COSC81/181 – Robotics – Fall 2019

Programming Assignment 2 (10% over the final grade)

The purpose of this assignment is to introduce you to a feedback controller for following a wall.

Instructions

Please read carefully the following tasks and write the program(s) accordingly. *You should do this assignment individually.*

The instructions on how to submit the project assignment is on Canvas, at the following link

https://canvas.dartmouth.edu/files/5616794/download?download_frd=1
Please look also at the general guidelines for writing code:

https://canvas.dartmouth.edu/files/5585034/download?download_frd=1

The Task

Assuming that the robot is deployed in an indoor environment (see the world file provided for Stage), write a simple ROS node in your catkin workspace (see http://wiki.ros.org/ROS/Tutorials/CreatingPackage) that implements a PD controller that follows the right wall at a given distance, using the laser sensor. (Grad students should make the robot follow the closest wall, that could be on the left or on the right.) A finite state machine should govern the different behaviors of the robot. Explain how the gains have been set. Note that also both left and right turns should be handled. When there is an obstacle in front, stop the robot. Be sure to describe the decisions you made to handle these cases in the report.

To run the Stage world, the following command should be run, in the folder where the archive is extracted:

```
rosrun stage_ros stageros -d 2017-02-11-00-31-57.world
```

Comments

Please see the course website for details on how to submit your project. Include screenshots of different behavior of the robot with the different gains you tested – in the Stage window you can click on "View->Footprints" so that the trajectory followed by the robot is visible. Since you'll do this project on your own, the "Allocation of Effort" part of your report should be very easy to write.

Evaluation

Your programs will be evaluated based on both their functionality and their coding style. In the notes for writing programs, 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.

ROS usage (10):

- Submitted file contains a well-formed ROS package.
- Package is named correctly.
- Package dependencies are correct.
- Package is configured correctly to build executable.

ROS Correctness (10):

- Becomes a ROS node correctly.
- Subscribes to correct topic and processes callbacks appropriately.
- Publishes to correct topic.
- Publishes messages of the correct type.

Task (50):

- Correct behavior and implementation.
- Work with left and right turns.

Style (10):

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

Documentation (20):

- Report is complete and clear.
- Required sections exist under readily identifiable headings.
- Evaluation section contains a description and figures on how the gains were tuned.
- Free of typos and grammatical errors.