An Undergraduate Course in Robotics and Machine Intelligence*

Ben A. Juliano and **Renee S. Renner**
Institute for Research in Intelligent Systems
California State University, Chico
400 West First Street
Chico, California 95929–0410

Abstract

In this paper, the authors present their experience in planning and curriculum development for a course in Robotics and Machine Intelligence. The authors discuss issues, solutions, and recommendations regarding administration of the course, robotics equipment used, student activities, and outcomes assessment. Future plans for the course, as well as ideas for other similar courses, are also outlined.

Introduction

In the Fall 2003 semester, the authors were awarded a National Science Foundation (NSF) Major Research Instrumentation (MRI)/Research in Undergraduate Institutions (RUI) grant for 2003–2006. The primary focus of the grant was to acquire robotics equipment for use in intelligent systems curriculum development, research, and outreach activities. Through this grant, the authors set up the Intelligent Systems Laboratory, or ISL (ISL 2006), to facilitate the development of inter–disciplinary intelligent systems courses and provide exciting collaborative research possibilities for students and faculty. The authors also formed the Institute for Research in Intelligent Systems, or IRIS (IRIS 2006), to oversee the ISL; in particular, IRIS handles policies and management issues pertaining to the ISL. One of the courses developed through this NSF grant was CSCI 585, *Robotics and Machine Intelligence*. This cross–disciplinary, project–centered course has been offered in four of the past five semesters beginning in Spring 2004. Each semester it is taught, several "exhibitions" give students the opportunity to showcase their robot designs and behavior/control algorithms.

In this paper, the authors present their experience in curriculum development, assessment, and curriculum adjustments with CSCI 585. The paper is organized as follows: first, the course organization and related administrative issues are presented; second, a brief description of the robotics platforms and programming environments used is given; third, the activities and "exhibitions" used in the course are described; fourth, a brief discussion that pulls together the first three sections is presented; and finally, a summary of lessons learned, along with plans and recommendations for future offering of this course, is presented.

Course Organization

Classroom

The authors are fortunate to have access to a computer laboratory within their college to teach this class in. The lab offers a lot of floor space for robotics exhibitions, and ample desktop space between computer terminals for working on robots for specific projects. Although the machines in the lab are relatively old, have inconsistent desktop settings, and have no Internet connectivity, the authors believe that the lab setup works well for a class like this. Other space needs pointed out elsewhere (Sutherland 2000) are also available through the lab: room security since the lab houses many test and measurement equipment used for other engineering classes, and in–room security through lockable cabinets.

Scheduling

CSCI 585 has been offered in four of the past five semesters. In three of those four semesters it was taught, the class met twice a week (Tuesdays and Thursdays) for one hour and 15 minutes. The fourth semester CSCI 585 was taught, the class met once a week (Thursday nights) for three hours. The authors prefer the latter scheduling since it has been their experience that 75 minutes is not sufficient to cover enough material and provide ample build, setup, and cleanup time. A three–hour time slot allows for lectures, robot building, and programming, or for one to two runs for robot exhibitions.

Instructional Support

The original proposal for CSCI 585 is that it would be co–taught by at least two professors: a computer science professor and an engineering professor. This setup has been recommended elsewhere (Maxwell & Meeden 2000). The authors co–taught this course the first time it was offered. (A third professor, from engineering, was scheduled to co–teach as well, but that ptofessor suffered a medical condition and was out for most of the semester.) In the next three

---

*This work is supported by the Intelligent Systems Lab at California State University, Chico and was partially funded by NSF MRI/RUI grant EIA–0321385.

semesters this course was taught, the professor of instruction has always been one of the authors, with the other author assisting with administrative details.

Fortunately, the authors have graduate and undergraduate research assistants working with them in the ISL. These students are also instrumental in assisting in the instructional delivery of CSCI 585. Each semester, the ISL Research Team will have at least one engineering student who is the "hardware consultant" for the team. The ISL Research Team is an important resource since they assist in troubleshooting hardware and software issues in the classroom, they assist in developing and setting up exhibition courses, and they get involved in some curriculum development and planning.

