Designing and Implementing Hands-On Robotics Labs

Michael Rosenblatt and Howie Choset, Carnegie Mellon University

THE HANDS-ON, HEADS-ON LABORATORY COMPONENT TO A NEW ROBOTICS COURSE DESCRIBED HERE PROVIDES A SELF-LEARNING EXPERIENCE FOR ENGINEERING AND COMPUTER SCIENCE UNDERGRADUATES, ENRICHING THEIR ROBOTICS EDUCATION AND TEACHING INTERDISCIPLINARY TEAMWORK.

AS PART OF A RIGOROUS ROBOTics curriculum at Carnegie Mellon University, we have developed and implemented open-ended design and construction lab experiences. In the long term, we want to develop similar lab-like experiences where students design and build artifacts to complement conventional science and mathematics education. Reporting on a first step in that direction, this article focuses on implementation of design and construction experiences for a robotics course.

Our choice for robotics was rather serendipitous: we are roboticists trying to teach robotics. Fortunately, because of its flexibility, robotics education provides an ideal setting for testing new educational approaches. Unlike traditional fields, robotics is still emerging. Relatively few programs exist at the graduate level—even fewer at the undergraduate level—and the courses that do exist are still new, open to rapid change, and amenable to new approaches. Course goals can change from year to year as the field adopts new technologies (especially computer microcontrollers) and theories.

Developing the course

In 1998, we introduced a lecture-based introductory robotics course titled General Robotics. For their weekly assignments, students programmed simulations of particular robot tasks. In an optional course component, teams of students could define and carry out an independent project. Approximately half of the 70-student class undertook projects, with the remaining students opting to write a series of reports. Of those who decided against the projects, many were afraid of being overwhelmed because they lacked prior project experience. Those who did undertake the projects gained valuable experience, but what they learned did not necessarily promote the course's curricular goal, which was to give a broad introduction to robotics in preparation for more advanced course work.

Before the semester ended, the two of us (Michael Rosenblatt, then a student in the course, and Howie Choset, the instructor) began discussing ideas to extend the hands-on component to all students in the course. Principally, we wondered how we could use a hands-on component to directly complement the curriculum, as opposed to the independently defined projects, which generally did not parallel the curriculum. This article describes our work over the past two years to design and integrate hands-on laboratory assignments within the existing general robotics framework.

What makes this work unique from other hands-on undergraduate robotics courses is its goal of teaching a robotics-specific curriculum. We are roboticists, teaching robotics to students who intend to study further into the field of robotics and possibly pursue careers in this or related fields. Other work—including the MIT 6.270 Robot Design Contest[1] and Case Western Reserve University's Autonomous Robotics Class[2]—state much broader educational goals, such as providing

1094-7167/00/$10.00 © 2000 IEEE

engineering design experiences and opportunities to tackle real-world issues in engineering. Although these aspects are apparent in our course—and in many cases we have designed assignments to foster these aspects—we have considered them under the umbrella goal of teaching the fundamentals of robotics (motion planning, kinematics, and so forth).

We hope that this article helps educators at other institutions who are interested in implementing similar hands-on robotics courses (see the "Replicating this course" sidebar). We strongly encourage you to visit the General Robotics Web site at http://generalrobotics.org.

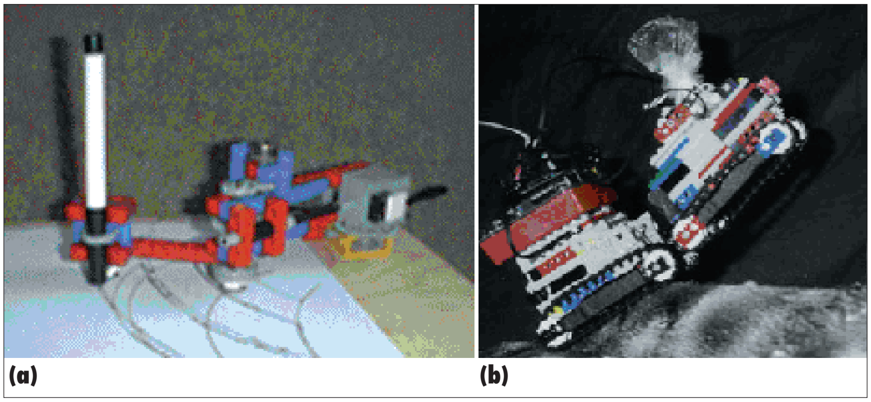


Figure 1. Lego bricks used to construct (a) a robot arm and (b) a mobile robot.

Educational considerations

Although the 1998 course was well received, it was certainly missing something—the build-it component. Students from computer science, electrical engineering, mechanical engineering, and related fields are all drawn to robotics because they want to see a mechanism interact with its environment: they want to see something process input and perform motion. So, instead of simply writing a program to demonstrate a robot arm's kinematics, students wanted to actually derive the kinematics, write the program, and build the arm. Initially, we wanted to serve this need in the context of a strict curriculum. We chose LEGO bricks because you can relatively easily "rapid prototype" with them without recourse to a slower, more expensive process (see Figure 1).

