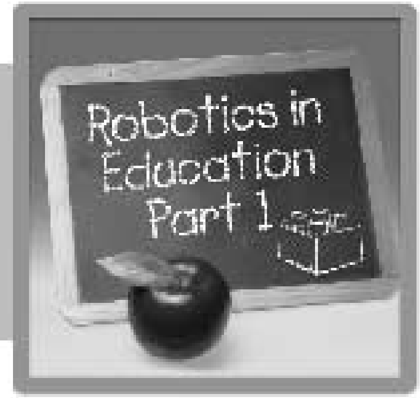
Achieving Educational and Research Goals with Small, Low-Cost Robot Platforms



Mobile Robot Labs

By LLOYD GREENWALD and JOSEPH KOPENA

Recently, there has been much interest in achieving educational [1]-[6] and research [7] objectives through the use of small, low-cost, configurable mobile robot kits. One such article [1] argues that robotics provides students with needed experience dealing with integrated systems building, real-world issues, teamwork, multidiscipline information, and critical thinking. Recent excitement with the initial success of robot-based courses has led to ambitious plans incorporating robotics into computer science (CS) curriculums [2]. Universities and high schools are employing these kits in artificial intelligence (AI), CS, engineering, and physics courses where they provide students with a new perspective on building integrated systems, allowing hands-on education and experimentation at low cost.

While our initial experiences with these platforms were similarly positive, we questioned whether these platforms could be pushed beyond their early uses and transitioned towards achieving substantial educational and research goals. This article reports initial results of this investigation—the construction and implementation of a series of detailed lab assignments using these platforms to tackle basic CS, AI, robotics, and engineering problems. These labs focus on dead reckoning, understanding sensors, real-time scheduling, and sensor sampling as well as feature detection and map building with sonar. We additionally report on our efforts to supplement these lab assignments with research and tool-building projects.

We assembled mobile robot kits similar to those used elsewhere, centering on the Handy Board microcontroller board [8], LEGO construction pieces, and sensors built with parts from various vendors. The microcontrollers are programmed in the Interactive C (IC) programming language. A detailed parts list is available from our Web site at http://itcsl.cs.drexel.edu. A second mobile robot kit option, based on the LEGO Mindstorms RCX microcontroller, is discussed later in this article.

Our primary goal was to see how far we could push the limits of this technology. In particular, we designed labs with a variety of sensors (e.g., servo-mounted sonar, wheel encoders), complex programming and CS concepts (e.g., multitasking, scheduling, resource limitations), and potential for careful experimentation and analysis. Additionally, we investigated the potential of using these platforms to teach more advanced AI and robotics concepts, such as Bayesian representation and reasoning, map-based path planning, resource-bounded reasoning, and real-time control. Finally, we developed research projects to use these platforms to explore infrared (IR) communications networks, teleoperation, and real-time control of robots over public networks.

In this article, we first provide detailed descriptions of the labs we have developed and then discuss the robot platforms, including the progression of hardware issues encountered. Finally, we share what we have learned from this endeavor. We believe our efforts and experiences can be of benefit to a wide range of CS and AI educators and researchers in assessing the potential of achieving educational and research goals with these platforms.

Robot Building Labs

In this section we present a series of detailed lab assignments constructed and implemented for an advanced CS course at Drexel University. The course is designed to appeal to advanced CS students with or without an AI background. We first present an overview of the series of labs and then focus on detailed presentations of the labs we believe are most novel or otherwise interesting. The series of labs are:

- Introduction to Mobile Robotics and IC
- Physical Structure and Drive Trains
- Simple Dead Reckoning

 1070-9932/03$17.00©2003IEEE

- Introduction to Sensors and Basic Behaviors
- Multisensor Behavior and Control
- Understanding Sensors
- Wheel Encoders and Odometry
- Virtual Path Following and Obstacle Navigation
- Understanding and Implementing Sonar
- Subsumption Behavior Arbitration Architecture
- Understanding Active Sonar and Simple Feature Detection
- Local and Global Map Building
- Path-Planning Algorithms.