Prerequisites and Demographics

Teaching an inter–disciplinary robotics course is challenging – there are a number of inherent issues and resource requirements for such an undertaking (Murphy 2000). CSCI 585 was designed to be open to all majors in the College of Engineering, Computer Science, and Construction Management at CSU, Chico. The College consists of five departments whose combined offerings encompasses ten programs. The authors expect the minimum programming experience from computer science (CSCI) majors to include an upper division course in data structures and algorithms and a course in assembly language programming. For non–CSCI (engineering) majors, the minimum requirement is having taken a college–level programming course and a course in processor architecture and assembly language programming.

The engineering and computing sciences are primarily male–dominated fields — this is exhibited by the gender distribution of the class given in Table 1. Additionally, since CSCI 585 is an upper–division course, Table 1 shows that most students that take it are Seniors. With some additional requirements, graduate students (GR) can also earn graduate credit when they take the course. The high enrollment in Spring 2004, compared to the low enrollment in Spring 2005, reflects the effects of a gradual decrease in full–time equivalent faculty (FTEF) in computer science. This is primarily the reason why the course is currently being offered only once a year.

Table 1: Gender and classification distribution.

	Gender		**Classification**				
Sem/Yr	**F**	**M**	**FR**	**SO**	**JR**	**SR**	**GR**
Sp/2004	1	32	0	0	3	20	10
Fa/2004	4	16	0	0	2	10	8
Sp/2005	1	7	1	0	1	5	1
Fa/2005	3	16	0	0	3	12	4

Table 2 summarizes distribution of majors. Majors listed are computer information systems (CINS), computer science (CSCI), electrical and computer engineering (EECE), and mechatronic engineering (MECA). "Other" represents a major other than the first four. Most of the students that take CSCI 585 are computer science majors. These majors are enthusiastic to take this class because it is the only class where they can highlight projects they have developed as tangible artifacts. It has been a long–standing issue with computer science majors that, since they are mostly involved with software projects, they typically have non–tangible projects that are difficult to share and get the layperson excited about.

Table 2: Major/Degree classification distribution.

	Major				
Sem/Yr	**CINS**	**CSCI**	**EECE**	**MECA**	**Other**
Sp/2004	1	22	3	6	1
Fa/2004	5	12	0	2	1
Sp/2005	2	4	0	1	1
Fa/2005	2	14	0	3	0

In Spring 2004, the student whose major is listed as "Other" was an applied math major earning a minor in computer science. In Fall 2004, the student whose major is listed as "Other" was a chemistry major with an "unclassified" graduate status in computer science. In Spring 2005, the student whose major is listed as "Other" was an undeclared freshman who happened to have extensive robotics experience. Since then, this student has decided to earn a degree in computer science.

Robot Platforms

The authors have been using the same two robotics kits each semester CSCI 585 is offered: the *LEGO Mindstorms Robotics Invention System 2.0* (Klassner & Anserson 2003; The LEGO Group 2006c) and the *Parallax Boe–Bot* (Parallax Inc. 2006b). The ISL has thirty Mindstorms kits – some are made available to students, and a few are used by the instructional team for proof–of–concept in preparing activities and challenges for the course. The authors make arrangements with Parallax, Inc. for a bulk, academic discount (from around $140 in Spring 2004 to $120 this Fall 2006, which includes shipping and handling). So, instead of having a textbook for the course, students are required to purchase their own Boe–Bot.

LEGO Mindstorms RIS 2.0

The students are first introduced to the ISL's LEGO Mindstorms kits (Part #3804 with over 700 parts). LEGO basics (part names, units of measure, etc.) are discussed and students get familiar with the kits by first building a "Tankbot" from the kit's *Constructopedia*. Students are also exposed to two ways of programming their LEGO creations.

