EECE 5550 Mobile Robotics Final Project

David M. Rosen and Derya Aksaray

Due: Dec 15, 2022

Motivation and background

Over the previous two decades, mobile robots have begun to play an increasingly important role in *disaster response*, providing emergency response teams with both improved situational awareness and physical control authority in hazardous environments.[1] One particularly prominent application is *reconnaissance*: in this task, emergency responders will deploy mobile robots to quickly survey a potentially dangerous environment (such as the interior of a collapsed building) to identify potential hazards and/or the locations of victims. This information can subsequently be used to develop an effective action plan that minimizes the risk to emergency workers.

Objective and problem formulation

In this project, you will design and implement a complete autonomous system to perform reconnaissance in a simulated disaster environment. Specifically, when introduced into a *closed* but *initially unknown* environment, your system should do the following:

1. Generate a ***complete*** map of the environment,
2. Locate any victims present in the environment.

For the purposes of this exercise, we will represent the environment using an occupancy grid map, and use AprilTags as stand-ins for simulated victims. Therefore, the specific outputs of your system corresponding to points (1) and (2) above should be:

1. A ***complete*** occupancy grid map of the environment.
2. A list of AprilTags discovered in the environment. Each element of this list must specify:
   - The ID (number) of the detected tag
   - The pose of the tag, *expressed with respect to the map frame*.

Rules of the game

You may construct your system using any off-the-shelf components that you like. For example, you may want to make use of the Cartographer package that we saw in Lab 3 in order to build occupancy grid maps, the move_base package from the ROS navigation stack in order to generate

[1] See (for example) Chapter 60 of the *Springer Handbook of Robotics* for a recent survey on the use of mobile robotics in disaster response.

and follow motion plans to move the robot around the environment, and/or a (basic) frontier exploration package such as explore_lite.

However, you should consider that a baseline system constructed *solely* from off-the-shelf components may not perform very well on the target task. For example:

- While a basic frontier-based exploration planner like `explore_lite` can ensure a complete *map*, it does not address the goal of searching for "victims" (AprilTags). You might consider either modifying `explore_lite`, or implementing your own planner, in order to improve your system's ability to find all of the tags present in an environment.
- The (estimated) pose associated with an AprilTag detection can be rather noisy, especially when the tag is viewed from a distance. You might want to think about possible approaches for ensuring that the pose of each detected tag is estimated accurately in the final map.

In addition to ROS packages, when implementing your own system you may find the OpenCV library (especially the image processing module) useful for operating on occupancy grid maps, and/or the GTSAM library useful for performing parameter estimation, should your approach require it.

Deliverables

Your team will do a **group presentation** in the last week of lectures to present the proposed idea and show preliminary results. This will be an opportunity to get feedback about your proposed design before the final report submission. Moreover, your team should submit the following:

1. The code for any ROS nodes that your team implements or modifies.
2. The launch file(s) used to start up your system.
3. A group report describing the design of your system. This does not need to be lengthy, but it must contain the following elements:
    - Use IEEE conference template for your report (`https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn`) including the following sections: Abstract, Introduction and Motivation, Proposed Solution, Results, Conclusion and properly cited references.
    - A clear description of your system's construction. Operationally, this means that a skilled practitioner should be able to reproduce your system *without* looking at your team's code.
    - The rationale for your design: that is, *why* your team adopted this design, and why it is a good solution for the task at hand [2].
    - Experimental results from an example application of your system. This should include a (brief) description of the environment in which your system was tested (i.e. type of location, approximate size of environment, number of tags present)[3], and a visualization of the final occupancy grid map and detected AprilTags reported by your system.

[2] You are **encouraged to do a benchmark analysis** by considering a baseline design (relying on the off-the-shelf tools) and comparing the performance of the baseline design with the one obtained from your proposed design.

[3] While you design your environment, please make sure that you have **at least** 5 obstacles and 6 tags in the environment, and the tags are distributed in the environment with sufficient distance from each other and located at various heights.