MAE 5180, ECE 5772

Autonomous Mobile Robots

Final Competition

Instructor:
Dr. Hadas Kress-Gazit

1 Overview

The final competition will combine several topics covered over the course of the semester, including sensing, localization, mapping and planning. The challenge will be performed using the Create simulator.

The final competition is to be completed in teams of two or three, but note that the **final report will be individual work**. All team members should have equal input in the team's work and all team members must write code. In the final report, each student must discuss their contribution. All members of the team must be present during the competition (unless otherwise excused by the professor) and should be prepared to answer questions about the techniques and algorithms used.

2 Objective

Your robot will be placed in one of k possible initial positions, with arbitrary orientation. The robot's task is to:

- Localize itself (see Section 4.2)
- Determine which of the optional walls are actually in the environment and produce an actual map (see Section 4.1)
- Navigate to as many of the given waypoints as possible, while trying to avoid "stay-away points" (see Section 4.3)

3 Logistics

Time and location: The final competition will take place on **Tuesday, May 12** during class (10:10-11:25am EDT).

Code: All teams must upload their code to Canvas by **8:30 am EDT, Tuesday, May 12**. The uploaded code (zip file) must include all files needed to run the program. The zip file must contain a function called `finalCompetition.m` which is the main function that will be run by the instructor. The zip file must **not** contain any simulator or Create toolbox files.

Teams: Students will work in teams of 2-3. Each team must fill out the registration form (one form per team) by **Friday, April 24, 11:59 pm EDT**.

Design Pitch: As described in Section 8.1, teams will present a design pitch on **Tuesday, April 28**. Due to access and time zone concerns, we will schedule the design pitch at a time that is maximally convenient.

4 Technical challenges

To succeed in the competition, the robot's program must include the following elements:

4.1 Mapping

The file `compMap.mat` will be provided at the start of the competition and will contain six variables:

1. `map`: an $n \times 4$ matrix consisting of known walls in the environment
2. `optWalls`: an $m \times 4$ matrix consisting of optional walls
3. `waypoints`: a $k \times 2$ matrix of k waypoints given as $[x, y]$. The robot will be started at one of these waypoints.

4. `ECwaypoints`: a $j \times 2$ matrix of j waypoints given as $[x, y]$. Visiting these waypoints will earn the team more points but they will typically be in harder to reach areas of the map.

5. `StayAwayPoints`: a $l \times 2$ matrix of l waypoints given as $[x, y]$. Driving over these waypoints (any part of the robot is over the waypoint) will deduct the team points.

6. `beaconLoc`: an $b \times 3$ matrix where each row gives $[tagNum, x, y]$ for a single beacon

Optional walls are walls that *may* or *may not* exist in the environment, and must be detected by the robot. Optional walls will not "appear" or "disappear," and they will not block off areas of the map such as to make them entirely inaccessible. Your final map (to be shown upon completion of the mission) should indicate which optional walls were present in the environment (see Section 6).

4.2 Localization

There will be **no overhead localization (truth pose)** available for the competition. Therefore, teams will be expected to implement various techniques to:

1. initialize the robot's starting position
2. localize the robot as it moves about the environment
3. avoid hitting obstacles (or relocalize if a collision is detected)

The robot will be started in one of k points in *waypoints* with arbitrary orientation. The robot will NOT be started in one of the points in *ECwaypoints*. No optional walls will be visible from any of the possible starting positions. Once the robot has determined where it has started, it will need to continue to localize itself as it moves about its environment. Note that once the robot leaves its starting position, it may be able to see optional walls (where present). Keep this in mind as you develop your localization algorithm.

The simulator used for running the code will have the "overhead localization" (i.e. truth pose) **disabled**. Your algorithms should not use the truth pose data.

4.3 Navigating to waypoints

The variables *waypoints*, *ECwaypoints*, and *StayAwayPoints* contain a list of x,y locations the robot should or should not visit. The robot will start in one of the points in *waypoints*. Visiting points in *ECwaypoints* will award the team a higher score than visiting points in *waypoints*; however, points in *ECwaypoints* will be more difficult to navigate to. A robot driving over points (any part of the robot goes over the point) in *StayAwayPoints* will incur a point penalty. Course staff will determine whether the robot drove over such a point.
Rules:

- The robot may move at a speed of **up to** 0.2 m/sec.
- The robot must indicate it has reached a waypoint by turning the power LED in the simulator from green to red. After leaving the waypoint, the power LED must be changed to green again. This can be done using the function `SetLEDsRoomba`.
- The robot may declare it has reached a waypoint if it is at most 0.2 meters away from it.