Robotics Invention System (RIS) 2.0 The initial programming is done using the LEGO Mindstorms RIS 2.0 (The LEGO Group 2006b), a simple, click–and–drag development environment.

BricxCC: Bricx Command Center for NQC Next, students are introduced to BricxCC (Hansen 2006), a GUI for NQC (Hansen & Baum 2006) which is a subset of C.

Parallax Boe–Bot

The other robot platform used in the class is the Parallax Boe–Bot (Parallax Inc. 2006b). The Boe-Bot has servo motors and a printed circuit board that allows for elementary circuit design and testing. The kit comes in either serial (Part #28132) or USB (Part #28832) versions. Each kit contains a BASIC Stamp 2 (BS2-IC) module, Boe-Bot robot kit (chassis, wheels, servos, whiskers, etc.), software, interface cable, and the *Robotics!* text (Parallax Inc. 2004) with projects. Students write PBasic 2.5 (Parallax Inc. 2005) code to control their Boe–Bots.

Activities and Exhibitions

The authors typically give around 10 activity sheets to CSCI 585 students each 16–week semester. Some of these activities involve exhibitions that span two weeks in the semester. Others are hands–on activities that facilitate getting the students familiar with the robotics platform they are using. Each exhibit specification sheet outlines AIMA–style (Russell & Norvig 2003) PEAS (Performance measure; Environment; Actuators; Sensors) description of the task environment.

LEGO Mindstorms

There are typically two exhibitions for the LEGO Mindstorms platform. Depending on enrollment and on availability of ISL Mindstorms kits (kits may sometimes be used for ISL research and outreach activities), these LEGO exhibitions required participation in teams of two or individually.

Mission Mars I The first LEGO exhibit is called “Mission Mars I: Martian Storm Rescue.” The exhibit simulates a rescue situation where a Martian storm has disconnected all communication lines to a space station. Some space scientists exploring the Martian terrain got lost and are potentially hurt — they must be rescued and brought back to the space station before the next storm.

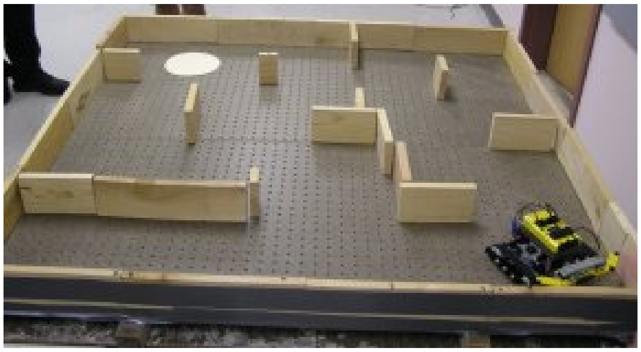

Figure 1: The *Mission Mars I* 3D maze.

The Martian environment is simulated using a 3D maze made from pegboards and reconfigurable walls that were designed and prepared by an ISL student research assistant (see Fig. 1). Typically, a 3D maze based on two pegboards was sufficient for the exhibit. The stranded space scientists were represented by a white disk on the maze and each team had a 3–minute time limit to locate the scientists. Most students use a tracked zero turning radius (ZTR) roverbot with double bumpers for detecting maze walls, and a light sensor to detect the target disk. Students were allowed to program their roverbots using RIS 2.0 (The LEGO Group 2006b), and the students quickly figure out its limitations.

The configured, simulated Martian environment is set upon a couple of saw horses for accessibility. Since the reconfigurable walls on the 3D maze were merely pegged on a pegboard, the students’ robots would sometimes push the walls slightly. This was a problem with the outer walls, but this was easily solved by duct taping the outer walls together for stability (see bottom of Fig. 1).

