



Programming Robots with ROS

Rally Race!! – Lab 5 ECET 581 – Turn-In Date: Dec 14, 2017

Race Day: Dec 12

The goal of this lab assignment is to put it all together to race the Rally Cars against the clock. We have explored the basic problems of PD control for line following and for path planning for a known map. We used AMCL in ROS and the Hokuyo LIDAR to localize ourselves in a known map. This assignment is intended for your groups of three, but feel free to work as you want. The primary “new” activity in this lab (besides improving the robustness and reliability of the code) is the addition of interpolation of Rally Car position based on IMU data.

In Python (or C if preferred):

1. Several maps have been provided on the website. KNOY_Speedway is the main race course. KNOY_DemoTrack is the triangle in the main atrium around the elevator. Load the test map from website, with waypoints, and use `amcl` and your local controllers to follow the waypoints around the course, transferring to the next waypoint, as appropriate. To get maximum points, you must read the IMU and interpolate the car's position between AMCL updates. Turn on the IMU data with the serial command: `IMU1`. Should you want to turn off the IMU data, use command `IMU0`. (Probably not necessary, but you will have to continuously read the serial input data to prevent a buffer overflow.)
2. Bonus points for using `move_base` and the navigation stack (check the ROS wiki) or for implementing obstacle avoidance for head-to-head racing using a webcam or other sensor information.

To read the serial data from the IMU, see [IMU serial description](#)

Practice in the morning, as needed. Time Trials begin at 11:00 am. Each team gets three official attempts at best time

