**Lab 3. Reactive Methods and Race 2**

Due date: Friday April 17, 2020 on Course Site

Project Lead: submit codes and peer evaluation form

Subscriber: submit lab report and peer evaluation form

**Major Tasks:**

1. Review and reinforce reactive methods for obstacle avoidance;
2. Design your own gap following algorithm and implement it in the f110 simulator;
3. Test your algorithm on race tracks and fine tune parameters for RACE 2.

**Requirements:**

1. Make sure your code include the author names, course and lab numbers, date and other related information in the comments. If you use code or code segments found from the internet, then clearly refer to the source of the code. By having your names in the comments, you agree that you have put effort in developing the code. Your codes shall follow the Python-ROS coding style guide: <http://wiki.ros.org/PyStyleGuide>
2. Do not change the parameters in the simulator as those are designed to match the real f110 car. However, you may change the structure provided in the template .py file to fit your algorithm.
3. No planning algorithms are used in Lab 3 or Race 2. Only reactive method is allowed.

**Lab 3. Reactive Method:**

* 1. A ROS package is provided in the f110\_ros/reactive\_methods folder and we are going to modify the template python code in the /src subfolder. Rename the python file to team#GapFollow.py and place it in the /scripts subfolder instead of the /src subfolder. Implement your gap following algorithm by a new node called team#GapFollow\_node.
  2. Download the map files provided for the race. Test the new node in the simulator with the provided maps and show your implementation to the TAs.
  3. Your package and gap following node will be tested on TX2 with a similar race track to determine your race time and ranking. We will use timer.py provided in Lab 2 to time the race. We will run the navigation mode under the simulator and your code

Your report for Lab 3 and Race 2 shall include answers to the following questions:

Q1. What is obstacle avoidance or collision avoidance? How is obstacle avoidance different from Automatic Emergency Breaking?

Q2. What is a reactive method? What is a deliberative method? What are the differences between the reactive and deliberative methods?

Q3. What are the details of your gap following algorithms? If you tried more than one algorithm, provide comparisons between the differnet algorithms.

Q4. How is your gap following method different from your wall following method? How does your wall following method perform in the two maps with obstacles?

Q5. How do you tune your parameters for Race 2? What is your race performance before and after tuning the parameters?

Q6. What difficulties and problems did you encounter in Lab 3? What did you try to solve them? What did you learn from this experience?

Code submission for Lab 3 and Race 2 shall include: the team#GapFollow.py file and any (optional) video that you make for Race 2.

**Race 2 Grading scheme:**

**Your ROS node will be tested in a different map than the ones provided to you. The test map will be similar to the test maps in terms of length, density of obstacles, and sizes of obstacles, etc.**

* 1. Simulator runs successfully with race track maps 🡪 20 points (show evidence in your report)
  2. Finish 3 laps in 120 seconds without collision 🡪 50 points
  3. Finish additional laps in 120 seconds without collision 🡪 each additional lap adds 2.5 points
  4. Ranking in the race (if a team does not finish the race consistently, then it earns 0 points)

Shortest time to finish 3 laps: race winner, earn 10 points

Longest time to finish 3 laps: ranks #10 and earns 1 point.

* 1. Quality of report: 60 points. Note this also includes the quality of the program, algorithm design, comments, etc.
  2. Peer evaluation forms, lab attendance, and TA observations will be used to adjust individual weights of the team members.