Massachusetts Institute of Technology

Robotics: Science and Systems – Spring 2005

Lab 9: The Course Challenge

Thursday, Apr 14 an on

Objectives and Lab Overview

Starting with this lab, your objective is to program your robot to solve the course challenge: "build a shelter on Mars". Your robot will traverse an environment with known map but also dynamically changing features, gathering components. Your robot will deposit all the components at a specified location on the map, stacking them in the form of a wall.

This lab will give you technical guidance toward integrating many robot modules to solve this complex task.

Time Accounting and Self-Assessment:

Make a dated entry called "Start of Lab 9" on your logbook's Self-Assessment page. Before doing any of the lab parts below, answer the following questions:

- **Programming**: How proficient are you at writing large programs in Java (as of the start of Lab 9)?
(1=Not at all proficient; 2=slightly proficient; 3=reasonably proficient; 4=very proficient; 5=expert.)
- **Hardware**: How proficient are you at modifying the hardware of your robot?
(1=Not at all proficient; 2=slightly proficient; 3=reasonably proficient; 4=very proficient; 5=expert.)
- **Robot Programming**: How proficient are you at using the software on your robot?
(1=Not at all proficient; 2=slightly proficient; 3=reasonably proficient; 4=very proficient; 5=expert.)

To start the lab, you should have:

- This handout, a handout on Visual Navigation, and a third handout entitled "The course Challenge" detailing the challenge task.
- Your robot and robot environment from Lab 8.

Physical Units

We remind you to use MKS units (meters, kilograms, seconds, radians, watts, etc.) throughout the course and this lab. In particular, this means that *whenever you state a physical quantity, you must state its units*. Also show units in your intermediate calculations.

Part 1: Solution Design and Team Organization

The challenge task is the most complex task your robot will face during this term. Your robot will have to scale up its performance (both the complexity and the duration). So far, your robots have performed in half-minute chunks. For the challenge your robot will have to be reliable for 10-20 minutes, an order of magnitude longer! This raises many issues at the level of algorithmic design, software implementation and integration, and solution reliability.

Many of the modules you will need have already been implemented as part of individual labs: e.g., wall following, motion planning, grasping, object transport, visual servoing. For each of these modules you may use your solution, our solution, or you may build on the lab experience and implement something completely new.

The first step in thinking about how to put it all together is deciding of your "philospphical" approach to control. Do you want to choose a reactive control structure, a deliberative control structure, or a hybrid? What are the trade-offs? A reactive approach to control may rely on wall following and complex sensing to locate objects. A deliberative approach relies on a map to get access to a model of the world and therefore may rely on simpler sensing. Think about your robot's performance and the trade-offs in complexity between the various approaches and pick an approach. Is your robot's hardware sufficient for the approach you have chosen? If not, what modifications will be needed? You may modify the robot's hardware, but keep in mind the deadline for freezing the hardware on your robot. You will not be able to make any changes after that time.

One important consideration to keep in mind is that your robot will have to travel to many specified points. How do you choose to implement precision navigation and error detection and recovery? One option is to rely on a reactive strategy that uses odometry augmented by wall following and the topology of the environment. Whenever the robot detects a topological feature such as a corner it could update its idea of where it is located. This strategy is simple and robust if all the goal points for the robot are reachable by wall following. However, your challenge map will have significant open spaces where there will be no walls to follow so you will need to develop a solution for robust navigation within open spaces as well.

Another option is to use visual landmarks to ensure the robot has an accurate estimation of its location. The Visual Navigation handout explains how you can solve the robust navigation problem using vision.

Now iterate and define a high-level algorithm for how you will solve the challenge. Iterate further until you have expressed a solution in terms of the basic robot cpapbilities provided by Carmen, implemented in previous labs, or that are self-contained behavior modules for which you have a concrete implementation plan.

The next step is to define the APIs between all the software modules. Now make programming assignments for each software module so that you can parallelize the implementation of this project. Develop a project timeline and be sure to include time for module development and testing, time for software integration, and time for project testing and evaluation.

To Hand In by April 20: Your Web report should detail your high-level approach for solving the challenge task. Be sure to outline the hardware and software architecture for your solution and include algorithmic sketches for your solution. Also include a breakdown on software modules and their APIS. Finally, include the team task allocation – i.e., how the team members plan to split up the work – and the proposed schedule. Be sure to allow plenty of time for software integration and testing.

Part2: Go For It!

Your challenge handout and the course syllabus provides the timeline for your challenge development. We will approach this as a two-stage process. The qualifying round will give you an opportunity to do "dress-rehearsal" for your robot. This will also give you a chance to learn about the weaknesses of your solution so that you can improve your robot's performance for the final run.

A word of advice: don't just hack! Think carefully about your solutions and make sure they have some degree of generality.

To Hand In by May 13: Your Web report should detail your solution to the challenge. Please justify your choices, and provide (i.e., link to) the testing data you collected during development.

Each group member must also turn in an individually-authored essay entitled "Building Brains for Bodies," about the lessons you learned by doing the labs and the course challenge. Please commit this to the same repository directory as your group report, and link to it from the group report.

Wrap Up

Report the time spent on each part of the lab in person-hours and indicate what elements were done individually, in pairs, as triples (and by whom) or as a full group.