The final map (to be shown upon completion of the mission) should indicate which waypoints were visited. Keep in mind that teams will be scored on the amount of time it takes to complete the mission, so an intelligent motion planning strategy will likely be necessary (as well as a method for keeping track of where the robot has already been).

5 Development and Testing

This competition builds on the code written for the homework and labs. You are welcome and encouraged to reuse any of the code and/or implement any algorithm you choose. Keep in mind that **integration of all the components is not trivial**. Spend time debugging the full system. The maps provided for testing will **not** be the same map for the competition, so your code should load `compMap.mat` (variables described in Section 4.1) and run without any additional tuning or debugging.

6 The competition

For the competition, one machine will be used to run the code for all of the groups. Each team will have 5 minutes to run their robot. Teams may restart their program as many times as they wish within the time bound. The run with the best score will be used to determine the winner.

The file `compMap.mat` will be placed in the group's code directory which will also be the working directory when the simulator is called.

Upon completion of the mission, the code should output:

1. data structure (such as the `dataStore`) containing:
   (a) robot pose as calculated by the localization algorithm
   (b) odometry
   (c) depth data
   (d) bump data
   (e) beacon data
   (f) the final map (same structure as `map` in `compMap.mat`, including known walls and all observed optional walls)
   (g) a list of the visited waypoints
2. a plot displaying the map, the robot's trajectory, and waypoints that were visited. Optional walls that are determined to exist should be drawn in black, optional walls that have not been determined should be drawn in red and optional walls that have been determined to not exist should not be drawn.

The code should be storing data to a global variable so that if the program has to be killed (using Ctrl C), the team will still have data to use for the individual final reports.

7 Scoring

The team's score for the competition will be calculated as follows:

- 10 points for each correct *waypoint* visited
- 20 points for each correct *ECwaypoint* visited
- -5 points every time the robot indicates incorrectly that it is at a waypoint
- -5 points every time the robot drives over a stay-away point
- 10 points for each optional wall that is correctly determined (is in the workspace or not)
- -10 points for each optional wall that is incorrectly determined (no points are deducted for optional walls that have not been determined)
- 10*(time limit (minutes) - actual time(minutes)) if all *waypoints* and *ECwaypoints* are visited before time runs out

8 Grading

8.1 Design pitch (40 points)

Teams will pitch their competition strategy to the professor. This will be in the form of a 5 minute presentation and must cover overall strategy, algorithm choices and integration plan. The presentation may take any form (ppt, pdf, writing on paper with a webcam pointing at the paper, no material) as long as the presentation is clear and does not exceed 5 minutes. All team members must participate in the presentation unless agreed otherwise with the professor.

The presentations will take place on **Tuesday, April 28**. A sign up process and a grading rubric will be posted on Canvas.

8.2 Competition attendance

All team members are expected to attend the competition unless agreed otherwise with the professor.

8.3 Competition code

If the team's submitted code does not run (i.e. throws errors), all team members **lose 25 points** of their final competition grade.

8.4 Final competition report (200 points)

Students are responsible for writing up **individual** reports, to be due on **May 19 by 11:59 pm**. These reports should include (page limits strictly enforced):

1. Overview of the team's approach - which algorithms were chosen, why they were chosen (at most 1/2 page)
2. Flow chart of the solution. This may be the same for all the group members (at most 1 page. No text should be included to describe the chart beyond the labels as it should be self explanatory)
3. Description of individual contribution and estimate of hours spent on integration and testing (at most 1/2 page)
4. Discussion of competition performance - what worked well, what didn't and why. Provide accompanying plots to support your analysis (at most 2 pages)

Reports should be submitted as a pdf file (`report.pdf`) and should be **single column, no longer than 4 pages, with font no smaller than 11pt**. Figures may be added as an appendix that is not subject to the page limit. Text in the appendix (other than succinct captions) will **not** be read.

Students are **highly** encouraged to use data gathered during development and during the competition to tweak the algorithms and to perform post processing in order to provide more insightful discussions in the final report.

Grading rubric for the report will be posted on Canvas. Student are **highly** encouraged to read the document and structure the report based on the rubric.