CSCE 274 – Robotics Design and Applications – Fall 2019

Project 4 (10% over the final grade)

Assigned: November 16, 2019
Due: December 5, 2019

The purpose of this assignment is to utilize all knowledge you have gained this semester programming the iRobot Create.

Instructions

Please read carefully the following tasks and program the robot accordingly. *You should do this assignment in the groups that have been assigned.*

The instructions on how to use the robot and how to submit the project assignment is on the webpage of the class, at the following links:
https://drive.google.com/file/d/1Z4sSVl08r5S98r2kKA0xmwTWbPnh8d63/view
https://drive.google.com/file/d/1JRj0926o3-jKDnJP5l_T_xM4pE0sSSgy/view

Tasks

Task 1. Augment the *interface* written for Projects 1, 2, and 3 including the detection of the docking station.

Task 2. Write a *program* that utilizes the augmented interface in the previous task and:

a. Initializes the robot, by setting it in *passive* and *safe* mode (already done in Project 1).
b. If the robot is stopped, once the *clean/power* button is pressed (already done in Project 01), initializes a wall following behavior (done in Project 3).
c. While the robot is moving, if it finds the dock, it will approach it and dock to its home base. When the docking is complete, the robot should stop, make a happy sound, and terminate properly the program. NOTE: you should **NOT** use the Dock command available on the robot.
d. If the robot is moving, when the clean/power button is pressed, stop the robot wherever it is (already done in Project 1).
e. Note that the program should continue listening for button presses when the robot is not docked, until the program is terminated (already done in Project 1).

Task 3

a. [Extra Credit] Process the log file in order to plot the position of the robot in a 2D graph, e.g., using matplotlib (http://matplotlib.org/users/pyplot_tutorial.html). To plot the position of the robot correctly, remember that the Create 2 is a differential drive robot and that the motion you perform are forward motion and rotation in place. Note that this code should be separate from the robot code and should be executed on your laptop by reading the log file created in step (g).
b. [Extra credit] Create an occupancy grid map, where the size of each cell is the size of the robot. The grid map can be arbitrarily initialized and the robot can be arbitrarily placed in the map. Each cell can be Occupied, Unoccupied, and Unknown Cells. You can use the distance and angle information for localizing the robot (as done for extra credits in Project 02), and bumpers and/or

wall sensors to detect obstacles. This information should be logged (similarly to the extra credit for Project 02), so that you can process the information offline and create the map. As a reference for working with images, you can use for example http://matplotlib.org/users/image_tutorial.html, but any other library that deals with images is fine (e.g., OpenCV).

Comments

Your programs will be evaluated based on both their functionality and their coding style. In the notes for iRobot Create 2 you can find an informal style guide to help give you an idea of what is expected together with the coding style that you should follow. Task 3 assumes that logging information from Project 2 is implemented already.

Evaluation

The solution will be evaluated according to the following.

Task 1 functionality (10):

- Correct reading and interpretation of docking data.

Task 2 functionality (70):

- Robot takes appropriate action when just the red beam is detected.
- Robot takes appropriate action when just the green beam is detected.
- Robot takes appropriate action when both beams are detected.
- Robot docks.
- Robot is actively charging once it docks.

Style (10):

- No duplication of executable code?
- No magic numbers?
- Names match functionality?
- Adequate comments?
- Comments match code?
- Consistent formatting?

Documentation (10):

- Report is complete and clear?
- Required sections exist?
- Free of typos and grammatical errors?
- Describing docking task in detail?

Extra credit (Task 3) (40):

- Correct logging of the file.
- Plotting of the trajectory.
- Occupancy grid correctly written and resembling environment from trajectory of the robot.