Homework 5

ENME 808T: Network Control Systems
Department of Mechanical Engineering | Fall 2019

Due: Friday, December 6th, 2019 by 5:00 pm

Project Description

For this assignment, we will do an implementation on real robots of some of the distributed controllers learned in class. The overall mission is a simulated *multi-robot search and rescue* operation.

In this scenario, a team of researchers were performing experiments in a facility located in a remote area when *an incident* occurred. You are tasked with the deployment of a team of robots to search and flag survivors so they can be rescued. The agents will be deployed from an air drop in the vicinity of the research facility. Outside the facility, no communication infrastructure is available, so the agents will need to ensure that they remain connected to exchange information while navigating a cluttered environment. For security purposes, only one of the agents is made privy of the location of the research facility.

To detect obstacles, each agent is equipped with *eight range sensors*, uniformly spaced around the body of the robots (i.e., in the directions $0°$, $45°$, ... , $315°$ from the center of the robot). Each sensor returns the distance to any object detected in the range $[\delta, \Delta]$. If no objects are detected, the sensors return a value of `Inf`.

While in the facility, agents are able to use the existing communication infrastructure to search for the researchers. A survivor can be flagged for rescue if an agent is sufficiently close. There are three submissions to be undertaken.

1. *Initial Deployment*

   The agents are initially deployed in the outskirts of the search area. The agents are equipped with short-range communication equipment, and their interactions can be modeled using a Δ-disk graph. Assuming that the agents are initially connected, their first objective is to rendezvous and ensure that all agents can communicate with another while avoiding collisions.

   For this submission, each agent had access to:

   - Current submission: ‘init’
   - Own state and identity.
   - Δ-disk graph neighbor states and identities.
   - Minimum safety distance δ.
   - Maximum interaction distance Δ.
   - Sensor data.
   - Mission specific info: none (empty array).

   The submission is accomplished when the graph becomes connected.

2. *Navigating the Cluttered Environment*

   Upon rendezvous, the agents will switch their objective to enter the search area while avoiding debris. To do this, the agents will enter into a leader-follower formation mode. All agents will have access to data from eight range sensors located uniformly around the body of the robots which will allow them to sense obstacles, but only one agent will have access to the last known location of the robots.

   For this submission, each agent had access to:

   - Current submission: 'navigate'
   - Own state and identity.
   - Δ-disk graph neighbor states and identities.
   - Minimum safety distance δ.
   - Maximum interaction distance Δ.
   - Sensor data.
   - Mission specific info:
     - Leader agent: facility location `xt`.
     - Follower agents: none (empty array).

   The submission is accomplished when all agents are inside the research facility.

3. *Search and Rescue*

   Once at the facility, the agents will switch to exploration mode to search for survivors. All agents will have access to four corner beacons that denote the boundary of the search space. While inside the facility, the network connectivity is given by a Delaunay graph. Survivors are marked when an agent is sufficiently close.

   For this submission, each agent had access to:

   - Current submission: 'search'
   - Own state and identity.
   - Delaunay graph neighbor states and identities.
   - Minimum safety distance δ.
   - Maximum interaction distance Δ.
   - Sensor data.
   - Mission specific info:
     - Domain: beacon positions at the four corners.
     - Own Voronoi Cell: Vertices of the polygon that defines the Voronoi cell for the agent.
     - Neighbor Voronoi Cells: Vertices of the polygons that define the Voronoi cells for the Delaunay neighboring agents.

   The mission is accomplished when all survivors are flagged for rescue.

Deliverables

There are two deliverables required for full credit:

1. Prepare a brief write-up that contains:
   (a) *A description of the controller(s).* Include the rationale behind your choice of controller(s), and possible derivations (e.g., if obtained from edge-tension functions).
   (b) *Observations on the performance of your controllers.* These are your concluding remarks. For each submission, did everything work as expected? If something went wrong, why do you think this was the case? Did the simulations match up with the actual implementation?

2. Video of Robotarium Implementation:

   Go to `https://www.robotarium.gatech.edu/sign-in`, select sign-in with email, and select sign-up at the top to create a Robotarium account. You will be prompted to verify your email and provide account information. Complete the form and write "ENME808T Final Project, University of Maryland" for the reason of use. Your account needs to be verified by someone before it gets created, and this takes time: **Do this as soon as you can!**

   After requesting an account, download the Robotarium API found in the zip file `ENME808T_Final_Project_Robotarium.zip` that is under the M-Files Module in Canvas. You will be making modifications to the `controller.m` file to incorporate your control design and algorithms.

   (a) *Simulation:* Your algorithms will need to be tested in simulation prior to running on the Robotarium by setting `simulate_true = true` in `main.m`. You can run each submission independently by setting the variable `missionChoice` in `main.m` to `'init'`, `'navigate'`, or `'search'` and running `main.m`. The script will display a message indicating whether or not your designs successfully complete the submission. Once your designs are implemented and successfully accomplish each submission, you will run the full mission by setting `missionChoice` to 'full'. Upon success, a script will check for satisfaction of actuator limits and collision avoidance, and inform you whether your designs are likely to be successful.
   (b) *Robotarium Experiment:* Upon successful execution of the full simulation, you are ready to submit your design as an experiment. Sign-in to the Robotarium website, go to the Dashboard, and select "+Submit New Experiment". Provide the requested information: Title, description, estimated duration. Enter 5 for the number of robots. There are three files that must be submitted for each experiment: `main.m`, `controller.m`, and `missionMap.jpg`. Once they are uploaded, make sure to mark `main.m` as the main file. Your experiment will be marked as pending and go into a queue. You will be notified via email when the experiment has been run. When you revisit the experiment in the Dashboard, you will see a video recording of the experiment, the saved .mat data file, and a .txt log file. Check the video for a successful implementation. If the experiment did not run as expected, inspect your data and the log files to figure out what went wrong. You will need to modify your design and repeat the process until you are successful. **Each experiment can take up to two days to run, don't wait until the last minute!** Once you have a successful implementation, submit a the video recording obtained during the Robotarium experiment for complete credit.

Good Luck!