MTE 544: Autonomous Mobile Robotics

Final Exam

December 16th, 9:00 AM – December 17th, 9:00 AM, 2017

Prof. Steven Waslander

Exam length: 24 hours. Submit your completed work on UW-LEARN as a single PDF solution file and a single code zip archive containing three folders labelled Q1, Q2, Q3. Handwritten solutions are acceptable, but please include scans or photos in the submitted pdf file that are legible.

Number of exam pages, including this one: **6**

You are allowed to use any resources available to you, including course notes, course code, textbooks, the internet etc. However, you are not allowed to communicate with any other human being about the exam or its questions at any point in the period of time you are taking the exam. Please also refrain from talking to anyone about this exam who has not yet completed it, or passing on questions or solutions to anyone until the exam is over. There is one exception to this rule, you may email me (stevenw@uwaterloo.ca) or phone in questions (cell: 647 760-6702) at any time, and I will try to respond as soon as possible. Keep in mind I may not be available at all hours, so be sure to read through and/or start all the questions before it gets too late for me to respond.

The exam is marked out of 100. Marks are indicated as [x] at the start of each question. Show all your work, and be sure to at least outline answers to all questions.

Good luck!

1. **Autonomous Drone Racing – Mapping and planning in 2D [30]:** In this question, 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. 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 full RealSense and for the simplified 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 explain 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 through the checkpoint 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?

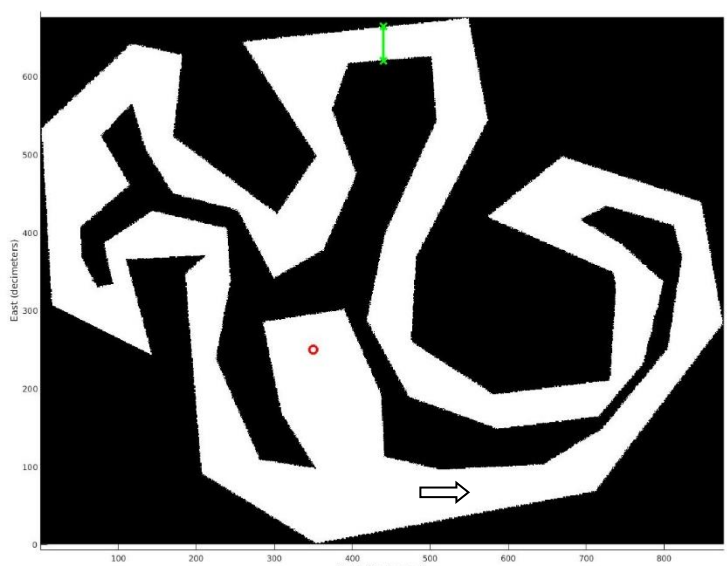

Figure 2: Racing course map, start/finish (red circle) and checkpoint line (green with Xs).

2. **Visibility Graphs and optimal RRTs [30]**. Visibility graphs provide the shortest path through any polygonal environment, but at the cost of prohibitive computational complexity if many vertices exist in the environment. RRTs can rapidly explore constrained space but don't provide efficient paths. In this question, we will search for methods to draw the two approaches closer together.

   a. For the environment in Figure 3 which is generated using planningEnv.m, find the shortest path from start to end using a visibility graph. Describe how the algorithm works using pseudocode, and present a plot of the full visibility graph and the resulting shortest path. What is the final path length?

b. Define the RRT algorithm using pseudocode, then run the algorithm on the same environment for a holonomic vehicle without any motion constraints.
c. Identify three methods to refine the RRT path until it approaches the shortest path solution using additional random sampling and adjusting connections in the RRT tree, without reverting to including the full visibility graph solution. It is not necessary to maintain a tree structure.
d. Implement your favorite solution and present the refined graph and resulting path once it is within 5% of the optimal solution. What were the runtimes for the visibility graph, the RRT and the refined-RRT methods you created?
e. What is the path length of the best solution you can achieve in under 2 minutes runtime, and do you expect your method to converge to the true optimal solution eventually?

Please make all plots large and ensure the graph edges and final path are visible in the final submission.