Mission Mars II The second LEGO exhibit is called “Mission Mars II: Rendezvous with a Rescue Shuttle.” The exhibit simulates a continuation of the search and rescue in the first LEGO exhibit. After the scientists have been rescued, they need to be transported back to the space station via a rescue shuttle. Unfortunately, another bad storm hit Mars and the landing strip and service areas are littered with debris, prohibiting a safe landing by the rescue shuttle. The landing strip, and the path to and from the service area, are marked by a distinct line on the Martian surface.

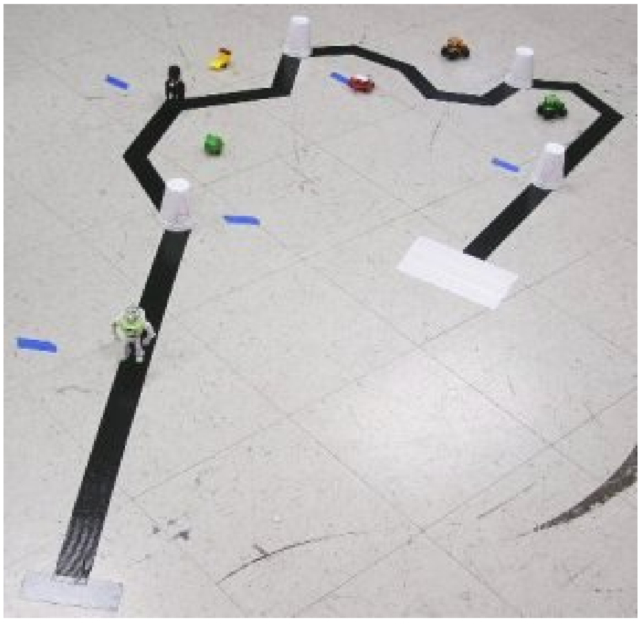

Figure 2: The *Mission Mars II* line–following course.

In this exhibit, the Martian environment is simulated using black duct tape that is taped on the classroom floor (see Fig. 2). Students need to design a wheeled Line Follower with Obstacle Sweeper (LFOS) roverbot that will (1) follow a service line (used to direct service vehicles) to the landing strip, (2) follow the line on the landing strip, and (3) stop at the end of the landing strip to help load the scientists onto an awaiting rescue shuttle — all within a 3–minute time limit. Obstacles on the line are swept by a moving sweeper on the LFOS roverbot. A light sensor is used to follow the line. In this exhibition, students were required to program their LFOS roverbots using NQC (Hansen & Baum 2006; Hansen 2006).

Parallax Boe–Bots

There are typically two exhibitions for the Parallax Boe–Bot platform. Before getting into the first Boe–Bot exhibition, the authors have three activity sheets that introduce the students to servo control, distance control and ramping, and the use of photoresistors and IR proximity detectors (IRPD) for navigation. These activities provide a gentle introduction since I/O for the Boe–Bots need to be designed on a breadboard. In this section, the authors present five exhibitions that they have used for this platform.

2D–Maze Cave Rescue This exhibition simulates a cave rescue. Some cave explorers have been trapped in a cave for more than a week. Their location is a tall cavern with a hole allowing some sunlight in, but the sun is about to set and the next day is not looking good. Students need to design a Boe–Bot that will use IRPD sensors to navigate a 2D maze representing the cave (see Fig. 3). The survivors' location is indicated by a flashlight pointed towards the floor, simulating sunlight coming into the cave — students need to use a photoresistor to detect the flashlight. Each student has a 3–minute limit in attempting the rescue.

Figure 3: The *2–D Maze Cave Rescue* course.

Convoy This exhibition is a modified version of a contest outlined in the *Robotics* text (Parallax Inc. 2004). Instead of individual Boe–Bots following a line, a convoy of three Boe–Bots per team will traverse the maze in shadow vehicle form (see Fig. 4). Although most of the ideas for the coding are available through the *Robotics* text, this is a good team exercise for the students. Team scores are based on the time needed to complete the course, minus percentages based on specified accomplishments (*e.g.* stopping at the final designated area, starting on non–tactile command, all Boe–Bots in the convoy finish). The authors used this exhibition in Spring 2004.