Instructionist and constructionist philosophy. Two main parameters of the 1998 course—class size and a rigorous curriculum—required us to teach the course under an instructional model. The course had 70 students and covered a specific list of topics that students had to learn to continue onto higher-level robotics courses. The instructional, or lecture-based, model suits these constraints well. The information passed from instructor to students can be well organized and controlled. Students all sit in a lecture hall and listen, hopefully gaining the knowledge. Learning is definitely one-way. Students get to put new knowledge and ideas to practice through written assignments, and in the case of our 1998 offering, also through some computer programming assignments simulating theories.

When designing the hands-on component, we considered a learning philosophy called *constructionism*,[3] an extension of Jean Piaget's constructivist developmental theory that emphasizes the importance of constructing artifacts as a catalyst for constructing mental models. In this approach, learning is student-centered—the information development and acquisition occurs through the student's actions and explorations. Often, when a student constructs an artifact, the intended result is not achieved initially and the student engages in an iterative debugging process that can be most informative.

While we have remained instruction-centered by retaining a significant lecture portion in the course, we have tried to retain some of the ideas behind constructionism when designing the course's hands-on laboratory portion. In striking a balance between the conventional and constructionist approaches to education, we have adopted a term called *directed constructionism*. This paradigm is directed in that the course follows a well-ordered sequence of topics in a course and constructionist in that assignments are not simple derivations or programs, but rather involve construction of open-ended projects that complement and reinforce the curriculum's principles.

Because of its physical nature, robotics education is also a good fit for directed constructionism. Many ideas introduced in a robotics course (through reading or lecture) can be physically implemented as a mechatronic system. The design space of mechatronic systems is immense, providing opportunities for rich variation in designs for a specific end goal. Also, creating these systems requires skills from various disciplines (mechanical engineering, electrical engi-

Replicating this course at other institutions

We hope that the ideas presented here will help those at other institutions replicate this course or design similar courses. The following should serve as a recipe of resources.

Financial resources: (n/3 * $600 + $800) + ($n$/6 * 3 * $6.75 *14). The first part of the formula accounts for equipment, where n equals the number of students (three students to a team), $600 per kit, and $800 for fixed costs in building demo environments and a planetary landscape. The second part of the equation accounts for teaching assistants: our ratio is now 1:6, working three hours per week at $6.75 per hour, for 14 weeks of the semester.

Laboratory resources. Figure T/4 workstations, where T is equal to the number of teams. A workstation consists of a computer with Interactive C installed and a work surface. Electronic test equipment (oscilloscope and multimeter) is helpful.

Teaching assistants. At least one teaching assistant must have sufficient experience with equipment to troubleshoot problems and give advice. This includes familiarity with the hardware and software systems of the microcontroller, sensor systems, mechatronics, and LEGO construction. (Some onlookers have considered this last item as trivial for university-level engineering students. They are mistaken.) To acquire this level of experience from little prior experience, or related subject (but not equipment) experience, a TA candidate should complete all our labs. Students who have taken the course will be qualified to be assistants for the next offering.

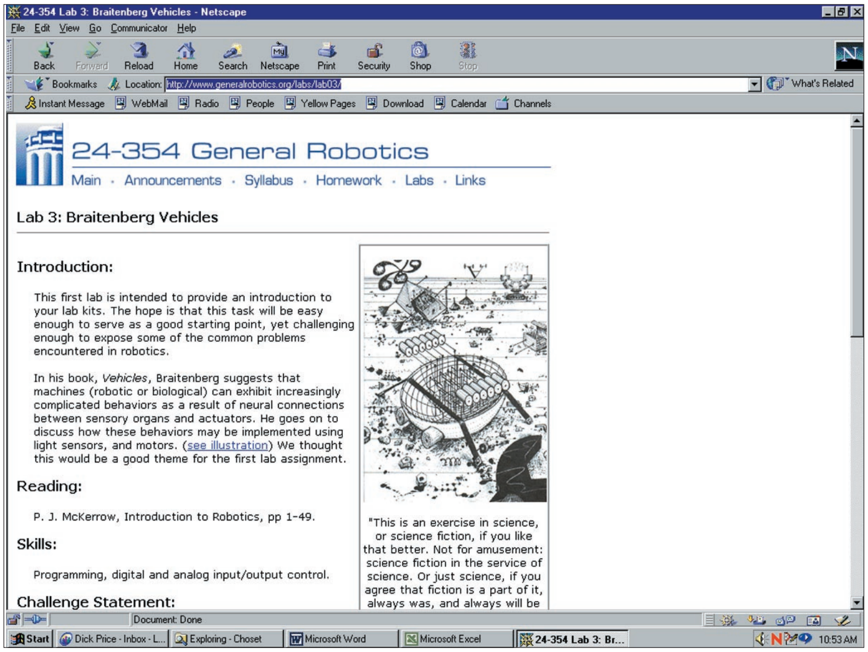
Table 1. General Robotics syllabus.