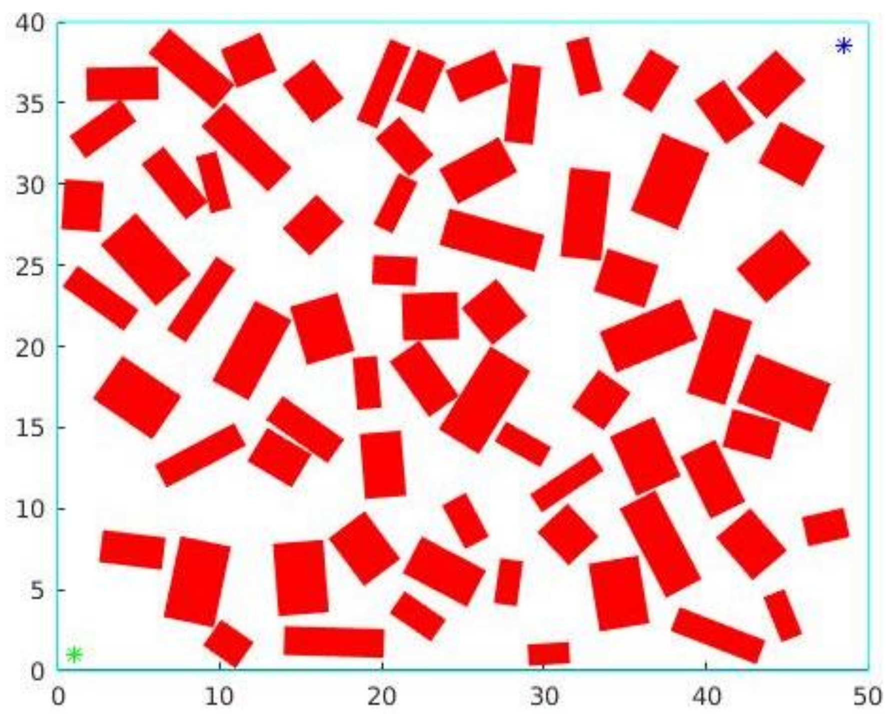

Figure 3: Environment for Question 2.

3. **Semi-SLAM for Warehouse Reconstruction [40]:** A ground rover is required to completely survey an industrial warehouse for 3D reconstruction. A blueprint outline for the warehouse is given, which defines the main obstacles on the floor, a path for the ground vehicle to follow to

complete the reconstruction, and an initial position for the rover to start from. The true initial position of the rover is not known accurately, but the operators attempt to place the rover at a known spot relative to the warehouse map, defined by the green circle in Figure 2. The prior uncertainty can be modeled as additive Gaussian noise about the mean, x0 = [10 40] m in blueprint coordinates, with covariance of 4 m^2 in the x and y axes. The operators have placed 200 uniquely identifiable fiducial markers on the ceiling detectable by the rover for visual localization, and the fiducial marker locations are mostly unknown. However, four known markers have been placed on the inside edges of the four main roof pylons, as depicted in Figure 4. This is semi-SLAM in that some of the features and all of the obstacle locations are known in advance, but the remainder of the features must be mapped while localizing the rover.

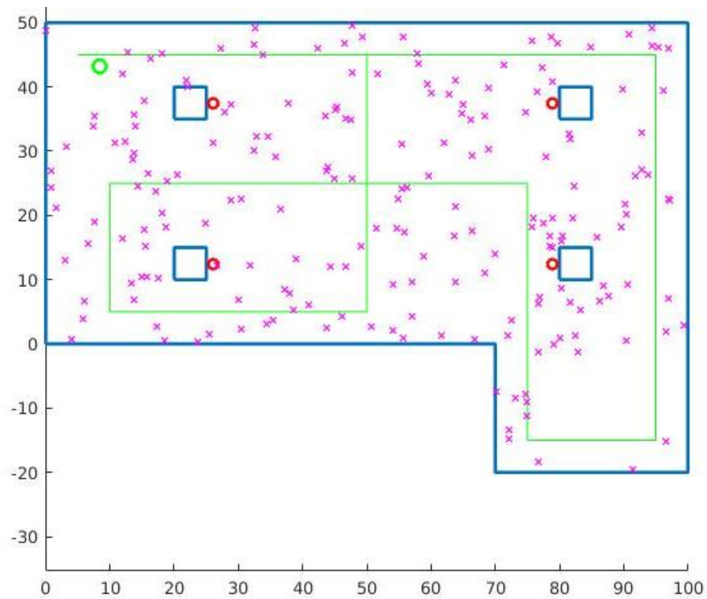

Figure 4: Warehouse blueprint in blue, rover path in green, known fiducial markers in red and unknown fiducials in magenta.

a) With a roof height of 10m, an upward camera on the rover can detect fiducials at 2 Hz over a 20m X 15m area, except when occluded by structure of the warehouse. We will assume that the camera has been properly calibrated and that the height is known and fixed, so that a known transformation from pixel location to 2D position is available. Define a measurement simulation function that returns the (x, y) position of a fiducial marker in meters relative to the rover's current position when in the field of view of the camera and not occluded by the structure. Define a measurement model for the x, y position measurements in terms of the rover and fiducial marker states. Use a measurement covariance of $\Sigma_{xx} = \Sigma_{yy} = 0.1$ and $\Sigma_{xy} = \Sigma_{yx} = 0.0$.

b) Define a semi-SLAM algorithm to simultaneously localize the rover relative to the known fiducials in two dimensions and map the unknown feature points as well. Assume a bicycle model for the rover, and create a 2D motion model with steering angle and forward speed as the inputs. Use disturbance covariance of $\Sigma_{xx} = \Sigma_{yy} = 0.02$, $\Sigma_{\theta} = 0.002$ and any other disturbance values set to 0.

c) Define the nonlinear Stanley steering control presented in class to control the rover to follow a path. Use a PI controller for speed control with a reference speed of 5 m/s.

d) Implement the SLAM and path following controller together for the scenario defined in **warehouse.m**. Present plots of the rover's true and estimated position, and the estimated positions of the unknown fiducial markers. Provide error ellipses for both rover and markers at reasonable spacing along the path. Also, provide plots of the control performance along the path in the x-y plane, and with respect to time for each of vehicle states.