An Introduction to Robotics Course: Our Experience

A. A. Lopez and Christopher C. Olson
Computer Science Discipline
University of Minnesota, Morris
Morris, MN 56267
alopez@morris.umn.edu
olso1606@morris.umn.edu

Abstract

This paper describes our experience with the first offering of an Introduction to Robotics course at the University of Minnesota, Morris. The course is an upper division elective undergraduate course in computer science. We discuss the selection of the textbook and laboratory exercises for the course. We chose to use Lego's Mindstorm robots because of their affordability and the availability of supporting materials and software. This paper presents both the faculty member and the student's perspective of the course.

1 Introduction

This paper describes our experience with an Introduction to Robotics course at the University of Minnesota, Morris. This course is an upper division 4-credit elective course and it was taught for the first time during the Fall 2004 semester. At our institution this course is classified as a 'systems' course and it has as a pre-requisite our Models of Computing Systems core course. The course did not have a scheduled laboratory period. The course was proposed by a colleague that has since left our institution. The student co-author had worked with that professor and he is very interested in robotics and plans to attend graduate school in computer science and specialize in robotics.

1.1 Faculty perspective and expectations

Due to a late resignation, this course was assigned to a different faculty member during the early summer. The only exposure to robotics that this new faculty member had had was a weekend workshop on robotics offered at Alma College in Michigan and sponsored by the National Science Foundation. The departing faculty member had had some of his students exploring robotics and was helpful in transferring some of that information to the newly scheduled instructor.

The faculty member was aware that the students enrolled in the course were particularly interested in implementation details. However, it is not in our institution's tradition to offer courses that are exclusively implementation orientated. The first challenge was to find textbook(s) with the appropriate balance between theoretical principles and practical applications and implementation [1, 2, 3, 4]. The second challenge was to identify affordable hardware the students could implement their algorithms on. The third goal was to identify suitable theoretical and practical exercises for the students to do. The faculty member had been exposed to four programming environments during the workshop he attended, but he thought that two programming environments would suffice for this course. The faculty member had also idealized the performance of sensors coupled to inexpensive robots.

1.2 Student perspective and expectations

Half of the students in this course were juniors and the other half seniors; all were computer science majors. They had all taken, or were currently taking, the Algorithms and Computability theoretical core course and the Software Design and Development course which teaches students algorithm development, runtime efficiency, and Java programming. These key courses were very advantageous for the students, because they aided in design and implementation of their algorithms. Since all of the labs in this course dealt with programming in either Java or C, the students were well prepared for the lab exercises.

The students coming into this course understood that they would be working with Lego Mindstorms robotic kits beforehand, and were anxious to start constructing robots. They believed this course would include mainly hands-on projects. They expected that every scheduled day of class would consist of a lab session where they would listen to lectures for half of the time and work on their projects for the remainder of the time. They did not expect the course to be largely theoretical and lecture-based.

2 Goals of the course

The goal of the course is to give the students as broad of a perspective of robotics (as a science, within the constraint of it being a single undergraduate course) as possible. Robotic paradigms including the hierarchical, reactive and hybrid paradigms were to be studied. The biological foundations of the reactive paradigm and navigational techniques were thought to be important. Path planning was thought to be a topic that both students and faculty would enjoy. And there was the need for doing some hands-on exercises during a 65-minute class period or as homework assignments.

It was thought that it was important for the students to understand reflexive behaviors, perception and subsumption principles.

3 Implementation

Once the goals were set for this course, the instructor and his teaching assistant had to make some final decisions on which textbook and robotic kits to use, as well as on the types of exercises and labs to assign for the students taking the course.

