MEAM620 Project 3

April 13, 2015

For Project 3, we will divide the class into three sections. Each section will pursue a different project: Project 3A, Project 3B, and Project 3C. Each project will have two phases. We will assign you to a section (A, B, or C) taking into account your preferences. However, you are free to form your own groups within the section.

The final deliverables for each project are:

1. **Group report:** A conference paper quality report, no more than 6 pages including images, describing your techniques and implementation and presenting clear figures and plots demonstrating your results. More specific requirements are included in each project description.

2. **Individual report:** Send a 1 paragraph e-mail to meam620@gmail.com that includes your name and group number and describes your personal contributions to the project and the contributions of your teammates.

3. **Demonstration:** Either a video or live demo of your algorithm running on a quadrotor.

4. **Code:** A .zip file of all your code.

1 Project 3A: Multirobot coordination

Project 3A will be a simulation-only project dealing with planning for multi-robot teams.

Assume we have a set of N robots at given start locations that need to complete (navigate to) M tasks at given goal locations. Further, assume that the robots are interchangeable, that is, it doesn't matter which robot visits which goal location, as long as all goal locations are visited. The problem is to assign robots to goals and then generate trajectories that will navigate robots to their assigned goals in an optimal and safe manner. In this project, you will implement some state-of-the-art methods to solve this problem.

1.1 Phase 1

For phase 1, read Sections 1-3 of the following paper:

M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," The International Journal of Robotics Research, 2014.

To show that you understood the paper, implement the centralized CAPT algorithm (C-CAPT) for circular, first-order robots operating in a 2D obstacle-free workspace. You may use code available online to implement the Hungarian algorithm **only**, provided you cite it properly. In your report, make sure to include the following:

1. Plot the runtime of your CAPT implementation for a team of 10, 50, 100, 200, 300, 400, etc. up to 1000 robots navigating to an equal number of goals. You can randomly generate start and goal positions.

You will be evaluated on both correctness and efficiency of your algorithm. Your computation times should be approximately the same as those reported in the paper.

1.2 Phase 2

For phase 2, test your C-CAPT algorithm using the quadrotor dynamic simulator. Note that you will have to extend your algorithm to 3D, and impose the change-of-variables described in Section 4 of the paper to maintain an acceptable vertical distance between robots. Generate the following deliverable:

1. Create a video simulating teams of 10 and 15 quadrotors navigating from randomly generated start positions to randomly generated goal positions. Use one example to demonstrate the case where there are more robots than goals, and another example where there are twice as many goal positions and robots. In the second case, you will have to iterate the C-CAPT algorithm twice.

Then, extend the C-CAPT algorithm with either one of these two options:

1. Adapt the C-CAPT algorithm using the decentralized CAPT algorithm (D-CAPT) found in Section 5 of the same paper: M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," The International Journal of Robotics Research, 2014.

2. Extend the CAPT algorithm to robots operating in known obstacle-filled environments using the Goal Assignment and Trajectory Planning (GAP) algorithm found in this paper: M. Turpin, K. Mohta, N. Michael, and V. Kumar, "Goal assignment and trajectory planning for large teams of interchangeable robots," Robotics: Science and Systems 2013.

After you have read and understood the paper, implement the algorithm for circular first-order robots operating in an obstacle-free workspace. You are free to re-use code you've previously written in other projects of this class (for example, your map creating code from Project 1).

Come up with at least 3 interesting test cases that illustrate the advantages and limitations of the algorithm you choose. Make sure to include the following in your report:

1. Discuss what you found the advantages and limitations of the algorithm to be. Describe each of your test cases and what they show about the algorithm.

2. Create a video showing a simulation of your test cases. Make sure to somehow show the communication rage and extent of the robots, and indicate if they switch goals with each other (for D-CAPT).

You will be evaluated on both the correctness of your implementation and the thoughtfulness of your test cases and analysis.

2 Project 3B: Vision-based quadrotor control

Project 3B will be an implementation of the Extended Kalman Filter (EKF) you developed in simulation in Project 2 Phase 3 on the KMel Nano+ quadrotor platform you used in the labs in Project 1. The ultimate goal is to demonstrate the tasks from Project 1 Phase 4 - hover, navigate to waypoint, and navigate to multiple waypoints - without state feedback from Vicon.

In phase 1, you will implement EKF1 from Project 2 Phase 2 on the Nano+ quadrotor. This means that you will be getting your pose estimates from the April Tag map while getting your velocity estimates from Vicon. In phase 2, you will implement EKF2 from Project 2 Phase 2 on the Nano+ quadrotor, where you get both your pose and velocity estimates from vision. To complete each phase, you will again have to successfully demonstrate to a TA your quadrotor's ability to hover, navigate to a waypoint, and navigate to multiple waypoints using your EKF. Note that for this phase, like in Project 1 Phase 4, **your group will only have 2 lab sessions to complete your project**.

Your report should include but not be limited to: plots of tracking error versus time, position versus time, and 3D plots of paths. You may use pose from Vicon as ground truth for comparison. It should also include any information needed to replicate your EKF, such as values for Q and R and how you chose them.

3 Project 3C: Estimation and Control for an Off-The-Shelf Quadrotor

Project 3C will be a hardware project focusing on implementing the estimation and control techniques we discussed in class on an off-the-shelf commercial platform. Specifically, we will be providing you with Parrot Bebop Drones (http://www.parrot.com/usa/products/bebop-drone/). Instead of using MATLAB, the robot software interface is through Parrot's API, which is written in C. We strongly recommend that you have prior experience working with C and/or C++.

For this project **only**, you are free to use any open-source code you find on the Internet. In your report, please list all the code you've used and cite its source. Also note that you are expected to describe all algorithms you are running in your report, even if you did not write the code yourself.

3.1 Phase 1

For phase 1, you must demonstrate the ability of your Bebop to travel in straight-line paths between two points. Include the following items in your report:

1. An illustration of how your code is interfacing with the Bebop's sensor outputs and control inputs.
2. A description, with equations, of the control and estimation algorithms you are running and a discussion of what the strengths and weaknesses of your approach are.
3. A demo of your Bebop moving between two points.

3.2 Phase 2

For phase 2, demonstrate different **closed** trajectories using your Bebop (for example, figure 8, circle, square, etc.). Include the following items in your report:

1. A description, with equations, of the techniques you are using for control and state estimation and a discussion of what the strengths and weaknesses of your approach are.
2. A video of your Bebop executing its trajectories.

You will be evaluated on your Bebop demonstration, as well as the clarity of your report in describing the Bebop's/your code architecture and your algorithm and the thoughtfulness of your discussion.