Week	Topics	Assignment
1	Robotics overview, vision	Written homework
2	Vision	Lab 1: Vision on Sun workstations
3	Intro to AI, hand out kits	Lab 2: Behavior-based robots
4	Motion planning: potential functions, roadmaps	Lab 3: Low-level mobile base control
5	Motion planning: cellular decomposition, configuration space	Lab 4: Motion planning
6	Mobility and nonholonomic path planning	Lab 5: Ackerman steering
7	Mechanisms	Written assignment
8	Review and midterm	Lab 6: Mars rover
9	Matrix transformations	Continue Lab 6
10	Forward kinematics	Lab 7: Forward kinematics
11	Forward and inverse kinematics	
12	Inverse kinematics and sensors	Lab 8: Inverse kinematics
13	Sensors and review for final exam	
14	Final exam	

neering, computer science, and more). Indeed, this interdisciplinary nature suggests the constructionist approach, in which students with specialized skills can gain knowledge from those with different backgrounds by working together through a creative design process.

Course demographics. In the 1998 course offering, students came to us with a diverse range of class statuses and backgrounds. This diversity guided us in applying the new course model to the 1999 offering. The course is open to all students in the university, although they must have had introductory C programming and calculus. Still, the overwhelming majority of students enrolled were from mechanical engineering, electrical engineering, and computer science. It is intended for the junior level, but approximately 30% sophomores and 20% seniors enrolled. Female enrollment was about 18%. Experience levels ranged from sophomores just learning basic robotic concepts to seniors who had already spent three or four years working in our university's research robotics labs. The 1998 offering had approximately 70 students. (These demographics remained relatively unchanged in 1999.) Our design challenge thus became to develop labs that would engage students with varied levels of experience and draw on the students' different backgrounds.

This challenge became an opportunity. Partially owing to budget constraints that limited the number of Lego kits we could purchase, we required that students work in three-person groups—nominally an electrical engineer, a mechanical engineer, and a computer scientist. As perhaps the first course whose undergraduates had to work with undergraduates in another major, our approach let students learn how to speak with people in different disciplines and see their individual contribution in the context of a big-picture project. During the 1999 course offering, we often wondered if the class's major contribution was to provide a multidisciplinary teaming experience for undergraduates.



Lab 3: Braitenberg Vehicles

Introduction:

This first lab is intended to provide an introduction to your lab kits. The hope is that this task will be easy enough to serve as a good starting point, yet challenging enough to expose some of the common problems encountered in robotics.

In his book, *Vehicles*, Braitenberg suggests that machines (robotic or biological) can exhibit increasingly complicated behaviors as a result of neural connections between sensory organs and actuators. He goes on to discuss how these behaviors may be implemented using light sensors, and motors. (see illustration) We thought this would be a good theme for the first lab assignment.

Reading:

P. J. McKerrow, Introduction to Robotics, pp 1-49.

Skills:

Programming, digital and analog input/output control.

Challenge Statement:

"This is an exercise in science, or science fiction, if you like that better. Not for amusement: science fiction in the service of science. Or just science, if you agree that fiction is a part of it, always was, and always will be

Figure 2. A sample Web page assignment: http://generalrobotics.org.

Laboratory assignment design goals. To guide the design of the course's laboratory component, we set up these metrics:

- Develop a hands-on artifact construction component that is closely tied with theoretical ideas in the curriculum.
- Provide ample tools, both physical and conceptual. In both cases, do not simply give minimum ingredients for a recipe, but enough to do a task many ways. (In the case of providing ideas, this is largely the job of the instructional component.)
- Give freedom to explore, but stay within the time constraints allotted per subject—including assignment time per week—and the relation to curriculum goals.
- Include a creative component where students can define their own path to a stated goal, thus encouraging design diversity.
- Present the assignments as goal-oriented design challenges.
- Present labs through a medium that will facilitate easy access to useful information.
- Develop a grading scheme that is fair, well-defined, and feasible given large enrollment numbers.

Implementation

We began to apply these new ideas in the second offering of General Robotics during the fall 1999 semester. Other than minor

Table 2. Descriptions of lab assignments.

