EE346 - Mobile Robot Navigation and Control

Fall 2023
Laboratory #4 (4%)
Date: November 8, 2023
Robot Person Following and Homing

Objectives

- Practice how to create a ROS package.
- Understand the logic for a robot to follow a walking person.
- Implement a robot trajectory controller in the homing application.

Procedure

In this lab, in Part I, you will first install and test a TurtleBot3 package that follows a walking person, to verify its operation and learn the logic behind it. Once you feel comfortable to use the package, in Part II, you will create your own package where the robot looks for a pole-like object in the environment and approaches the object until it is within 15cm of the object.

Part I: Robot Person Following

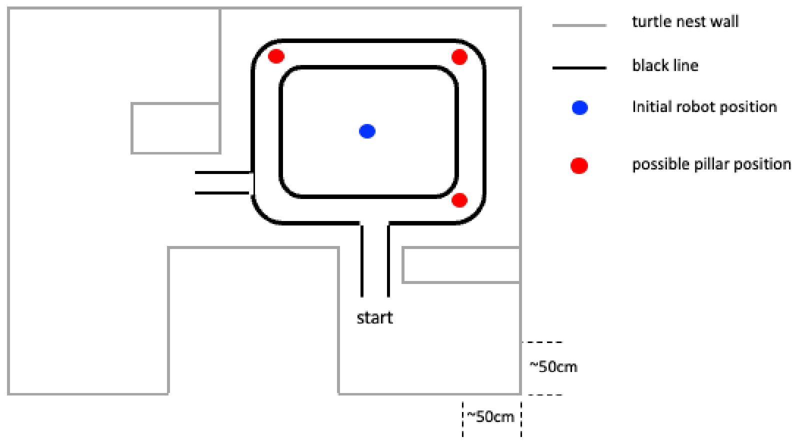
Visit the page at https://emanual.robotis.com/docs/en/platform/turtlebot3/applications/, download and install the package in "1.1 TurtleBot Follower Demo". Recreate the demo shown in the YouTube video so that your TurtleBot3 can follow you as you walk. Please note that this TurtleBot3 demo was developed for the LDS-01 LiDAR whereas the LiDAR we use on our TurtleBot3 is LDS-02. So, some parameters in the current follower package might need to be adjusted. Secondly, as is mentioned in the description of the demo, for the follower to work well, you may need to run it in an open environment. So it might not work as well inside the lab as in the hallway outside of 433.

Once you have succeeded in duplicating the demo, show it to a TA. Submit the revised package as a GitHub link and inform a TA of the link to your GitHub. Instructions on how to set up a GitHub site will follow.

Part II: Homing

In this part of the lab, you will create a package that looks for a pole-like object in the robot environment and approaches the pole once it is found. The location of the pole is not known a priori, and you can assume that only one such object exists in the robot environment and *that it is the closest object to the robot*. You should use the trajectory control algorithm that we have studied in the class and is described in Siegwart's book. Your robot should stop near the pole within 15cm without hitting the pole for the homing task to be considered a success. Tune the gains of the trajectory controller so that you robot can approach the goal position *as quickly as possible*. Your solution can be built by modifying the solution in Part I or from scratch. In the latter case, you can refer to the C++ example at this GitHub site to experiment with a simple pole/pillar detection method (study the "`LaserCallback`" method).

To test your solution, your robot should be placed in the center of the robot environment at an unknown orientation. A pillar is placed somewhere in the open area so that the assumption that it is the closest object to the robot holds true (see figure to the right). As a first step, your robot should detect the direction of the pillar and turn so that its heading is oriented toward the pillar. Once successful, in the second step, your robot should detect and move toward the pillar until it is within 15 cm of the pillar, all along a single smooth trajectory.



Once you have completed the first or second step of this part, demonstrate it to a TA. Record the time it takes for your robot to complete this task. Submit the package by uploading it to a GitHub package (same GitHub as in Part I) and inform the TA of your GitHub link.

Marking

If you are able to complete the lab before the end of the second week, you will receive 2% for Part I and 2% for Part II. If you are not able to complete any parts of the demo within the lecture session, you will get a 20% penalty of the part weight, and an additional 20% for each day of delayed demo (checking by a TA).