Two Lab Exercises for an Introductory Robotics Class

Jennifer Kay

Computer Science Department
Rowan University
201 Mullica Hill Road
Glassboro, NJ 08028
kay@elvis.rowan.edu

Abstract

Two lab exercises for an undergraduate Introduction to Robotics class are discussed. The first is an early assignment that uses a robot to measure the length and width of a box. The second is presented much later in the semester; students are assigned the task of coming up with an interesting project to implement with their robot kits. This second assignment resulted in significant student enthusiasm and interesting projects. Selected projects using a Handy Board on a Lego base are described.

Introduction

The Robotics class at Rowan university is designed for Junior and Senior computer science majors, and expects these students to have a solid mathematical background (including some calculus and linear algebra), as well as a basic background in programming, data structures, and algorithms.

Students are introduced not only to traditional industrial robot arms and machinery, but also to the field of autonomous mobile robots and current research in robotics. Robotics is taught from a Computer Science perspective, emphasizing the software, artificial intelligence, mathematical, and algorithmic aspects of the course, and de-emphasizing the low-level hardware control.

Although the course never meets in a computer laboratory, students are split into groups of two or three individuals, and each group is provided with a Handy Board and lots of Lego parts. Over the course of the semester, the groups are given several small assignments. These assignments help the students become more comfortable with the equipment and begin to understand its capabilities.

Early Lab: Measuring a Box

Educational Objectives

After completing this assignment, students should:

- Be comfortable writing programs in the robot programming language.
- Have experience using some of the sensors included in their robot kits.
- Have an understanding of the need for sensor calibration.
- Have developed a crude calibration algorithm.
- Understand that apparently minor changes in the environment (including such things as lighting and initial position of the robot) can have a major effect on the performance of the system.

In addition, students may:

- Gain additional design experience.

Background Material

This is generally the second lab assignment I give to students in my Robotics class. In their first assignment, students build the robot base (to my specifications), and implement an obstacle avoidance program that we have previously discussed in great detail in class. Thus, by the time they get the box assignment, students have a basic physical configuration for their robots, and an understanding of how to download and execute programs on the robot.

In my class we use Handy Boards [3] with Lego bases. This system, as well as Lego Mindstorms system, requires students to have some competency with Lego design. Fred Martin has written an excellent introduction to Lego design that really helps the student (or instructor) who may not have had extensive experience with Lego as a child. [7]

Students programming in Interactive C (IC) clearly benefit from the IC manual, which is available on-line. [4]

Description of Hardware and Software Required

As noted above, my class uses Handy Boards, programmed using Interactive C, and mounted on Lego bases. Students are provided with a variety of touch sensors as well as two types of range sensors, though a single type of sensor can be used.

A less expensive option for a laboratory is the Lego Mindstorms [6] system. The Lego Mindstorms kit comes with its own visual programming environment, which is frequently too limited for use in this type of course. However several open source alternatives exist for programming the Mindstorms in a wide variety of languages including Ada, C, Scheme, and Java. O'Hara and Kay provide a survey of open source software for educational robotics in [8]. More advanced robots such as Kheperas [5] and Pioneers [10] are certainly also an option, though the price is probably beyond that of many departments.

Students will appreciate it if you leave a few cardboard boxes of different dimensions in the lab so that they have some examples on which to test their systems.

Step-by-Step Instructions

The following is a slightly modified version of my most recent box measuring assignment. I have left in some Handy Board specific details (e.g. using the "start" and "stop" buttons), which would obviously be modified to suit your robot.

Introduction. This assignment describes three distinct programs to measure the outside of an orthogonal (i.e. made up of right angles) cardboard box. I encourage you to write programs 1 and 2 before writing program 3. Bring your robot to class on the due date ready to demonstrate the program with the highest number that works correctly, e.g., if you get 1 & 2 working, but not 3, you should bring your robot to class ready to demonstrate program 2. It is better to bring a completely working program 2 than a buggy program 3. In addition, bring to class a hard copy of the code, which should be clearly written and well documented. You may assume that the box will not move while your robot executes your program.

Program 1. Write a program that measures one side of a cardboard box using your robot. I will bring in a cardboard box to class that you will not have seen before the class starts.

Demonstration directions: You will place the robot at any location and orientation you like outside of the box, and then press the start button. Your robot should measure the side of the box, and when it finishes measuring it should stop moving and display the length of the side of the box, in centimeters, on its display (with a nice message, not just a number!)

Program 2. Write a program that computes the volume of a cardboard box. You will manually enter the height and width of the box in centimeters (see below). You may assume the box will be an integer number of centimeters high and wide, though the length may be a floating point value).

Demonstration directions: You will enter the height and width of the box, place the robot at any location and orientation you like outside of the box, and then press the start button. Your robot should measure the side of the box, and when it finishes measuring it should stop moving and display both the length of the side of the box and the volume of the whole box. You may use any method you choose to input the width and height of the box.