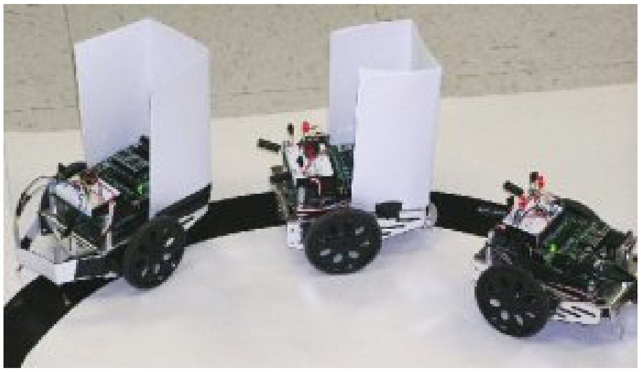

Figure 4: A student team's Boe–Bots maneuvering the *Convoy* course.

3D Grid Maze This exhibition is preceded by an activity designed to give students the experience of building their own sensors. This activity, a *Sensors Workshop*, was first introduced in Fall 2004 when the authors needed wheel encoders for a 3D Grid Maze, which involved navigation through a walled maze with floor cells marked by black tape (see Fig. 5). The wheel encoders are used to facilitate limited localization by the Boe–Bots. Students, particularly the CSCI majors, had fun with the soldering portion of the sensors workshop — an activity they do not get to do within their own program. The biggest challenge for the students was to develop a scheme that will use the wheel encoders for localization while storing a map of the maze — all within the hardware limitations of the BS2. The PBasic programming language also has its limitations.

Figure 5: The *3D Grid Maze* course.

Micromouse Challenge Due to the issues encountered with the 3D Grid Maze exhibition, the authors decided to try a different approach: use the ISL's collection of Compass AppMod (Parallax Part #29113) electro–mechanical compasses for localization. This idea was tried in Spring 2005 when the enrollment for the course was low (see Table 1) since the ISL only had 10 compasses available. The maze was modified from a 3D grid maze to a simple 2D maze consisting of black electric tape on a whiteboard panel resting upon a couple of saw horses (see Fig. 6). In this exhibition, students used IRPD sensors to detect the walls on the maze, a photoresistor to detect the target flashlight, and a Compass AppMod for navigation and localization. The biggest difficulty encountered, in addition to the hardware restrictions of the Boe–Bot, was the sensitivity of the Compass AppMod to electro–magnetic fields (EMFs). This was a major problem in the computer lab where EMFs were generated by the computers and wiring in the room.

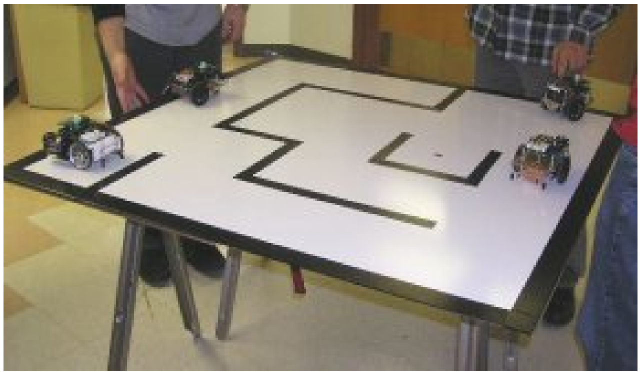

Figure 6: The *Micromouse Challenge* course.

Team Garbage Collection In Fall 2005, the authors modified the exhibition showcasing localization by reverting back to a *Sensors Workshop*. Students built a color sensor from three photoresistors and three LEDs that take RGB readings from the floor. The challenge was to try to identify as many “targets” (red squares) from an arena with 19 strips of varying shades of gray from one “home” corner to the other (see Fig. 7). The class was divided into two teams (that semester, they called themselves the “Jedis” and the “Siths”). Pairs of students, one from each team, had three minutes to identify as many distinct targets as possible. Each time a target was identified, a Boe–Bot needed to return to its “home” corner before seeking out another target.