Assignment	Description	Curriculum Aspect
Lab 1: Vision on Sun workstations	Take an image file and write a program to threshold, histogram, and detect edges.	Robot vision algorithms
Lab 2: Behavior-based robots	Build and program a robot that will drive to a light source in a test environment.	Behavioral AI Learn to use robot building kits
Lab 3: Low-level mobile base control	Build a robot base that can translate and rotate accurately.	Understand robot base mechanics Understand necessary hardware functionality for certain motion planning algorithms
Lab 4: Motion planning	Implement the wave front motion planning algorithm on the mobile base, such that the base will plan a path through a 4 × 8-foot world given a numerical map of obstacle locations.	Motion-planning algorithms
Lab 5: Ackerman steering	Build a robot base that uses an Ackerman steering configuration and program it to escape a “parallel parking” situation where it is trapped perpendicular to, and in between, two rectangular boxes.	Ackerman steering and alternative drive configurations
Lab 6: Mars rover	Build a teleoperated rover to navigate some ridiculously challenging terrain you’ve only seen in low-quality digital photographs. Navigate your robot from an approximately known start to finish via text-based computer terminal control while watching progress through a small video screen with a bad camera angle.	Robot locomotion Space robotics design Design and programming for teleoperation
Lab 7: Forward kinematics	Build a two-degree-of-freedom planar jointed robot arm and program it to plot points (with a pen attached as an end-effector) described by a table of theta1 and theta2 coordinates corresponding to shoulder and elbow angles.	Forward kinematics Robot arm mechanics
Lab 8: Inverse kinematics	Implement a software function that takes an *x-y* coordinate and drives the arm to that point. Then have the arm draw a picture of a shape you have chosen.	Inverse kinematics

schedule changes, the course’s lecture component remained largely unchanged. However, we did eliminate self-defined projects and added a lab component that would follow the established curriculum (see Table 1 for the syllabus). From the design perspective, we had to develop eight separate lab assignments and to research and assemble lab kits for executing these labs. Logistically, we had to secure funding for the equipment, obtain lab space, train a small staff of teaching assistants, and prepare supporting lab materials such as test environments in which the robots could operate.

Creating and assigning the labs. In creating the lab assignments, we began by identifying which curriculum topics were most important and best suited for lab implementation. After identifying topics, we began the imaginative process of designing tasks that would allow for creativity while requiring an understanding of the topic at hand. Here is where we closely referred to our list of design goals. Table 2 describes the lab assignments.

Students accessed lab assignments through the Web (see Figure 2), a medium we chose for its ease in disseminating information. The Web also provides a convenient forum for linking supplemental information and help topics, which were often referenced as external resources. Because all the assignments required the students to program robots on a computer, the Web-based format gave them instant access to this information while working. Also, educators from other universities can easily replicate our implementation using our Web site.

Each lab’s format included

- *An introduction*: the subject, educational goals, and background information.
- *A challenge statement*: a very general statement, such as “Build a robot (creature) that chases light.”
- *Evaluation*: specific criteria; demonstration and testing procedures. This often discussed the parameters of a test environment for the robot (for example, a table with guides drawn on it).
- *Grading*: actual scores corresponding to the degree of success in achieving criteria stated in evaluation tips.
- *Things to consider*: help information, related subjects of interest, and additional readings.

We particularly considered the amount of information and help provided to the students in each lab assignment. Each lab had a pedagogical focus congruent to the curriculum. We aimed to minimize time-consuming aspects of a project that were outside this focus by providing help tips and, in some cases, sample designs for a particular subsystem. For example, we suggested a potentiometer and drill bit size to make an encoder for a robot arm; students need not waste their time researching this.

Determining the amount of information to provide involved a delicate balance. If we provided too much information, student creativity might be constrained. If we gave too little, peripheral aspects might detract from the assignment’s focus and we might exceed the target weekly assignment time for an individual class. Maintaining this balance has been one of the most challenging aspects of developing these assignments; we expect it to be the most evolving aspect as we improve assignments from year to year.

Students formed three-person teams before we handed out lab kits. Because each team would assume responsibility for the equipment in the lab kits, teams had to remain static for the semester. Students chose teams themselves, but we encouraged that teams have one member from each of the primary field constituting the class—mechanical engineering, electrical engineering, and computer science.

After teams formed, the three members signed a responsibility statement agreeing to share the financial responsibility of either returning the lab kit in complete, working condition at semester's end or paying the replacement cost. Students then received a cardboard box containing all their kit's components, which they were to keep for the semester. No one complained about the responsibility agreement. The kits held up well throughout the semester. Out of the 23 kits we lent out, only minor LEGO pieces were lost (but many were returned as extra in other teams' kits), and two Handy Boards had screen failures due to undiagnosed reasons (replacement screens cost $8). As anticipated, many of the sensors and wires were permanently damaged, in some cases making then unusable. After assessing replacement needs for the 2000 class, we estimated that we'd need approximately $15 per kit to replace nonrenewable resources.