To enter the height and width of the box, I suggest the following technique: Press the stop button and let go and pause briefly. Repeat as many times as necessary to indicate the height of the box. (A message on the display confirming the number of presses would be helpful). Then press the start button. Now press the stop button and let go and pause briefly. Repeat as many times as necessary to indicate the width of the box. (Again, a message confirming the number of presses would be helpful). Then press the start button. Finally, place your robot as desired, and press the start button again to start it measuring.

Program 3. Write a program that computes the volume of a cardboard box. You will manually enter the height of the box in centimeters (see below). You may assume the box will be an integer number of centimeters high, though the length and width may be floating point values).

Demonstration directions: You will enter the height of the box, place the robot at any location and orientation you like outside of the box, and then press the start button. Your robot should measure two sides of the box (without any human intervention). When it finishes measuring it should stop moving and display the length, width, and volume of the box. You may use any method you choose to input the height of the box.

How to Evaluate Results

Clearly the accuracy of the students' measurements are one important metric. Code design and style is also important. Having students demonstrate their systems to the class is valuable in demonstrating different problem-solving techniques.

Final Lab: Do Something Cool

Introduction

Because my course does not have any class time in the lab, the early lab assignments are somewhat limited in their scope. This project is designed to have the students attempt a larger task, with an assignment that they have more time to complete.

I treat this assignment as a mini research project for the students. Students are required to provide a written proposal before they begin their work, and oral and poster presentations are required in addition to a final demonstration and write-up.

Educational Objectives

After completing this assignment, students should:

- Have experience developing a more complex robotic system.
- Have integrated concepts that they have learned in the class into their robot.
- Have experience creating a project specification.
- Be able to design a research poster to present their work.

In addition, students may:

- Gain additional design experience.
- Have a lot of fun.

Background Material

In this lab, students are asked to come up with an interesting robotics project and implement it. Because the assignment is so general, it is appropriate for students with virtually any prior hands-on experience in robotics.

The first time I taught Robotics, this was not the final project; it was followed with a collect ping-pong balls competition. Much to my surprise, the students were less enthusiastic about the competition than about the "cool" projects. Using this as a final project the next time I taught the course, and giving the students more time to work on it, proved more successful. I plan on continuing to use it as a final class project.

This assignment generates tremendous excitement in the class. During the first few days after the project is assigned students frequently approach me, whispering their ideas so that other groups can not hear.

Step-by-Step Instructions

Introduction. This should the most enjoyable, but also the most challenging of your lab assignments. Your goal is to "do something interesting" with your robot. You may work as a single team, or have a pair of teams work together on a multi-robot project.

While I encourage you to come up with the "coolest" project you can, you need to think carefully about what you can accomplish in the time allotted. Your project should be designed with two goals in mind. The first should be an interesting project that you are reasonably confident you can achieve, given what we have learned in class and your prior project experience. The second should be more of a stretch that builds on your first goal and goes further, trying to accomplish something that you are not certain you can do.

Part 1: Talk To Me. Your team (or you and your partner team) should outline a project you are interested in pursuing. Then, as a group, come and talk to me for a few minutes about your idea. I want to make sure that your two goals are reasonable: big enough to be interesting and challenging, but limited enough that you should be able to complete your project in the available time.

Part 2: Project Proposal. Once I have spoken with your group and agree that you have a good idea, write a one or two page project proposal. Be sure to include:

- Project Summary. This should be a one or two paragraph high-level overview of the final system. You should write your summary in such a way that a Junior or Senior computer science student (including those without any experience in robotics) would understand the intended behavior of your system.
- Project Description. This will be the bulk of your proposal. Explain in detail what you plan to do, and how you plan to do it. Include your plans for the robot design, the sensors you expect to use, and thoughts about the software design. Your plans for how you expect to achieve your first goal should be quite detailed. Your plans for how you expect to achieve your second goal should show that you have thought about the problem and have some ideas of how you will accomplish it.
- An estimated time line. By now you should know that things always take longer than you expect. One rule of thumb that I use is to multiply the amount of time you expect it will take you by two, and then add one. So if you expect a task will take you two hours, allow 5 hours. If you expect a task will take you three days, allow a full week. I recommend that you allow yourself a full week to do Part 4 of this assignment.

Part 3: Do It! After you have submitted your written proposal, and I have approved it, get to work.

Part 4: Final Project Submission. You will be expected to provide the following:

Final Written Report. This report should include:

- Project Summary (1 to 2 paragraphs). Similar to the one in your proposal, this should summarize what you actually did and should be written for a general computer science audience.
- Project Description (1.5 to 4 pages not including figures). A detailed description of your robot, including what your robot does, how your robot accomplishes the task, and design decisions and their rationale (e.g. why did you use a touch sensor rather than an IR sensor, why you decided to use tracks instead of wheels). Note that you should not include your code in this section.
- Lessons Learned (0.5 to 2 pages). Describe the aspects of your project that were harder or easier than you expected. What advice would you give to other students who decide to do a similar project in the future?
- Appendix 1. Include an 8.5 x 11 reduction of your project poster (see below).
- Appendix 2. Include photographs of your robot that show the design.
- Appendix 3. Include a complete print-out of your code.

