AA 274A: Principles of Robot Autonomy I
“HW5”: Final Project Check-In
Due: Friday, November 22nd

“Homework 5” is in quotes because, although this is listed as a homework assignment, it is really an introduction to the final project. The goal of this problem set is to get you started on the final project.

1 Finite State Machine

Please read the posted final project overview, it details the problem you’ll be solving for your final project this quarter.

To help kickstart your project, please provide a diagram of the Finite State Machine you plan to use to control your robot’s high-level behavior.

2 Extensions

This problem is very open-ended, and is designed to make you start thinking more about the final project and what you are capable of as a roboticist! As a team, your job is to write your own node that adds a little character to your robot, and makes it stand out from the crowd.

Here are some possible ideas:

- A filter that uses the pointcloud to track an object moving around the turtlebot
- Identifies a ”puddle” on the ground and moves around it
- Actually stop at a stop sign (or some other landmark) even though it is no longer in view
- Provide an accurate distance estimate of a detected object using pointcloud data
- Accurate position estimate of detected objects
- Follow a moving object (bring in a real cat or dog!!)
- Autonomously explore an unknown environment and build the map as much as possible
- Line-following
- Improve upon any of the existing `asl_turtlebot` functionality
- Implement an algorithm that was covered in class, but not in the homework
- Implement a fancy cool algorithm you heard about from another class

Be creative, leverage visual aids (RViz and create your own props!), make use of all the tools available (RViz, TF trees, Gazebo, and much more that we have not talked about in this class!), and any tools and existing packages you find on the internet is fair game (just make sure you cite them)! You are welcome and encouraged to build upon code from previous homeworks.

For this check-in, please describe what custom functionality your team will implement. Describe how you'll make your robot unique!

In terms of workflow and reproducibility, we recommend creating a repo for your team which includes core files (e.g., `asl_turtlebot`) and your team's unique files that rely on these core files. The idea is that someone can clone your repo, and very quickly see it work on their own TurtleBot. If your team uses a repo for this problem, please share the link.

This is a group submission, so please add your groupmates to the gradescope submission.