ME 597: Autonomous Mobile Robotics

Final Exam

December 13th, 9:00 AM – December 14th, 9:00 AM, 2014

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: 647 309-9198) 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. **EKFSLAM with 3D features from a Microsoft Kinect [30].** Robots navigating in indoor environments lose a great deal of information when they rely only on 2D measurements. In this problem, we will incorporate 3D point information into the 2D SLAM problem for an Ackermann steering robot moving on flat terrain. We will assume that features are already extracted and can be accurately corresponded between frames, and that the projections of the features into 3D space are automatically performed. As such, the measurements to each feature will be in spherical coordinates relative to the Kinect camera center, as a range, azimuth and elevation in the robot body frame.

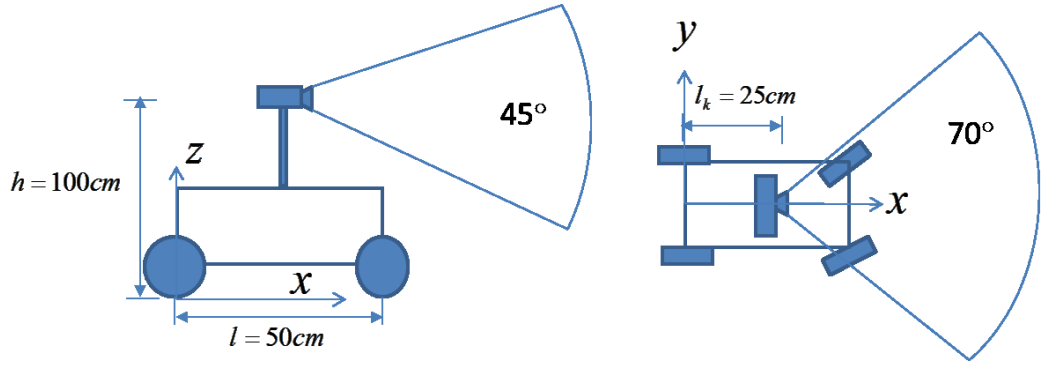

Figure 1: Ackermann robot dimensions and Kinect field of view.

a) Define the standard two wheel bicycle robot model for the Ackermann steering vehicle, and define constant velocity and steering angle inputs of 3 m/s and 3°. Use Gaussian disturbances with covariance $\sigma_{xy}=(0.05)^2\ m^2$ and $\sigma_{\theta}=(0.01)^2\ rad^2$ and an update rate of 10 Hz.

b) Define an expanded state vector that captures the 3D position of the features in the environment as well as a measurement model that includes range, azimuth and elevation measurements (relative spherical coordinates from the camera to each feature). Use Gaussian noise of $\sigma_r=(0.08)^2\ m^2$ and $\sigma_{ae}=(0.01)^2\ rad^2$. The update rate is also 10 Hz, the maximum detection range is 6 m, the field of view of the sensor is 70 degrees horizontally and 45 degrees vertically, and the Kinect is positioned so the image plane is perpendicular to the robot forward (robot x) axis, at the center of the robot and at a height of 1m off the ground. See Figure 1 for a representation of the robot and sensor configuration.

c) Generate a random environment with 200 features located around the vehicle's circular path between the heights of 0 and 2 m. Implement the motion and measurement models in simulation and plot the robot pose and measurements to features at three different instances over a simulation that completes a single near circle (if there were no disturbances, it would be a circle). Try to present the results in such a way that the measurement vectors are clearly visible and correspond to particular features, by only plotting visible features and by positioning the plot view to best illustrate the measurement vectors. Note: there is no need to consider the features as obstacles, simply pass through them for this question.

d) Define an EKF to solve the SLAM problem using 3D features. Assume perfect correspondence of features and be sure to define your feature initialization strategy and your motion and measurement update steps including linearization.

e) Implement your solution and present the complete true and estimated trajectory of the robot, plus the same three single viewpoints with 2D error covariance estimates for the robot and the visible features in the x-y and x-z planes after the EKF measurement update. Are there differences in estimation accuracy between the three axes, and if so, why?

2. **Aerial Racing – Mapping and reactive planning in 2D [35]:** In this question, we will define a combined mapping and planning strategy for small aerial racers. Since these vehicles have reliable attitude and altitude stabilization, and sufficiently good onboard sensors for highly accurate pose estimation, we will focus on mapping of the race course and using trajectory rollout to find fast routes.