Lab kit design. Kits had to be ample enough to let students accomplish assignments in a variety of creative ways. Additionally, they had to be cost-effective, rugged enough to last for years, serviceable, and provide for easy startup while not abstracting important pedagogical aspects. We selected LEGO Technic (including motors) as a hardware component, the Handy Board (Motorola 68HC11-based) microcontroller board for control, and our own selection of assorted sensors and other components. See the "Lab kit components" sidebar for details of the equipment we chose.

Logistics. We assigned no lab hours to the course; students had to complete all lab work was as part of regular homework assignments. We did, however, make laboratory facilities available to the students for, on average, 30 hours per week. This lab was an existing undergraduate laboratory run by the Mechanical Engineering Department and included 20 workstations, each with a Windows PC, oscilloscope, function generator, and power supply. Most of this equipment was only used occasionally, with the exception of the PCs, which had Interactive C installed. In addition to existing lab equipment, we supplied tools for making sensor and motor wires (soldering irons, solder, clamps, wire cutters, hot glue, and heat guns for heat shrink tubing). We also supplied a drill and 15/64-inch bit, necessary for modifying LEGO Technic bricks to mount the 1/4-inch base shaft encoders and potentiometers used in certain labs. The Mechanical Engineering Department required the lab to be staffed by teaching assistants when open.

STUDENT RESPONSE WAS OVERWHELMINGLY POSITIVE. THEY ENJOYED THE LEGO LABS, AND THE DESIGNS THEY PRODUCED VARIED IMMENSELY.

In the 1999 offering, the TA staff included Michael Rosenblatt, who had experience with the lab kit equipment, and a graduate student, who was well educated in the robotics theory taught in the course, but had no experience with the equipment and little experience in building robots. Although the graduate student quickly became experienced enough to answer equipment-related questions, many of the early questions and troubles our students had related to equipment and robot building, which resulted in an overcommitment of Rosenblatt's time. For the 2000 offering, we have arranged for eight undergraduate TAs and have lowered the enrollment to 50 students.

In addition to the time required by TA duties, setup took several hours for most lab assignments for preparing demo environments such as Cartesian grid tables and obstacles. As the extreme case, the planetary robotics assignment required that the TAs sculpt a 4 × 8-foot, 3D terrain out of foam, wood, and plaster and paint and build a remote-controlled video camera pan-tilt mechanism out of LEGO elements. This assignment's preparation took over 30 hours for the two 1999 TAs. The 2000 offering has a committee of six people preparing the planetary robotics lab.

Grading. Evaluating and assigning grades to student projects was one of the most difficult issues we encountered—and it is not fully resolved. Based on the class's size and the precedent set by engineering classes to grade objectively, we intended initially to evaluate students solely on the basis of robot performance. Students asked why, when we encouraged a creative process, we graded solely on end functionality. They suggested that if a group had a unique design, we should take that into account while assessing their accomplishments. How do we encourage creativity without detracting from the importance of producing working results? Getting a robot to work on demo day demonstrated not only an understanding of the material but also a care taken to test and ensure the robot's reliability.

This grading philosophy worked well in that those whose robots demonstrated successfully and performed the best probably did have the best understanding and put in the time and care to test and tune their robot projects. However, it did occasionally fail to assess a student's grasp of a concept when a student did understand, but the robot still failed to work owing to opaque problems such as an intermittent disconnection of a sensor wire under the heat-shrink tubing that was upset on the way to the demo. The current grading scheme is not forgiving to such events. When there was an obvious, faultless error—such as a visible sensor disconnection—we would let students repair the robot and take a later slot in the demo hour.

In the 2000 offering of the planetary robotics lab, students will keep design journals and have several design checkpoints during the three-week assignment. There, they will propose and discuss their designs and be evaluated based on design process including exploration of alternatives, reasoning, testing and evaluation, and final implementation. This is in similar spirit to the Case Western Reserve University Autonomous Robotics class, in which students are graded on process and evaluated through a design notebook.[2]

However, for the other labs, we must hold true that our primary focus is not engineering design, but robotics. An important concept to be learned in robotics is to develop robust, thoroughly tested solutions.

Lab kit components

Designed to be flexible, cost-effective, and rugged, the kits cost about $600 each.

Lego Technic as a prototyping tool

Over the past two decades, Lego Technic has evolved from a children's toy to a fully capable engineering systems prototyping tool. In addition to its well-known structural components, it offers gears, motors, wheels, sensors, and even programmable microcontrollers. It is cheap (relative to other prototyping tools), familiar, allows for very rapid construction, and is readily available.

We chose the Lego Dacta Robolab 9790 kit. Dacta is the Lego educational (selling to schools) division. The Robolab 9790 contains essentially the same parts as the retail Lego Mindstorms robotics kit, which sells for $200. The 9790 costs $220; the main difference is that the kit comes in a hard plastic box, which is convenient for students. The parts in both these kits provide ample (but not abundant) resources for building the robot projects assigned in our class. Both kits come with the Lego RCX microcontroller, which accounts for a significant portion of the cost.

