F1TENTH Autonomous Racing **(Due Date:)**

Lab 4: Follow the Gap

Instructor: INSTRUCTOR *Name:* STUDENT NAME, *StudentID:* ID

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.

1 Learning outcomes

The following fundamentals should be understood by the students upon completion of this lab:

- Reactive methods for obstacle avoidance

2 Overview

In this lab, you will implement a reactive algorithm for obstacle avoidance. While the base starter code defines an implementation of the F1TENTH 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.

3 Review of F1TENTH Follow the Gap

The lecture slides on F1TENTH 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

4 Implementation

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 F1TENTH 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/f1tenth/f110_ros. You will only have to edit `reactive_gap_follow.py`. There is also a test map (`levine_blocked.pgm`) for you to evaluate on.

5 Deliverables and Submission

Submit the following as `studentname_lab4.zip` (replace studentname with your name):

1. Your package named `studentname_reactive.zip` including the `reactive_gap_follow.py` node. **Make sure it compiles before you submit after changing the package name.**
2. Make a youtube video of your reactive method around the Levine Loop in the simulator. Add this link to a text file named `studentname_lab4_video.txt`
3. Make a youtube video of it making its way around the custom map we will provide to you. Add this link to the same text file named `studentname_lab4_video.txt`

6 Grading

We will test your code by accelerating the car down a straight towards a wall in the Levine map, and your safety node should stop the car before collision.

6.1 Rubric

Topics	**Points**
Compilation	10
Implemented Find-Max Gap	40
Implemented Find best point	30
Levine Video	10
Custom Map Video	10
Total	100

7 Extra Resources

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