Figure 2: Aerial Racing.

a) The multirotor racers can be modeled in 2D with acceleration inputs in the body x and y directions as well as a heading rate input. Disturbances in the inertial frame will affect the position and velocity and are normally distributed with covariances of $\sigma_{xy}=(0.05)^2$ m^2 and $\sigma_{vxy}=(0.03)^2$ m^2/s^2, and will affect heading with covariance $\sigma_{\theta}=(0.02)^2$ rad^2. Define a motion model for the vehicle with five states in the inertial frame and be sure to rotate the body acceleration inputs correctly. Use an update rate of 10 Hz. For safety reasons, set vehicle input command limits so that the vehicle does not exceed a maximum acceleration in any direction of $a_{max} = g \sin(20^\circ)$, which corresponds to a maximum bank angle of 20°. g = 9.81 m/s^2.

b) The racers will rely on four sonar sensors mounted on a gimbal to detect the free space in front of them while moving through the forest. The gimbal will ensure that the sonar rig is

held parallel to the ground during the race. Each sonar has a response cone that is 30° wide and a maximum range of 15 m, and the four sonar must be individually fired to avoid interference between sonar. The update rate for the sonar is also 10 Hz, but only one sonar can be fired at each timestep. Design a sonar configuration that will work well for racing, and define your measurement and inverse measurement models for each sonar. The range measurement is affected by Gaussian noise of covariance $\sigma_r=(0.5)^2$ m^2 and set the inverse probabilities for unoccupied space at 0.2 and for occupied space at 0.6, as the sonar is imprecise in where the objects are but good at detecting free space.

c) Assume perfect knowledge of the robot position (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 and current heading to the goal location as a desired direction. Be sure to explain how you select what direction to point the vehicle sensors, which sensors to fire and what inputs to use for trajectory rollout. A waypoint is considered reached if the vehicle crosses the line emanating from the endpoint of a segment perpendicular to that segment and outward from the center of the course, as depicted in Figure 3.

d) Implement the algorithm defined above and fly a single circuit of the course, defined in racing.m and depicted in Figure 3, from the start through the two waypoints and back to the start point in the order specified. Use only the currently constructed map for planning, and note that the overall map size is 111.2 X 73.3 m, and is defined in 0.1 m grid cells. Show your map and path at six points, half way through and at the end of each segment of the race.

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

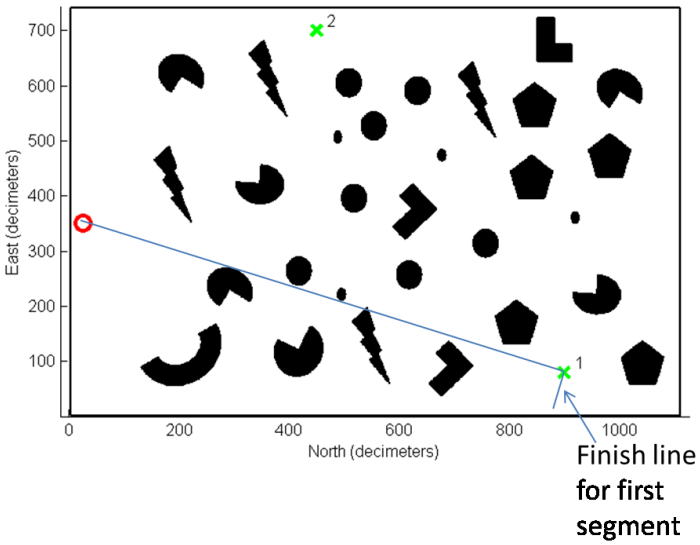

Figure 3: Map and first line segment with finish line for aerial racing.

3. **Motion Planning for Cleaning Robots [35].** Avidbots, a Waterloo area startup in the Velocity Foundry, has requested your help with defining planning algorithms for cleaning commercial spaces with a robotic cleaner. The robot has a straight front with dual brushes and curved tail squeegee, and is 50 cm X 50 cm. It has two drive wheels of radius 12.5 cm on each side of the vehicle at the back, as depicted in Figure 4. The drive wheels can only be driven with rotation rates between [0, 8] rad/s, meaning the wheels cannot be rotated backward, as this interferes with the squeegee operation.

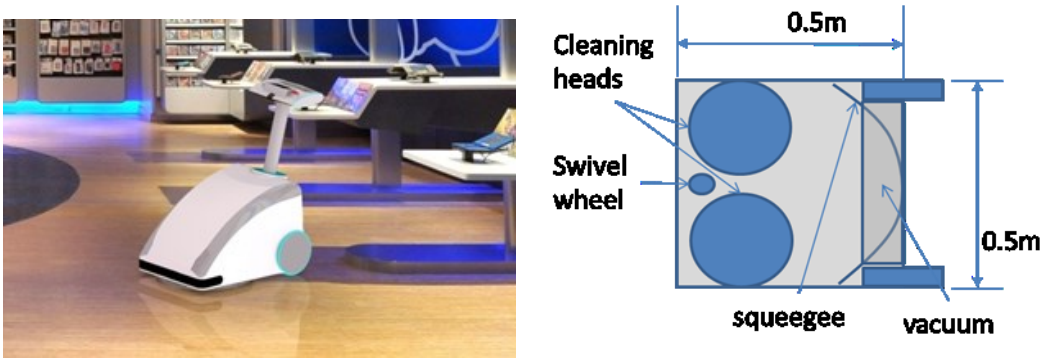

Figure 4: Avidbots Cleaning Robot, preliminary design and top-down schematic.