When purchasing the kits, we shared their cost with a middle school interested in extending its existing Lego collection to include the RCX microcontroller. So, when we received the kits, we stripped out the RCX controllers and interface hardware and software, and kept the rest of the kit. For those considering offering a similar course, any sufficiently large Lego Technic kit (by rough means of evaluation, greater than $60), supplemented with two Lego Technic Motor kits (roughly $39 each), should be adequate.

The MIT Handy Board microcontroller

We initially considered using the Lego RCX microcontroller that came with the kit. However, we declined to use it, in spite of its low cost, because it has limited I/O capability (three inputs and three outputs), making it poorly suited for some of our assignments. Instead, we chose the Handy Board, which was designed by MIT Media Lab research scientist Fred Martin and is intended for student robot projects (see Figure A). It has built-in batteries, interfaces directly to motors and sensors, has a liquid-crystal text display, and comes with an easy-to-learn C-programming environment called Interactive C. The Handy Board is an open-source descendant of an early 6.270 Board, which was designed for the MIT 6.270 Autonomous Robot Design Course. Users can program the Handy Board from a computer and then execute the downloaded program untethered.

Interactive C is a subset of the C programming language also developed for the 6.270 course. In the spirit of Logo, the children's programming language, it lets users instantly execute commands or downloaded functions.[1] This feature is particularly useful for interfacing electromechanical systems to software because students can easily test their implementations as quickly as they construct them.

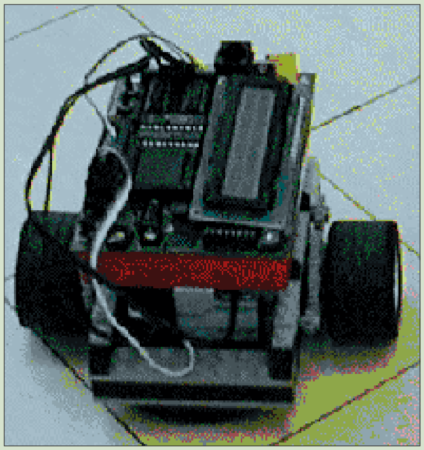


Figure A. The Handy Board sitting atop a Lego mobile base.

The Handy Board costs $300, assembled and tested. Interactive C is available in freeware and commercial versions for any platform, costing $35 per license. We use the commercial Windows version, because the free version (a DOS program) does not implement 2D arrays, a feature useful for our motion-planning lab.

General robotics sensors and miscellaneous electronics

We supplemented the Lego and Handy Board kits with our own selection of off-the-shelf electronic parts from various electronic supply companies, such as infrared sensors (Figure B). In many cases, we included parts with a specific purpose in mind (photoresistors for a light-seeking robot assignment). In some cases, however, we found parts in surplus catalogs and included them because of low cost and creative potential (such as rolling-ball inclinometers). A complete part listing is available at our Web site.

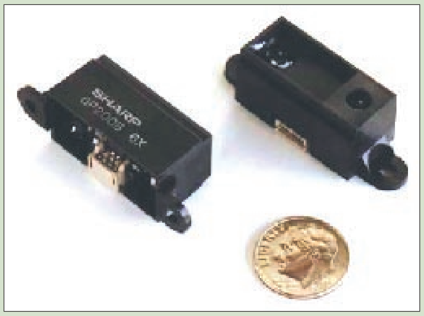


Figure B. Sharp infrared sensors.

Evaluation

The primary means of measuring our success in implementing the directed constructionism approach is to assess how we answered the specific challenges we foresaw and responded to unanticipated challenges. Student response was overwhelmingly positive. They enjoyed the LEGO labs, and the designs they produced varied immensely.

Lecture–lab integration. A primary design goal for the new lab component was to ensure that the labs related to the curriculum introduced in lecture. This is central to the success of the directed constructionism approach. The syllabus and lab descriptions in Tables 1 and 2 show clearly that most assignments corresponded to the lectures. However, some lab assignments represented the theories better than others. In one case, the lecture focused on Ackerman steering theory (the type of steering found on normal automobiles where the front wheels steer and the rear wheels remain fixed). The assignment was to build a robot that could escape a parallel parking scenario. Although the assignment related to the theory, it was not clear that the theory could be applied to the problem to gain an advantage in implementation. One student pointed this out to us in an email:

> We are confused about this lab.... Right now it seems as if this lab is very disjointed, where we haven't learned anything to help us with this and all that seems feasible is trial and error. I tried looking at the book and couldn't find answers there. Will anything relative to this lab be covered in lecture tomorrow?