The biggest issue with this exhibition was the difficulty in setting up an arena so that the strips of gray had a consistent shade change while still being distinct enough for the Boe–Bots to rely on for localization. For the arena shown in Fig. 7, it was observed that the team that took the black corner had a more difficult time.

Discussion

Recognizing the challenges of teaching robotics (Murphy 2000) in an undergraduate institution within a state university system, the authors continue to develop the delivery and content for CSCI 585 at CSU, Chico. Instructional support is a big issue. When the authors team–taught the course in its first offering last Spring 2004, the authors still struggled in trying new ideas and discovering platform limitations on–the–fly. Based on their experience, although one or the other author is typically assigned to teach CSCI 585, the other author tends to help out with at least the exhibitions since these require a lot of preparation. The authors also rely on instructional support from students (Rosenblatt & Choset 2000), typically from their research group. These students are instrumental in helping out with exhibition setup and testing, as well as in hardware and software troubleshooting. Training has not been an issue since these students either took the course previously or have been involved with ISL research and/or outreach activities with the authors.

Figure 7: The *Team Garbage Collection* course.

Table 3 summarizes the key similarities and differences between the LEGO Mindstorms and Parallax Boe–Bots. The variability between the Mindstorms and the Boe–Bots provide students the opportunity to make mechanical and control design decisions based on platform–specific restrictions. Based on the authors’ experiences, students had a better classroom experience working with the Mindstorms first before working with the Boe–Bots. To alleviate the regular demands for fresh batteries, students have been allowed to use rechargeable batteries (provided by the ISL) on the Mindstorms kits. Power requirements on the Boe–Bots prohibit the use of rechargeables. Students quickly learn how to conserve battery power by shutting off their robot’s power when not in use, by using wheeled robots instead of tracked ones, and on one occasion a student replaced the battery pack on their Boe–Bot to accommodate size D batteries instead of the usual size AA. Additionally, the Boe–Bots are not only reasonably priced, but also provide an avenue for supporting a “local” company. The authors suggest that students purchase a small, inexpensive toolbox (for around $7–$12) to safely store and transport their Boe–Bots.

The authors have considered other potential, alternate robot kits for use in this course. Although the current ISL Mindstorms kits are sufficient for use as classroom–provided kits in this course, one possible replacement is the recently announced LEGO Mindstorms NXT (The LEGO Group 2006a). Potential candidates to replace the Boe–Bots are the Vex Robotics Design System (Innovation First Inc. 2006) and the Parallax SumoBot Robot Kit (Parallax Inc. 2006a).

Table 3: LEGO Mindstorms vs. Parallax Boe–Bot features.

Feature	LEGO Mindstorms	Parallax Boe–Bot
processor	Hitachi H8 series 8–bit, 16 MHz 32 KB RAM 16 KB ROM	BASIC Stamp 2[1] 8–bit, 20 MHz 32 B RAM 2 KB EEPROM
input	four buttons three sensor ports IR interface	16+2 dedicated serial I/O (USB available) breadboard
output	built–in LCD screen one internal speaker three actuator ports	
ISA	57 instructions variable firmware multi–tasking support	33 instructions
coding	RIS 2.0 GUI NQC, Java, Lisp	PBASIC

Having two exhibitions per platform seems to be working well for this course. Whenever possible, the authors recommend students work in teams of two with the Mindstorms platform. Since the Boe–Bot is currently used as the course textbook, teamwork in a Boe–Bot exhibition would require a task similar to the Convoy (see Fig. 4) or the Garbage Collection (see Fig. 7). The authors are still looking for ways to introduce localization and navigation challenges in both the Mindstorms and Boe–Bot platforms. For example, this semester, one of the authors introduced the students to rotation sensors for the LEGO Mindstorms. The idea is to use these sensors to facilitate internal mapping and backtracking for navigation problems.