a) Your first task is to define large scale planning methods, without low-level vehicle kinematic constraints. The planning is to be performed on the map presented in Figure 5 and created with Matlab file floormap.m. Of particular interest are 5 high dirt areas near doorways and at the dustiest corners of large rooms. Find the shortest path from the base station (red circle), through each of the 5 interest points (green circles) and back to the base station. You can ignore the vehicle dimensions for collision checks (treat the robot as a point). Implement your solution method, present your shortest path and explain how you selected the interest point sequence.

b) For more detailed cleaning, a more accurate model is needed for the robot's motion. Define a motion model for the vehicle, ignoring disturbances. Define the inputs required to generate a set of motion primitives including a straight line segment of arbitrary length, d, at constant velocity 0.5 m/s and 90° turns left and right for cleaning inside and outside right angle corners with maximum wheel turn rates of 4 rad/s. When cleaning inside a 90° corner, it is not necessary to define a path that exits leaving along the wall exactly.

c) For each of the motion primitives, define a bounding rectangle that can be used as a simple collision check with the environment. What percentage of the bounding volume is actually cleaned when executing the maneuver (try to make this as large as possible)? Using the environment generated by the Matlab file floormap.m, depicted in Figure 5, randomly sample 15 configurations within the map and confirm whether each of the motion primitives can be performed at that location. Label the points on a plot and tabulate the results for the collision checking.

d) Assume that 90° turns can be completed inside a 1 m square for both inward and outward turns and that the robot leaves the square adjacent to the wall. Define a wall following algorithm that uses the motion primitives defined above to clean as closely as possible along the boundaries of the environment and obstacles. Explain the decision process you take to make sure that all 12 obstacles are properly circled. Note: you may use prior knowledge of the map to define the sequence of obstacle cleaning.

e) Implement your wall following algorithm and include plots that present the vehicle path through the environment. Feel free to remove the colors from the map if it will improve readability, or to present the path at different stages if it will help clarify the sequence of operations.

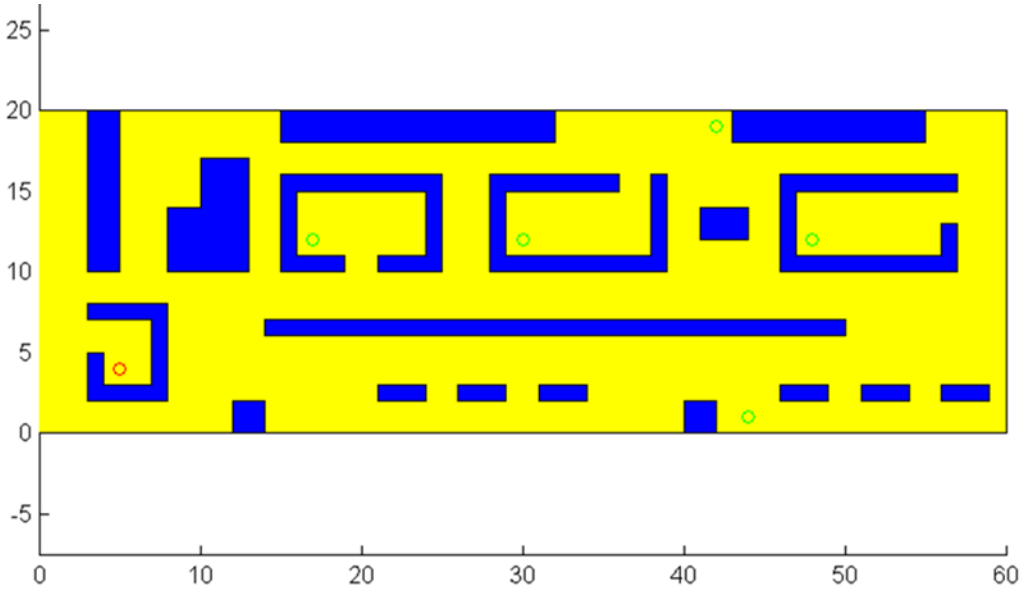

Figure 5: Cleaning area floor plan. Blue area is occupied, yellow is free, red circle is home base and green circles are interest points.