This email is a good representation of our students' willingness to provide feedback regarding our ideas. In a similar situation, feedback on the first part (forward kinematics) of the two-week kinematics–robot arm lab led us to alter the second part (inverse

Online resources

General Robotics course Web site: www.generalrobotics.org
MIT 6.270 Autonomous Robotics Design Competition: http://web.mit.edu/6.270
Case Western Reserve University, Autonomous Robotics: http://www.eecs.cwru.edu/courses/lego375
The Handy Board Web site: www.handyboard.com

kinematics) portion to provide more interest for the students. While they recognized an arm's usefulness, several students expressed that targeting a *x*–*y* coordinate was very mundane. Based on this feedback, we added a criteria for the second-week demonstration: pick any shape and have the arm draw it. Students had their arms draw a simple circle, stars, and even a fish.

Some labs intentionally diverged from the lecture schedule and became standalone curriculum. These cases represented points in the curriculum when the theories presented in lecture did not have physical analogies that could be implemented in lab and when a particular curriculum was better explored in lab without theoretical supplement. One particular example of this divergence was the Mars rover lab (which served as an in-class contest).

In the Mars rover lab, students designed and built a robotic rover to explore an extraterrestrial surface through teleoperation. They got only "satellite images" of a surface, which was constructed out of Styrofoam to include obstacles such as small stones, large cliffs, and a ravine (see Figure 3). Students also got topographical maps with specified landing and goal zones (see Figure 4). The design challenge was to construct a rover to get from the landing zone to goal zone. They would operate the robot from a computer terminal in a separate room while viewing progress through a video camera.

This assignment's goal was to learn about robot teleoperation, explore different types of locomotion, and participate in a simulated space robotics scenario. This assignment did not relate to the topics being discussed in lecture at the time, but it did let the students work on a more significant project. Many responded that this was the course's most fulfilling project.

This two-week assignment and contest was so well received and proved to be such a great learning experience that next year we will suspend standard lectures during the contest and instead meet with individual groups about their designs. This approach will help foster the class's design experience component.

Parallel to the lecture-related versus unrelated assignments disparity was a variance in creative freedom, with the unrelated labs being the more open-ended. We are continuing to gauge this relationship between relevance to lecture and creative freedom. While assignments such as the Mars rover lab have proven to be an excellent experience for students, other labs, such as the path-planning and motion lab, which directly reflected content taught during the lectures, allow for better control over what students learn. Both types are important. Students appreciated both types of assignments, but recognized that even the more constrained lecture-related lab assignments allowed for more creativity than many of their past lab experiences. One student commented on this in an exit interview:

> The way to differentiate these labs from the labs that students typically do in engineering courses is that there wasn't a list of numbered instructions handed out to students giving a linear path from point A to point B. Students had to figure most of it out for themselves, which teaches them a lot more than simply developing their abilities to read and follow instructions. I'm pretty sure that was thoroughly perfected back in first grade. The labs allowed for creative freedom, while also achieving the goals of the curriculum.

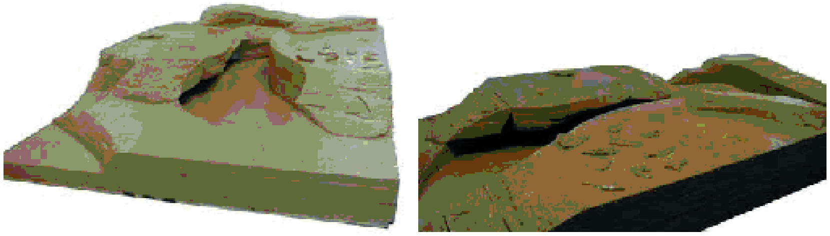


Figure 3. "Satellite" images of Mars.

The lab culture. The lab component also had unanticipated side effects that strengthened our belief in the directed constructionist approach. The long hours spent in the lab and intensity of the assignments brought a unique sense of community to the course. Another result was an emergent collaborative learning environment. Groups intermingled and helped each other construct new and unique designs. Discussions with members of other teams often solved problems and sprouted new ideas.

BY TAKING A DIRECTED CONstructionism approach to robotics education, we have created a learning experience that helps students better understand instructional material. We accomplished this by integrating goal-oriented lab assignments and a traditional lecture-based curriculum. In addition to acquiring an understanding of the material, students participated in a unique multidisciplinary design experience. The combination of applying theory to practical hands-on tasks and working across disciplines resulted in an engaging learning experience for both students and faculty.

Teaching the course was a challenge that kept us constantly on our toes. Although labs paralleled the lecture in topic (except the Mars rover lab), the instructor made a conscious effort in lectures to use lab work as practical analogy for the theory being presented. Likewise, lab assistants frequently mentioned the theories being presented in lecture to answer a practical question during lab hours. This synchronicity strengthened the connection between theory and practice, but required additional planning for course faculty and staff, including weekly meetings (sometimes more often). Also, every group attacked each lab differently, which forced lab assistants to learn a particular group's methods before offering help. Lab assistants thus had to be knowledgeable in all disciplines being applied and have a strong intuitive understanding of the task—an understanding that usually only develops from lots of experience.