Some students who have taken this course have chosen robotics for their senior project or their graduate project. The inter–disciplinary nature of this course provides students with a unique opportunity to collaborate with students from other disciplines. The authors encourage the engineering majors to consult with a computer science major regarding their programming questions; likewise, computer science majors ask the engineering majors their hardware questions (specially on the Boe–Bot where I/O circuits need to be designed). The authors also have students who got so interested in robotics and intelligent systems after taking this course that they either got internships or are now working with robotics–related companies or national research labs.

Summary, Conclusions, Recommendations

The authors acknowledge that without their NSF grant award, they would not have had the chance to offer this Robotics and Machine Intelligence class the way it is currently being offered. With so many challenges associated with teaching robotics (Murphy 2000), one of the best ways to get things started is to get a grant to acquire the necessary initial robotics equipment. Team–teaching the course is ideal; having trained student assistants as instructional support staff is highly recommended. The authors are considering the possibility of integrating participation in robotics competitions as part of the course experience (Maxwell & Meeden 2000). The authors also recommend giving students the opportunity to work with more than one robotics platform in a class like this. The experience of working with two robot kits allows students to compare and contrast the capabilities, limitations, and related issues associated with the design and implementation of autonomous robots.

Acknowledgments

Special thanks to the ISL Research Team for the Spring and Fall 2004, and the Spring and Fall 2005 semesters for their involvement in the the CSCI 585, *Robotics and Machine Intelligence*, activities and exhibitions.

References

Hansen, J., and Baum, D. 2006. NQC: Not Quite C. http://bricxcc.sourceforge.net/nqc/.

Hansen, J. 2006. Bricx Command Center 3.3 for NQC. http://bricxcc.sourceforge.net/.

Innovation First Inc. 2006. Vex Robotics Design System. http://www.vexlabs.com/vex-robotics-design-system.shtml.

IRIS. 2006. Institute for research in intelligent systems. http://iris.ecst.csuchico.edu.

ISL. 2006. Intelligent Systems Laboratory (ISL). http://www.gotbots.org.

Klassner, F., and Anserson, S. 2003. Lego mindstorms: Not just for K-12 anymore. *IEEE Robotics and Automation Magazine* 10(2):12–18.

Maxwell, B. A., and Meeden, L. A. 2000. Integrating robotics research with undergraduate education. *IEEE Intelligent Systems* 15(6):22–27.

Murphy, R. R. 2000. Guest editor's introduction: Robots and education. *IEEE Intelligent Systems* 15(6):14–15.

Parallax Inc. 2004. *Robotics with the Boe-Bot v2.2*. Rocklin, CA: Parallax, Inc.

Parallax Inc. 2005. *BASIC Stamp Syntax and Reference Manual, version 2.2*. Rocklin, CA: Parallax Inc.

Parallax Inc. 2006a. SumoBot Robot Kit (Part #27400). http://www.parallax.com.

Parallax Inc. 2006b. The Boe–Bot Robotics Kit (Part #28132 or #28832). http://www.parallax.com/.

Rosenblatt, M., and Choset, H. 2000. Designing and implementing hands-on robotics labs. *IEEE Intelligent Systems* 15(6):32–39.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach, 2/E.* Prentice Hall.

Sutherland, K. T. 2000. Undergraduate robotics on a shoestring. *IEEE Intelligent Systems* 15(6):28–31.

The LEGO Group. 2006a. LEGO Mindstorms: NXT Technology Overview. http://mindstorms.lego.com/Overview/.

The LEGO Group. 2006b. LEGO Mindstorms Robotics Invention System (RIS) 2.0. http://mindstorms.lego.com/ris.

The LEGO Group. 2006c. LEGO Mindstorms Robotics Kits. http://mindstorms.lego.com/.