16-311 Spring '22 - Lab 8
SLAM and Navigation

Total number of points: 100

OUT: Mar 24, 2022 - DUE: Mar 24, 2022, 1pm

Full Name: _____

Andrew ID: _____

1 Packages to install

Be sure that the following ROS packages are installed in your system. Run first a general update.

- `$ sudo apt-get update`
- `$ sudo apt-get install ros-melodic-dwa-local-planner`
- `$ sudo apt-get install ros-melodic-move-base`
- `$ sudo apt-get install ros-melodic-slam-gmapping`

2 Task 1: Navigation / DWA in Simulation

1. Launch a complex, highly symmetric gazebo world:
   `$ roslaunch turtlebot3_gazebo turtlebot3_world.launch`
2. Launch the `navigation` package:
   `$ roslaunch turtlebot3_navigation turtlebot3_navigation.launch`
   - Which nodes/packages the launch file is launching? Let's go to package and give it a look!
     `$ roscd turtlebot3_navigation`
     `$ cd launch`
     and open the file `turtlebot3_navigation.launch`. Can you explain what's going on there? Which packages / nodes are being executed? Is there a MAP? Where is it located?
3. Launch the `teleop` node:
   `$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`
4. Move the robot around, what do you see? What's going on there?

5. Play with `rviz`, switch components on / off!
6. In `rviz`, set a `2D Pose Estimate` for the robot. What's going on there? Which package is being used?
7. In `rviz`, set a `2D Nav Goal` for the robot. Use the pointer to select a point and hold it for selecting an orientation. The robot shall start 'thinking', a path shall appear, and the robot shall move and hopefully reach the goal pose. Which node / package is making this happening?
8. So far, so good. Now, give a 'wrong' pose estimate and repeat the process. Check what happens.
9. You wonder how / if you can tune the `navigation` behavior? Check the `.yaml` files in the foled `params`. If you edit them with `sudo` you can change the values and see what happens.

   A better option is to copy the entire folder package in your `catkin_ws/src`: your local `turtlebot3_navigation` package will be found first, such that it's the one that will be executed. There, you can make all the changes that you want whithout change any 'system' files.

3 Task 2: SLAM in Simulation

So far we enjoyed the presence of a map, let's use SLAM (based on a Particle filter) to create a map! Shutdown the navigation node.

1. Launch the SLAM nodes:
   ```
   $ roslaunch turtlebot3_slam turtlebot3_slam.launch
   ```
2. What's going on? Move the robot with `teleop`.
3. In `rviz` add `Odometry` to see where you'vebeen moving.
4. Check the active rostopics, anything interesting / new there?
5. Save the current map! (in a location of your choice)
   ```
   $ rosrun map_server map_saver -f ~/map
   ```
6. Open the saved map files. How do they look like? Does it make sense?
7. Let's use the save map for navigation, as in Task 1, but now you pass your own map:
   ```
   $ roslaunch turtlebot3_navigation turtlebot3_navigation.launch
               map_file:=$HOME/map.yaml
   ```
8. Repeat the steps done before.
9. What about understanding and tuning parmeters for mapping?
   ```
   $ roscd turtlebot3_slam
   ```

4 Task 3: SLAM in real world

Follow Dr. Eduardo for repeating the steps with TurtleBot robots in the Lab!