

Reactive Methods Lab

Instructor: Rahul Mangharam

Name: Student name(s), *PennId:* PennId(s)

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- All sources of material must be cited. The University Academic Code of Conduct will be strictly enforced.

Goals and Learning Outcomes: The following fundamentals should be understood by the students upon completion of this lab:

- Reactive methods for obstacle avoidance

1 Overview

In this lab, you will implement a reactive algorithm for obstacle avoidance. While the base starter code defines an implementation of the F1/10 Follow the Gap Algorithm, you are allowed to submit in C++, and encouraged to try different reactive algorithms or a combination of several. In total, the python code for the algorithm is only about 120 lines.

2 Review of F1/10 Follow the Gap

The lecture slides on F1/10 Follow the gap is the best visual resource for understanding every step of the algorithm. However, the steps are outlined over here:

1. Obtain laser scans and preprocess them
2. Find the closest point in the LiDAR ranges array
3. Draw a safety bubble around this closest point and set all points inside this bubble to 0. All other non-zero points are now considered 'gaps' or 'free space'
4. Find the max length 'gap', in other words, the largest number of consecutive non-zero elements in your ranges array
5. Find the best goal point in this gap. Naively, this could be the furthest point away in your gap, but you can probably go faster if you follow the 'Better Idea' method as described in lecture.
6. Actuate the car to move towards this goal point by publishing an AckermannDriveStamped to the '/nav' topic

3 Instructions

Implement a reactive algorithm to make the car drive autonomously around the Levine Hall map. You are free to implement any reactive algorithm you want, but the skeleton code is for the F1/10 follow the gap algorithm in lecture. You can implement this node in either C++ or Python but the skeleton code is only in Python. You can download it from <https://github.com/mlab-upenn/f110-fall2019-skeletons>. You will only have to edit `reactive_gap_follow.py`. We will also be releasing a test map with obstacles for you to evaluate it on.

4 Deliverable

Submit the following on canvas by the due date:

- Your package including the reactive_gap_follow.py node, names as {student_name}_reactive.zip (ensure compilation)
- Make a youtube video of your reactive method around the Levine Loop in the simulator. You will also need to make a video of it making its way around the custom map we will provide to you. Add this link to this video(s) your zip file

5 Extra Resources

UNC Follow the Gap Video: <https://youtu.be/ctTJHueaTcY>