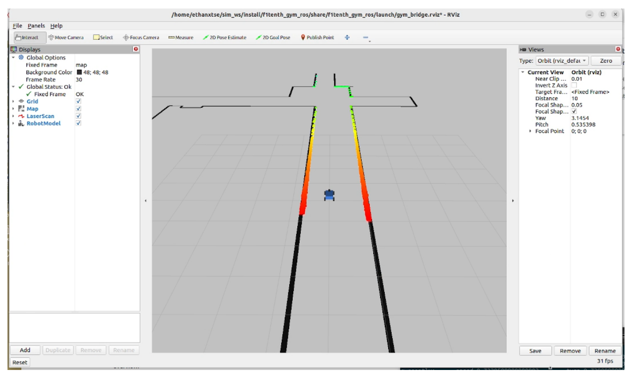

Fig. 4. Photo of the F1Tenth simulation running in RViz.

Fig. 5. Image of car driving with commanded velocities from a previously recorded ROS bag.

VI. Conclusion

The goals of this lab were to assembled the car, install and use the F1Tenth simulator, and drive the car using ROS 2 and record a ROS 2 bag file. We were successfully able to achieve all three of these tasks. During our building phase, we ran into a problem interfacing with our Yahboom ROS 2 expansion board. We determined this was purely a hardware problem with the board itself and the TA let us borrow the TA car to complete the rest of the lab. Ultimately, this lab has provided us with a broad overview on ROS 2, and autonomous mobile robotics as a whole. These are skills we are sure to use in later labs.

Acknowledgment

The authors would like to thank Dr. Michael West, Matthew Woodward, and Sidhant Gulati.

References

[1] F1TENTH gym environment ROS2 communication bridge. Matthew Woodward, GitHub. web site: https://github.gatech.edu/ECE4804-AMR/f1tenth_gym_ros, accessed Feb. 01, 2024.