3.1 Course textbook

The first challenge was to choose a suitable textbook for the course. Many of the textbooks that we encountered were both too technical and more suitable for a mechanical engineering course, or too applied and lacking in the general principles that we like to include in our courses or too artificial intelligence based. In the end the instructor chose to use the book Introduction to AI Robotics written by Robin R. Murphy, a professor in the Department of Computer Science and Engineering and in the Cognitive and Neural Sciences at the University of South Florida [4]. This book turned out to be quite suitable for our course. It discussed various theoretical concepts in robotics and artificial intelligence. It is structured around the three different types of robotic paradigms; hierarchical, reactive, and hybrid deliberative/reactive. It discusses the methodologies used in potential fields and subsumption, as well as behavioral schemas, schema theory, sensing techniques, architectural designs, obstacle avoidance, path planning, and map making.

The students found this book to be easy to read and not too complicated to understand. The author incorporates pictures and descriptions of robots from schools and government agencies such as MIT, Carnegie Mellon University, Georgia Tech, DARPA, and NASA to inform the readers that the theoretical robotic architectures discussed in the book have actually been implemented. She also includes informative graphs, models, and code snippets which help to illustrate her point. Some of her graphs and models were not clearly explained, but this lead to an intelligent group discussion on the meaning of her illustrations. Overall, this is an exceptional book for an undergraduate introduction to robotics course.

3.2 Robot building kit and sensors

We decided to buy four Lego Mindstorms Robotics Invention System 2.0 building kits for the students, the latest in the Mindstorms construction sets. At an educational price of $170.00, our total came to almost $700. This kit is equipped with two motors, two touch sensors, one light sensor, and a RCX programmable controller with 32 kilobytes of memory. These sensors were adequate for a couple of our projects. Later we decided to purchase two infrared (IR) relative distance sensors for $34 a piece from Techno Stuff [8] and borrowed one rotational sensor from the instructor's personal kit for experimentation. We found that the IR sensors worked very well at detecting an object directly in front of the sensor. It does not have a wide angle and/or distance detection range. They sometimes did not sense objects that were at an offset of three to four inches perpendicular with respect to the straight IR signal. We also purchased four 8" x 15" storage bins to store and organize the Lego parts for each group. We thought that it would be easier to keep track of the smaller parts if they were organized in some fashion. This idea worked well for the tiny parts, but it could not contain the RCX, all of the wheels, IR tower, and construction manuals which had to be carried separately. The students found this to be quite cumbersome and suggested not to use the storage bins, but to use the original Mindstorms packaging.

3.3 Programming assignments

We designed four related lab assignments, which increase in difficulty, for the students to practice implementing the system architectures and robotic paradigms stressed in the lectures, however, implementation of these architectural concepts was not required. Since there were eight students taking the course, they were split up into pairs and each couple received a Mindstorms kit.

3.3.1 First lab exercise

For the first lab, we had the students construct a robot from the kit, and program it to move forward one meter, stop, turn right 90 degrees, and back up a half a meter using two different programming environments. Figure 1 below shows how this lab is done.

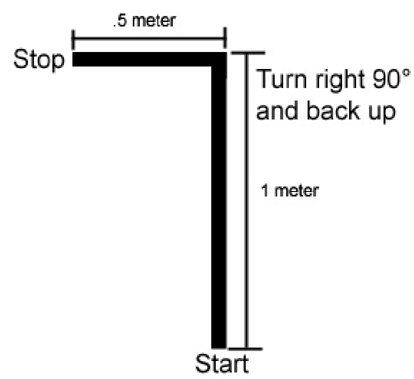

Figure 1: Diagram of the first lab assignment

The purpose of this assignment was to get the students acquainted with constructing Lego robots, controlling the navigation of the robot, and to become familiar with both LeJOS and NQC (Not Quite C) [6]. LeJOS is a Java-based replacement firmware for the Lego Mindstorms RCX brick [7]. It contains Java libraries which includes classes that control sensors, motors, timers, process threads, navigation systems, and many other functional aspects of the RCX. NQC is a C-based language which uses the default Mindstorms 2.0 firmware. It also contains functions and queries to manipulate the robot's sensors, timers, sounds, and LCD display. It is important to note that we do not teach C anywhere in our computer science curriculum, so the students had to learn this on the fly. For subsequent exercises, the student groups had a choice of programming in either C or Java and they split about 50-50 in their choice of languages.