Project Poster. Design a 2 foot by 4 foot poster that explains your project. This poster should be targeted at a general audience of college students so that they will come away with an idea of what your project is about, as well as more details to catch the interest of those with a background in robotics. We will discuss effective posters in class.

Final Oral Report. Each group will give a 15 minute presentation (30 minutes for paired groups) including a demonstration of their system to the class. All team members must speak. Your oral report will be evaluated by your fellow classmates as well as the instructor.

Other Notes. You should be able to accomplish your project using only the materials in one robot kit. If you wish, you may spend a maximum of $10 of your own money *per team* on additional equipment (I expect that most teams will not spend any additional money, but this is to give you a little flexibility with your robot).

How to Evaluate Results

The most difficult part of this assignment is helping the students design a project that is of the right scope. The first goal should be something very straightforward. The second goal should be significantly more challenging, but it is important to make sure that students do not choose something that is going to be impossible.

Selected Projects

The following describes a few of the projects that students have developed in response to this assignment. Groups were encouraged to present their work at a local student research symposium. URLs are provided in the references with links to posters presented at this symposium for most of these projects.

Autonomous Vehicle With Artificial Intelligence. [1] This robot found negative obstacles and measured the depth of the drop.

HAL: Habitat Abstracting Logic. [2] Despite the contrived acronym, the students developed a very nice "recycle-bot" that would collect a can, pick it up using a pneumatic gripper system constructed from Lego parts, and deposit the can in the "recycle bin."

Rescue-Bot. [9] One of the students who developed this robot was a volunteer fire fighter. Rescue-Bot followed fire fighting rescue techniques to enter a room and retrieve a victim. It followed the wall using a left-handed search until a victim (a small Lego character) was found. When the robot has found the victim, it exited the room by heading towards the light at the door.

Self Recharging Robot. [11] This robot used a capacitor to simulate the robot's power supply. When the capacitor's charge was sufficiently low, the robot searched for its power source (a battery) and recharged the capacitor.

Tracking Optical Mechanism & Juvenile Electronic Roving Robotic Intelligence (T.O.M. & J.E.R.R.I.). [12] In this two team project, one robot (the "mouse") had a light source fixed to it and drove in a pattern unknown to the second robot (the "cat") which followed the light source on the mouse. The cat was programmed to move at a slightly higher speed than the mouse, and when it intercepted the mouse the two would stop. Beyond designing and programming the robots, these students made cat and mouse costumes for them!

Acknowledgments

The box measuring assignment was developed after discussions with Frank Klassner of Villanova University several years ago and is based on an assignment that he required of students in his AI class.

References

[1] Autonomous Vehicle With Artificial Intelligence, Designed by AJ Mieskolainen, Katherine Liong, and Michelle Glickman. Poster available at: http://elvis.rowan.edu/~kay/robots/stem/avwai/avwai.pdf

[2] HAL: Habitat Abstracting Logic, Designed by Melissa Bingham, Eric Cummings, Vincenzo Buttari, & Sherryll Chang. Poster available at: http://elvis.rowan.edu/~kay/robots/stem/stem2002/hal.pdf

[3] The Handy Board, http://www.handyboard.com/ [accessed March 2003].

[4] Interactive C Manual, http://www.newtonlabs.com/ic/manual.html, [accessed October 2003].

[5] K-Team: Khepera II Mini Robot, http://www.k-team.com/robots/khepera/ [accessed March 2003].

[6] Lego Mindstorms, http://mindstorms.lego.com/eng/default.asp [accessed March 2003].

[7] Martin, Fred G., The Art of LEGO Design, The Robotics Practitioner: The Journal for Robot Builders, Vol. 1, No. 2 (Spring 1995). Available at: ftp://cherupakha.media.mit.edu/pub/people/fredm/artoflego.pdf

[8] O'Hara, Keith J., and Kay, Jennifer S., Investigating Open Source Software and Educational Robotics, The Journal of Computing Sciences in Colleges, Vol. 18, No. 3, February 2003.

[9] Rescue-Bot, Designed by Melissa Short and Kathryn Thorsen.

[10] Robots and Robotics Accessories, http://www.activrobots.com/ [accessed, March 2003]

[11] Self Recharging Robot, Designed by Brian Spadafora and Loc Do. Poster available at: http://elvis.rowan.edu/~kay/robots/stem/stem2002/self_recharge.pdf

[12] Tracking Optical Mechanism & Juvenile Electronic Roving Robotic Intelligence (T.O.M. & J.E.R.R.I.) Designed by Joyce Barton, Zimmy Shah, Danielle Latorres, Hung Nguyen, Hae Ryong Park, & Quy Ngo. Poster available at: http://elvis.rowan.edu/~kay/robots/stem/tomjerri/