Grading remains an unresolved issue. Grading solely on performance serves several purposes. It accommodates the large number of students in the course, provides an objective standard for evaluation, and reinforces the importance of producing working

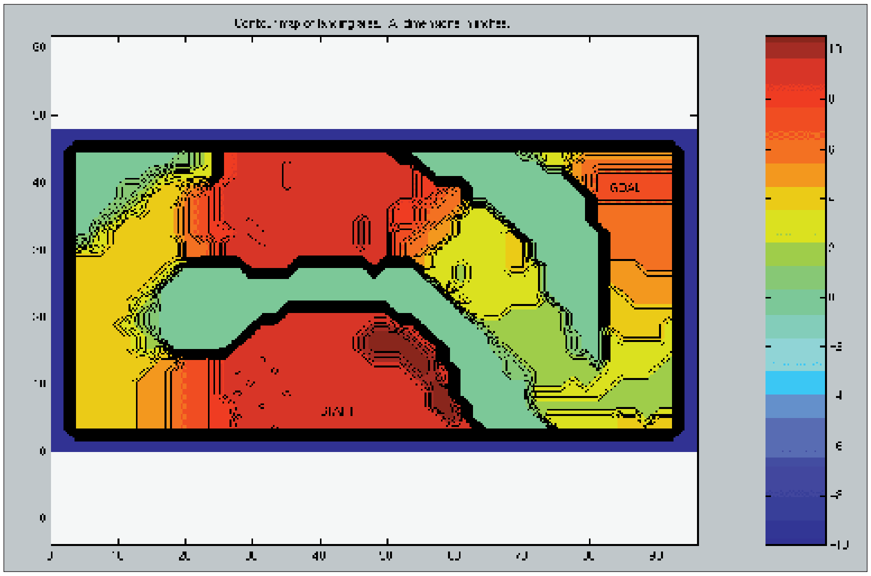


Figure 4. Topographical map of Martian terrain.

results. Conversely, it discourages creativity and often does not reflect the amount of effort put into an assignment where 90% of the robot works but a minimal 10% prevents it from accomplishing the task. In such cases, the current grading criteria does not take into account the 90% functionality, and a poor grade becomes negative reinforcement of what in many cases were 10- to 15-hour efforts. We are considering several hybrid design-review and performance grading schemes.

Looking further ahead, we will seek to improve the course and directed constructionism concept on a number of fronts. The Mars rover lab was unique in that it provided an exciting and imaginative scenario, an aspect that clearly excited the class. Exploring how to extend such scenarios to other lab assignments is a potential topic for future investigation. To facilitate such scenarios, supporting technologies and devices could be developed (such as a remote video camera tripod for the Mars landing). Beyond this course's borders, we will seek to explore how to extend directed constructionism to other academic subjects and to secondary and elementary education. ■

Acknowledgments

We thank our reviewers for their kind critical reviews. We also thank Fred G. Martin for helping us clarify the article's direction. We gratefully acknowledge the financial support of NASA Ames (1999) and Caterpillar (2000) for this course. We thank Douglas Hull for being the consistent workhorse for General Robotics 1999 and 2000 and also Steven Klim and Adnan Akay for supporting this class, especially the logistics. We also thank the students of General Robotics 1998, 1999, and 2000 for their feedback and for being willing alpha and beta testers of our new course design. Finally, we thank Robin Murphy for her continued support and friendship.

References

1. F.G. Martin, *Circuits to Control: Learning Engineering by Designing Lego Robots*, doctoral dissertation, Program in Media Arts and Sciences, Massachusetts Inst. of Technology, Cambridge, Mass., 1994.
2. R.D. Beer, H.J. Chiel, and R.F. Drushel, "Using Autonomous Robots to Teach Science and Engineering," *Comm. ACM*, June 1999.
3. S. Papert and I. Harel, eds. *Constructionism*, Ablex, Westport, Conn., 1991.

Michael Rosenblatt is a senior at Carnegie Mellon University, where he studies intelligent media design. Over the past year, he has helped design and implement a hands-on supplement to Howie Choset's introductory robotics course. He is very interested in new technologies for toys and learning and has spent two summers working in this field with the Epistemology and Learning Group at the Massachusetts Institute of Technology, Media Laboratory. Contact him at mr4@andrew.cmu.edu.

Howie Choset is a professor in the the Mechanical Engineering Department and Robotics Institute at Carnegie Mellon University where he serves as director of the undergraduate robotics minor. His research interests include motion planning and design of hyper-redundant mechanisms, coverage path planning for robotic de-mining and auto-body painting, and sensor based mapping. He is a member of IEEE (IEEE AdCom), ASME, and WanKY. Contact him at choset@cs.cmu.edu.