3.3.2 Second lab exercise

The next step in the lab assignments was to introduce the use of light sensors to aid in following a line. The environment was setup whereas the robot would start at one end of the course and follow a black line made from ¾" wide electrical tape. As shown at point A of Figure 2, the robot will eventually come to a T-intersection and will have to make a decision whether to go forward or to turn left and follow that line. Eventually, the robot will encounter another intersection at point B and will have to make the correct decision on which line leads to the end of the course. The purpose of this lab exercise was to familiarize the students with the use of light sensors and the implementation of the Reactive paradigm. The overall student consensus on this lab was that it was not too difficult and that it was a great example of the SENSE – ACT paradigm.

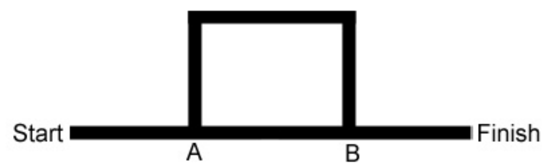

Figure 2: Line follower lab environment

3.3.3 Third lab exercise

To expand on the last lab, the instructor extended the environment and added obstacles to complicate navigation. The idea of this lab is similar to the last; follow the lines to the goal while avoiding obstacles. Figure 3 is the extended line following environment.

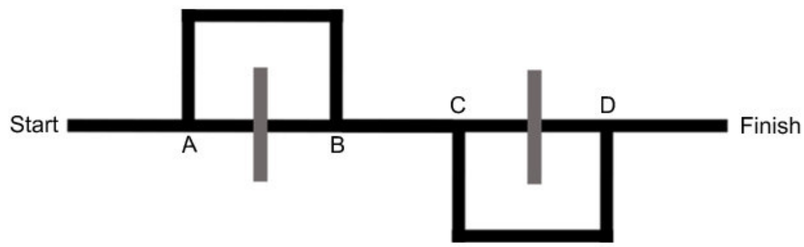

Figure 3: Extended line follower environment

It contains two obstacles between points A and B and C and D. Once the robot runs into either of these two obstacles, it must back up and find the path around the obstacle. This exercise requires more navigational intelligence than the previous lab exercise. Just like the last two projects, each student pair built and programmed a robot. One group even implemented a subsumtion architecture to aid in the robot's navigation to the goal state. The purpose of this exercise is to introduce the touch and IR distance sensors, as well as to influence the students to create more intelligent navigation control systems. The students found this course to be very acceptable, yet challenging.

3.3.4 Final lab exercise

The final robotic lab was exactly like that of the 2005 MICS robotic competition. It consisted of a 12 foot square world which included a 9 foot square grid in the center. Each grid square was 18 inches wide, which creates a 6 x 6 cell course for the robots to navigate through. The objective of this exercise was to find a black square somewhere within the course, while avoiding obstacles. The obstacles did not have to lie on the gridlines. They were placed in a random arrangement so that the robots had to be intelligent enough to detect and traverse around them while still searching for the black goal state. Starting at the origin, coordinate (0, 0), the robot would begin its search for the goal state. Once the robot found the black square, it had to produce a beep sequence which initiated its return back to the start state. Figure 3 below illustrates this environment in more detail.

