ROB521: “Lab” #4 – Mapping and Planning in 2D

1. **Autonomous Drone Racing – Mapping and planning in 2D:** In this “lab”, we will define a combined obstacle detection and planning strategy for autonomous drone racers. Since these vehicles have reliable velocity, attitude and altitude stabilization, we can rely on velocity inputs to the vehicle to control its motion. Similarly, we will rely on high accuracy onboard positioning from RTK-GPS and inertial measurement units, and therefore simply assume that the true drone pose is available.

Figure 1: Drone Racing.

a) The multirotor racers can be modeled in 2D with velocity inputs in the body x and y directions as well as a heading input (direct control of heading). Disturbances in the inertial frame will affect the position and are normally distributed with covariances of $\sigma_{xy} = (0.05)^2$ m^2, and will affect heading with covariance $\sigma_\theta = (0.02)^2$ rad^2. Define a motion model for the vehicle with three states in the inertial frame and be sure to rotate the body velocity inputs correctly. Use an update rate of 5 Hz. For safety reasons, set vehicle input command limits so that the vehicle does not exceed a maximum velocity in any direction of $v_{max} = 20$ *m/s*, which corresponds to a maximum speed of 72 kph.

b) The racers will rely on an Intel RealSense 3D depth and image sensor mounted on a stabilizing gimbal to detect the free space in front of them while moving around the course. The gimbal will ensure that the sensor rig is held parallel to the ground during the race. The RealSense has a field of view that is 69° wide, a maximum range of 10 m and a minimum range of 0.3 m. The update rate is up to 30 Hz and the resolution is 1280X720, but we will use 128 depth pixels in the horizontal plane at 5 Hz for the exam (every 10th pixel along the image center line). The range measurement is affected by Gaussian noise of covariance $\sigma_r=(0.05)^2\ m^2$. Define an inverse measurement model for the simplified 2D RealSense to be used throughout this question.

c) Assume perfect knowledge of the robot position and orientation (these high cost racers are using RTK-GPS and weapons-grade IMUs). Define an algorithm for online mapping and motion planning that uses trajectory rollout to find its way to the half-way checkpoint and then return to the to within 50 cm of the start/finish point. Be sure to consider how you select what direction to point the vehicle and its sensor, what speed to fly at, and how to navigate around the course without prior knowledge of the course map.

d) Implement the algorithm defined above and fly a single circuit of the course, defined in racing.m and depicted in Figure 2, from the start (red circle) through the checkpoint (green line) and back to the start point. Use only the currently constructed map for planning, and note that the overall map size is 87.6 X 67.6 m, and is defined in 0.1 m grid cells. Show your map and path at six evenly spaced points in time around the loop.

e) After the first lap, how could you modify your planning algorithm to improve your lap times?

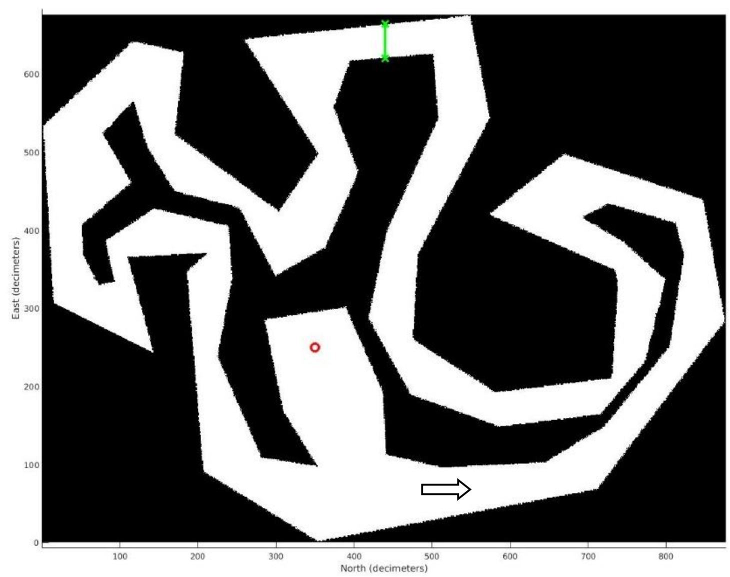

Figure 2: Racing course map, start/finish (red circle) and checkpoint line (green line with Xs).

Submission: Please create a PDF with the six evenly spaced points of map creation and vehicle trajectory and include the time it took to complete your first lap. If you are particularly proud of your solution, feel free to include a discussion of how you did it, and the code as an appendix for posterity, but this is not necessary.