Taken individually, each lab has specific educational goals that are achieved through readings, lecture, robot building, programming, and a carefully designed set of experiments and analyses. The labs build on each other as a sequence, incrementally converging to a multisensor, multibehavior platform capable of exploring advanced educational goals. A differential-drive mobile robot is built that carries its own microcontroller and batteries and eventually includes paired, forward-facing IR sensors and photosensors, a ground-facing reflectance sensor, multiple bump switches, wheel encoders for odometry, and a servo-mounted sonar module and photosensor. Successive behaviors include time-based dead-reckoning, light seeking/avoiding, ground-reflectance change sensing, collision detection and response, wall following by touch and proximity, dynamic IR calibration, distance sensing, path following with odometry, collision avoidance, free-space and corner feature detection, and path planning to navigate simple maps built using sonar. Background reading includes material used to teach early labs [6], [8]–[11], as well as more advanced material in later labs [12]–[16].

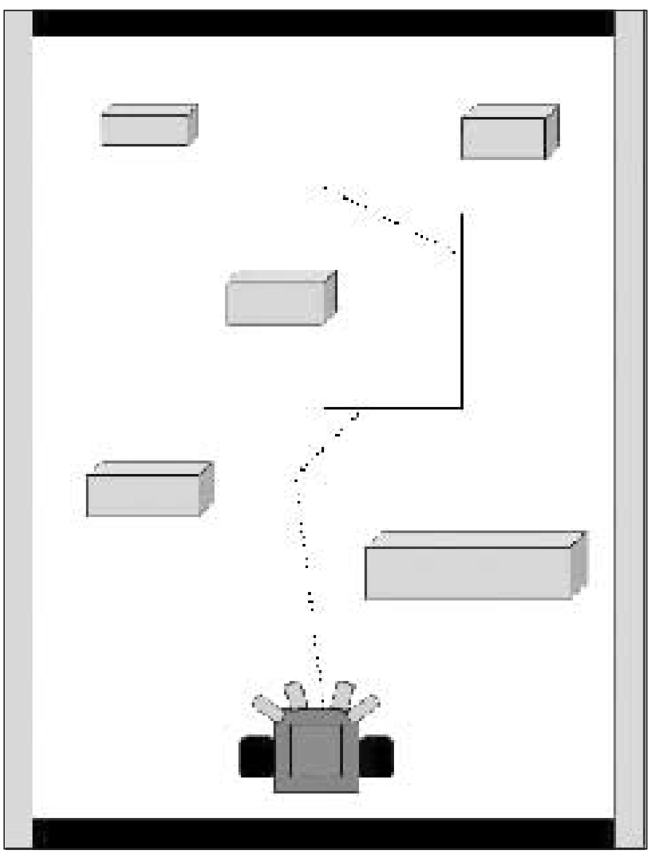


Figure 1. A diagram of the virtual path following task with odometry using wheel encoders. The solid line depicts a pre-specified virtual path. The dotted line indicates potential disturbances to this path due to reactive obstacle avoidance.

There is explicit repetition and reinforcement built into the labs; concepts are taught multiple times with differing hardware and software methods in each incarnation. For example, dead reckoning is taught once by mapping time to distance in a robot that only moves forward and back, a second time with a differential-drive robot, and a third time employing wheel encoders instead of timing. Behavior arbitration is first explored with ad-hoc multitasking methods, then with control theoretic techniques, and finally with subsumption. Navigation tasks are repeated as a richer sensor set evolves.

In the following sections, we describe sets of lab assignments that we found novel or otherwise interesting in more detail, including dead reckoning, understanding sensors, real-time scheduling and sensor sampling, as well as feature detection and map building using sonar.

Dead Reckoning

As mentioned, dead-reckoning experiments and analyses occur in three separate labs. In the first two occurrences, dead reckoning is taught by varying motor speed and duration, manually measuring the change in position and orientation, and noting variance in measurements over multiple runs. This forms the basis for crude odometry that may be used to program movement patterns and for limited robot localization.