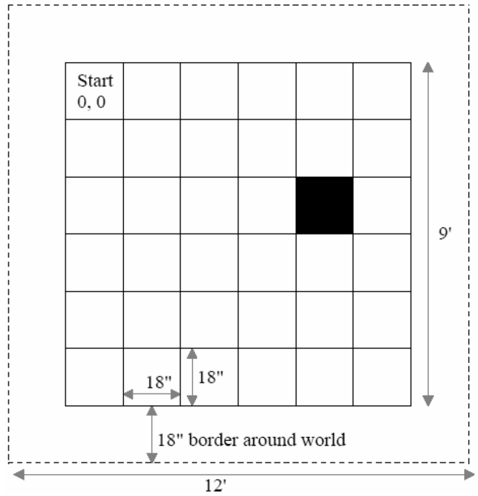

Figure 3: MICS grid environment for its robotic competition [5]

For this assignment, two groups combined forming two larger groups consisting of four members; therefore, each new group had two Mindstorms kits at their disposal for those who wanted to use two RCX bricks simultaneously. Even though the teams were larger, the students still found this project to be very difficult to accomplish, because the instructor set the deadline for this project to be due, at the end of the semester, nine days after it was assigned. This was not a sufficient amount of time for the students to complete all of the requirements for this project. Both groups implemented their navigational techniques and coordinate positioning systems, but neither robot was able to complete its objective when obstacles were introduced. Simply, there was not enough time for the students to perfect both navigation and obstacle avoidance control systems.

4 Robotic Competitions

The University of Minnesota, Morris was proud to host the MICS conference during the spring 2004, including research and paper presentations, the programming contest, and the robotic competition. A team of two highly motivated UMM Computer Science students, including the student co-author, planned and programmed a robot for this competition, but due to communication and registration problems, the group was, unfortunately, unable to compete in the robotic competition. However, the student co-author later became the teaching assistant for our Introduction to Robotics course due to his knowledge and experience with Lego Mindstorms. The two authors, together, agreed that the 2005 MICS robotic competition would make a challenging and educational final project for the course. The teams in the course could then use their robots to compete in the 2005 MICS robotics competition, thus increasing UMM's participation in the symposium.

5 Conclusions

The students seemed to have enjoyed the class. There was unanimous consensus on the part of the students that we should have done more hands-on (programming) exercises and cut back on the amount of theory covered in the class. Several students also mentioned that it would had been nice to have a variety of robots, including some of them that would be more sophisticated than the Lego's Mindstorms.

From the instructor's stand point, it was hard to figure out what would be a reasonable programming exercise. When some of the programming assignments were first mentioned in class, several students seemed to be overwhelmed by the complexity of the assignment. The instructor frequently had to reduce the expectations for the assignment.

The students believed the deadlines for the projects were not proportional to the amount of work needed for each project. The first lab was too easy, the second and third labs were drawn out too long, and the final project deadline was too short. A deadline of nine days was extremely short for this lab, which limited the student's ability to create a more functional robot. This is why the students were upset to hear that the instructor expected more from them.

A significant factor in this class is that the students are not used to writing memory conserving programs. The majority of their programming experience has been with object orientated programming languages in an environment with a significant amount of virtual memory. When faced with a 32 KB memory constraint, they had troubles thinking of memory efficient data structures that they could use in their programs.

Future offerings of this course should focus on identifying a larger set of suitable programming exercises, increase the number of hands-on exercises, try to find a textbook that has a briefer introduction to the theory of robotics and, hopefully, get an instructor with more experience in robotics.

References

[1] Lego Mindstorms Programming, B. Bagnall, Prentice Hall PTR, 2002

[2] Introduction to Robotics: Mechanics and Control, J. Craig, 3ed, Pearson Prentice Hall, 2005

[3] Creative Projects with Lego Mindstorms, B. Erwin, Addison Wesley, 2001

[4] Introduction to AI Robotics, R. Murphy, MIT Press, 2000

[5] http://mics.cs.uwec.edu/robot_rules.pdf

[6] "Not Quite C." 11 February 2004. http://bricxcc.sourceforge.net/nqc/.

[7] “LeJOS: Java for the RCX.” 23 October 2004. http://lejos.sourceforge.net/.

[8] Sevcik, Pete. “Lego Mindstorms compatible sensors and controls.” Techno-Stuff Robotics. http://www.techno-stuff.com/.