Lab 5: Turtlebot Mapping

Sasha Belotserkovskaya, avb27@case.edu; Kristina Collins, kvc2@case.edu; Shawn Qian, xxq82@case.edu; Connor Wolfe, cbw36@case.edu

I. Introduction

This assignment presents an example of odometry mapping using a teleoperated Turtlebot. A maze was created with objects in the lab, and the robot piloted around to gather mapping data. Procedures and results are presented here.

II. Testing

A. Demonstration Steps

First, connect to Turtlebot using the instructions in the lab assignment. The Turtlebot acts as the ROS master. Recommended to use Google's DNS for stability. Also, note that connecting to the Turtlebot in one terminal does NOT connect you to it in other terminals. To wit, for deeplearning02:

- **host deeplearning02w 8.8.8.8** Take note of the robot's IP, using Google's DNS.
- **ifconfig** Take note of the computer's IP.
- **ping deeplearning02w 8.8.8.8** Check to ensure that data can be exchanged.
- **source /opt/ros/indigo/setup.bash**
- **export ROS_MASTER_URI = http://deeplearning02w.eecs.cwru.edu:11311** Set the Turtlebot as ROS master.
- **export ROS_IP=XXX.XX.XXX.XX** Use the computer's IP here.
- **source /devel/setup.bash** Switch the source to match the location of the written nodes.

It's wise to verify connection by checking rostopics with the command **rostopic list**. Next, open RVIZ, run the map command, and begin teleoperation:

- **rosrun rviz rviz &**
- **rosrun gmapping slam_gmapping scan map:=/map2 &**
- **rosrun teleop_twist_keyboard teleop_twist_keyboard.py**

In RVIZ, change the map to map2 and the fixed frame to odom. For the map shown in Figure 4, we included the LaserScan topic with red points.

The Turtlebot can then be piloted by teleoperation, as shown in Figure 1, and the map will automatically populate.

When the map is populated to your satisfaction, close the teleoperation screen (CTRL+C), and type the following command to save the map:

- **rosrun map_server map_saver -f baconMap map:=map2**

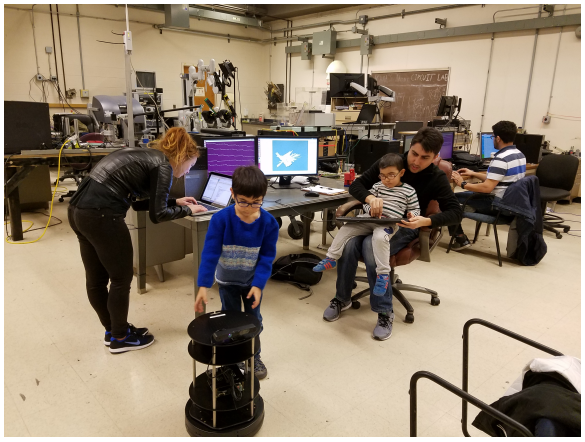

Fig. 1. Turtlebot teleoperation procedure.

Fig. 2. A view of the maze we created for the robot to map.

B. Tips and Notes

It's wise to keep an eye on the Turtlebot and make sure it doesn't bump into anything. The instructions above and in the README recommend using the Google DNS and addressing the turtlebots by IP rather than hostname. We demonstrated our code on deeplearning03. Finally, note that connecting to the Turtlebot in one terminal does *not* create a connection in other terminal windows. For this reason, it's wise to use bash commands (CTRL+Z, B+G, F+G) to run both nodes simultaneously in the same terminal window. (This makes processes a little more difficult to identify and kill when necessary, but still saves time).

III. Testing & Results

We created a "maze" for the robot to travel through, shown in Figure 2 and Figure 3.

Due to the zeal of our young collaborators (See Figure 6), we were not able to record multiple poses with significant

Fig. 3. Another view of the maze.

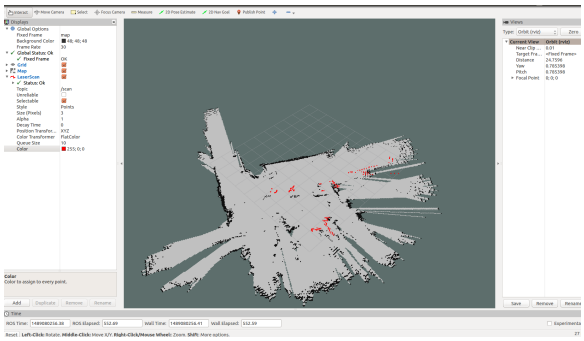

Fig. 4. Mapping in RVIZ.

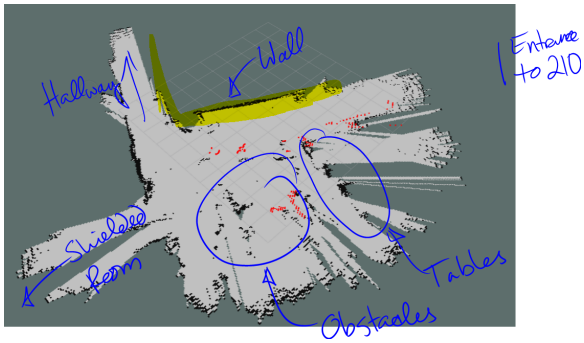

Fig. 5. Map, labeled. The highlighted wall was the most reliable landmark.

rigor. However, features of the maze and the room may be readily identified in the map, as shown in Figure 5. The wall of Glennan 210 made for an excellent landmark. The odometry remained quite consistent and did not seem to accumulate significant error, i.e., the robot could map an area, travel a significant distance across the lab, return to its starting point and remap the area without any visible conflicts in registration.

Our results may be accessed on Github from the following link: https://github.com/shacks2017/teambacon

IV. Conclusion

SLAM mapping with the Turtlebots by this method is fairly robust and good clean fun. One point that could use improvement: Trying to include the robot model in the RVIZ display threw camera calibration errors, so the only way we found to figure out the robot's position relative to obstacles was by inference.

Fig. 6. Our young collaborators.

V. Acknowledgment

We gratefully acknowledge the teleoperating prowess of Armaĝan and Yalin Koyuturk.