For these platforms, the more novel approach to dead reckoning is to actually mount wheel encoders on the drive trains and employ advanced robotics techniques to map encoder counts to position and orientation displacements. Specifically, this lab involves completely rebuilding a robot base in order to mount optical encoders and slotted disks on the drive train. Gear ratio, wheel base, and load distribution must be re-engineered to aid odometry. Once encoders are mounted, students measure wheel base and diameters, program kinematic equations mapping encoder counts to position and orientation changes, and embed these functions into feedback or feed-forward control loops to implement odometry. We were able to take advantage of existing encoder function libraries for the IC programming environment to collect encoder count data from the sensors. Odometry error is measured by programming fixed movement patterns (e.g., a square), then analyzed and corrected using the UMB Mark procedure [11], [13]. These experiments give students experience with the difference between systematic and nonsystematic errors. Students can then create virtual path-following behaviors and combine them with obstacle-avoidance behaviors. Figure 1 depicts the virtual path-following task assigned to the students. Student robots reactively avoid unexpected obstacles while using odometry to maintain a prespecified path.

Odometry capability is crucial to effectively building maps and performing path planning in later labs. However, the implementation of wheel encoders and programming odometry calculations proved very challenging for the students for several reasons. First, recollection of basic geometric and trigonometric principles needed to implement odometry proved surprisingly challenging for advanced CS undergraduates. Second, only in hindsight did most students realize that the 5:1 gear ratio that worked well in early labs did not provide enough torque and was too fast for accurate odometry.

The third, and probably most troubling difficulty with respect to the platform, involved resource utilization. Students reported that the tight feedback loops needed to effectively use encoder function libraries interfered with their ability to concurrently implement other feedback-loop-based behavior, such as obstacle avoidance. Many students could either demonstrate virtual path following or obstacle navigation but not both simultaneously. We have yet to track down the true source of these difficulties.

We note that, if coupled with a graduate AI course, a fourth dead reckoning lab assignment could be created. The manually derived odometry equations could be substituted for a process of learning the encoder to distance mapping with a neural network or similar data intensive method.

Understanding Sensors

While many students may have experience with machines that move in the world, very few have ever dealt with sensors before, especially noisy sensors. Therefore, understanding sensor noise, calibration, and reliability is a primary educational goal. Simple experiments with photosensor readings under different shielding conditions and at different orientation to light sources are explored in an early lab. A more careful analysis of sensors is treated as a lab of its own.

In this lab, students explore the properties of IR sensors and the added benefit of coupling IR sensors with ambient-light photosensors. Students take IR readings at varying distances from an object of varying color and illumination. This experiment involves a test object consisting of a (solid) paper cube with both white (W) and green (G) faces. Each color is tested at distances ranging from 0.5 to 6 in from the robot. Additionally, each test is repeated under varying illumination conditions. Figure 2 shows example IR readings from this experiment. The figure depicts each color tested with no ambient light, overhead fluorescent light, incandescent light projected on the obstacle face, overhead fluorescent and incandescent lights, and all three light sources. The chart shows insensitivity to fluorescent light but variability due to both incandescent light placement and cube color. From this experiment, students

Recently, there has been much interest in achieving educational and research objectives through the use of small, low-cost, configurable mobile robot kits.

conclude that IR reading alone is not sufficient to determine object distance under a variety of conditions. However, when coupled with photosensors for ambient light and assumptions about the reflectance of objects present in the trial arena, students can create virtual distance sensors for obstacles less than six inches away. These sensors dynamically calibrate IR readings with ambient-light readings. Figure 3 depicts a task in which students combine IR and ambient-light readings to approach and orbit a stationary obstacle.

