

CS 8803 – O01: Artificial Intelligence for Robotics

Project 3 : SLAM Warehouse

Fall 2017

Project Description

The goal of Project 3 is to give you practice integrating all aspects of the course into a robot control system. As a continuation of project 2, you will write a planner to guide a robot through a warehouse to retrieve and deliver boxes, but in this project you will not be given a map of the warehouse. Instead, your robot will have (noisy) sensors and must use them to navigate around the warehouse. While you do not necessarily need to attempt to build a map of the warehouse using the data from the sensors, using some form of SLAM will make your path planning much easier and result in a lower cost for delivering boxes.

Submitting your Assignment

The specifications for your robot are described within the `partC.py` file. You should complete the file to satisfy the specifications described therein, and then you should submit the `partC.py` file to the Project 3 Assignment on T-Square. Do not combine the file into an archive (zip,tar, etc) , and ensure it remains named `partC.py`. Files that do not follow this format will receive a penalty of 25%.

The submission must consist of a Python 2.7 file labeled `partC.py`. Optionally, you may use external modules such as `numpy`, `scipy`, etc that are present in the Udacity Runaway Robot autograder.

Your python file must execute NO code when they are imported. We encourage you to keep any testing code in a separate file that you do not submit. They should also **NOT** display a GUI or Visualization when we import them or call your function under test. If we have to manually edit your code to comment out your own testing harness or visualization you will receive a 25% penalty. You are encouraged to use the “verbose” flag set by the testing suite to enable/disable any printouts your code makes, as running without print statements greatly speeds up your code and will allow you to do more before hitting the 10 second timeout.

Due Date:

This assignment is due Monday, April 23rd, Midnight (Anywhere On Earth). We strongly suggest that you complete the assignment at least a few days early, giving you ample time to test it and submit it early.

Testing Your Code

We have provided a testing suite similar to the one we’ll be using for grading the project, which you can use to ensure your code is working correctly. The included test cases are NOT complete, and you will need to develop other, more complicated, test cases to fully validate your code. We encourage you to share your test cases (only) with other students on Piazza.

You should ensure that your code consistently succeeds on each of the given test cases as well as on a wide range of other test cases of your own design, as we will only run your code once per graded test

case. For each test case, your code should complete execution within the proscribed time limit (10 seconds). You may partial credit for any boxes that are delivered before the 10 second timeout.

Grading

You will be graded based upon the percentage of boxes you are able to deliver in each test case. Each test case is weighted equally, even though they have differing numbers of boxes. Note that the test suite keeps track of the “cost” (how many moves it takes to deliver the boxes) and reports that statistic to you for purposes of improving your code, but if you deliver all boxes you will receive full credit regardless of the “cost”.

Academic Integrity

When completing this assignment, you must use ONLY the specified API functions within the partC.py file. Specifically, you may not retrieve data from the testing suite such as the “true” state of the warehouse using any method other than the simulated measurement data provided to the process_measurement method. Student code accessing data structures within the testing suite will be considered a case of academic dishonesty.

You must write the code for this project alone. While you may make limited usage of outside resources, keep in mind that you must cite any such resources you use in your work (for example, you should use comments to denote a snippet of code obtained from StackOverflow).

You must not use anybody else’s code for this project in your work. We will use code-similarity detection software to identify suspicious code, and we will refer any potential incidents to the Office of Student Integrity for investigation. Moreover, you must not post your work on a publicly accessible repository; this could also result in an Honor Code violation [if another student turns in your code]. (Consider using the GT provided Github repository or a repo such as Bitbucket that doesn't default to public sharing.)