Spring 2018
CIS 693, EEC 693, EEC 793: Autonomous Intelligent Robotics
Assignment 3 (9% of total grade)
Due Date: ~~Wed 2/28 11:59 AM~~ 3/2 Friday 11:59 AM

GOAL

The goal of this assignment is to give students first-hand experience of working on real robots. More specifically, students will learn how to use ROS packages for SLAM, localization, and navigation, and how to create programs to drive the robot to visit specific positions.

For technical questions, please first post on Piazza and leave it there for at least 24 hours. You are indirectly helping other people by doing so. If no one knows the answer, please email the instructor.

ACCESS TO ROBOTS

To complete this assignment, you will need to work on real robots (Turtlebots) that are available in the instructor's lab. Due to the limited number of robots, we have to distribute the resources in a first-come-first-serve manner.

The robots are only used for testing, i.e., software development should be done offline using your own laptop/desktop machines. You may want to host your code in the cloud (github, bitbucket, etc), so that you can work on it anytime anywhere.

Please use the following spreadsheet for reservations. Only one reservation can be made at a time: make another reservation after your current one has been used. A reservation ensures that one specific robot cannot be used for any other purposes.
https://docs.google.com/spreadsheets/d/1mZeSqv3JwE4jkl8DK5LtxG_XCXnvznJX_t4DQ7bNa_g/edit?usp=sharing

Students will use their own account (initial of first name followed by last name, e.g., szhang for the instructor) and password ('turtlebot' by default). It is necessary to change the password ASAP.

INSTRUCTIONS

This assignment can be done using either C++ or Python (no preference). Students will first use ROS package, GMapping, to build a map of the lab, and then do autonomous navigation: let the robot visit the four corner areas of the lab one after another.

For each new terminal, you need to run the following two commands before running anything else:

```
export TURTLEBOT_3D_SENSOR=astra
source /opt/ros/indigo/setup.bash
```

1. Turn on the robot (both computer and robot base).

2. Open a terminal and run the following command to start the robot base driver. The "screen" part is optional and enables more printouts that can be useful for debugging.

    ```
    roslaunch turtlebot_bringup minimal.launch --screen
    ```

3. Open another terminal and run the following command (for gmapping, which is a ROS implementation

of FastSLAM).

```
roslaunch turtlebot_navigation gmapping_demo.launch --screen
```

4. Open another terminal and run the following to start ‘rviz’ for visualization:

```
roslaunch turtlebot_rviz_launchers view_navigation.launch --screen
```

5. Finally, in a new terminal, run the following command for teleoperation.

```
roslaunch turtlebot_teleop keyboard_teleop.launch --screen
```

6. Keeping the teleop terminal to be focused, use the keys of “i, j, m, l” to control the robot moving forward, turning left, moving backward, and turning right.

   Steps 2-6 are demonstrated in this video (ssh is unnecessary):
   https://youtu.be/9efBzYAi1QI

7. After driving the robot around, a map of the lab is expected to be generated and shown in rviz. Run the following command to save the map:

```
rosrun map_server map_saver -f /path/to/your_map
```

8. Now you have a map (that includes a yaml file and a pgm file) ready to be used for autonomous navigation.

   To do that, simply close all your ROS applications, then in a new terminal run the command in Step 2 to start the robot base driver, and run the following command for navigation:

```
roslaunch turtlebot_navigation amcl_demo.launch map_file:=/path/to/your_map.yaml
```

   To visualize the map and robot’s pose, the command in Step 4 needs to be run as well. This video shows the steps of autonomous navigation (ssh is unnecessary):
   https://youtu.be/iGZPWjddans

9. The instructions above can be used for SLAM and autonomous navigation WITHOUT creating your own code.

   In order to let the robot visit the four corners in the lab in order, students will need to create code to send navigation goals to the robot (one goal at a time). More specifically, goal messages (in the ROS type of geometry_msgs/PoseStamped) need to be sent to the ROS topic of /move_base_simple/goal.

WHAT TO TURN IN

Students will submit a link to the YouTube video (in the body of your submission email) that shows the robot autonomously navigates in the lab to visit the four corner areas.

Students also need to turn in a single file (in tar/zip/rar format), and name it as your last name (initial in uppercase) followed by the initial of your first name (uppercase). For example, the file name should be “ZhangS.tar” for the instructor.

Students will need to create a package (that includes two launch files) in such a way that

1. running the following command will make the robot ready for taking an initial position through rviz (students will decide what to be included in the launch files),

   roslaunch ZhangS assign3_bringup.launch

2. initial position of the robot can be assigned through "click and drag" on rviz
3. running the following command will start the robot's autonomous navigation behaviors.

   roslaunch ZhangS assign3_navigation.launch

The instructor may ask students for demonstrations. In that case, please do not take it personally.

Send an email, with your submission in the attachment (less than 2M), to the instructor's Gmail address with subject line: **"class assignment for yymmdd"**, where yymmdd is the due date. You can leave the body of the email empty.

If you have any questions about this assignment, feel free to ask the instructor. Not strictly following this submission instruction will reduce your points by 20%.

Late submissions receive a zero.