This is the students' first introduction to using sensors to understand features in the world (i.e., distance to an obstacle). This virtual sensor can then be used for obstacle avoidance. Additionally, in this lab students use this virtual distance sensor in combination with state-based control methods to approach an obstacle, stop at a given distance, and orbit the obstacle at that distance. To do so, students must pay careful attention to the relationship between sensor sampling rates

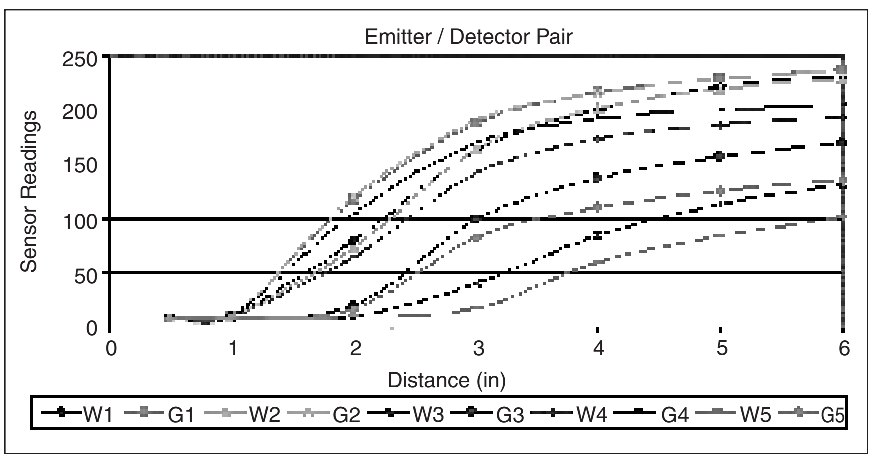


Figure 2. *IR sensor readings under varying object distance, color, and illumination. By combining IR and photosensors students understand sensor uncertainty, sensor fusion, and how to map from sensor readings to real-world distances.*

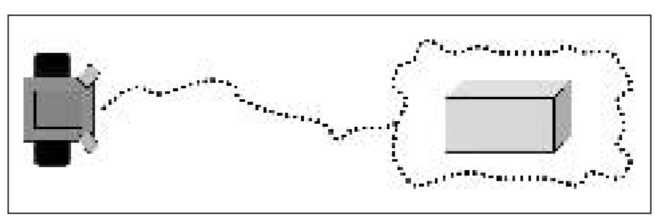


Figure 3. *A diagram of the obstacle orbiting task. Students use closed-loop feedback control to have the robot approach and orbit a stationary obstacle.*

and the frequency of orientation changes during state maintenance. This issue of sensor sampling rate is discussed more carefully later in this article. As the robot orbits the obstacle, illumination changes are handled by dynamic calibration of IR with photosensors.

We note that the IR sensors used in this lab proved to be surprisingly unreliable, frequently broke, and returned invalid readings. Additionally, success in this lab depends on careful design of shielding for the IR and photo sensors.

If coupled with a graduate AI course, the experiments performed in this lab could form the basis for an exploration of Bayesian representation and reasoning. For example, a Bayesian network consisting of IR reading, photo sensor reading, and, possibly, a light-frequency sensor reading as evidence variables, could be built to calculate a query regarding the distribution of distances to the nearest obstacle.

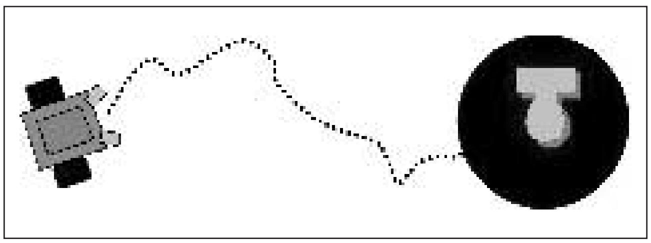


Figure 4. *A diagram of the sensor sampling rate task. Students experiment with the relationship between sensor sampling rate and movement velocity by programming their robots to approach a light at varying velocities and to stop as quickly as possible after a change in ground reflectance is detected by ground-facing reflectance sensors.*

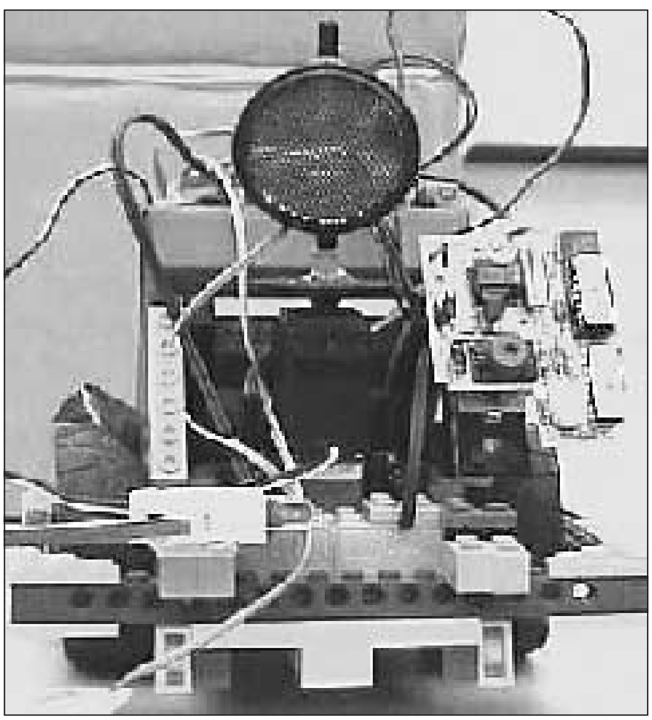


Figure 5. *A sonar module mounted on a programmable servo turret. With this mounting a single sonar can be an effective sensor, especially when combined with robot movement odometry.*

Real-Time Scheduling and Sensor Sampling

It is easy to overlook any discussion of sensor sampling rates. A common assumption is that sensors should be sampled as often as possible or at a fixed rate determined independent of the robot task. However, from a practical view, with the limited resources of these platforms, every task that competes for processor resources should be carefully understood. As an educational goal, we chose sensor sampling rate as an introduction to real-time scheduling of limited resources.

This lab consists of a carefully designed experiment to understand sufficient sensor sampling rates with respect to a desired behavior. Students make use of a robot platform with a ground-facing reflectance sensor and forward-facing photosensors. The task is to combine light-seeking behavior with a behavior that stops the robot as soon as ground reflectance changes abruptly. As depicted in Figure 4, the experimental setup is a light centered over a circular disk of paper that is dark if the main floor is bright or vice-versa. Students experiment with sensor sampling rate by varying the speed of the robot, distance to goal, and the sampling rate of the ground reflectance behavior (e.g., by inserting sleep statements in the feedback loop or modifying the time ticks allocated to the function using IC's multitasking capabilities). The objective is to move as quickly as possible toward the light yet stop as soon as the ground reflectance changes (within a fixed distance). In doing so, the students determine the relationship between speed of movement and sensor-sampling rate.

Several other lab assignments explore real-time task scheduling issues. Multiple behaviors are scheduled using IC's multitasking capabilities very early in the labs. Later, the students are introduced to subsumption [16] as a principled approach to behavior decomposition that does not directly address issues of real-time scheduling.

Sonar, Feature Detection, and Map Building

The most ambitious goal of this work is to make effective use of a sonar module mounted to a servo turret. In a succession of labs, we explore the sonar sensor itself, the integration of active sonar sweeps into one-dimensional (1-D) features, and the integration of sonar sweeps and odometry to build two-dimensional (2-D) maps. Figure 5 depicts an example servo-mounted sonar turret.

The first of these labs teaches students about sonar and how to attach a sonar module to the Handy Board. These directions are based on schematics developed at http://lcs.www.media.mit.edu/groups/el/projects/handy-board/software/sonar.html and also discussed in [6]. Students take sonar readings at various distances (just as in the IR experiments) and angles using the programmable servo turret. This provides an understanding of sonar beam spread as well as reflection. The students also learn to use physical laws—in this case the speed of sound—to map from sensor readings to real-world features. IR and sonar can be directly compared for their usefulness as distance sensors. We note that the sonar module we used proved to be brittle and easily broken by student mishandling.

The next sonar-derived lab introduces simple pattern matching. Students store 1-D sonar sweeps and devise algorithms for detecting the presence of walls, corners, and free space. More advanced feature detection using angle histograms to detect walls and localize a robot in a walled space is discussed [11]. Examples of sonar sweeps collected by students are depicted in Figures 6 and 7. This lab requires that students integrate readings over a half-circle sweep. These feature-detecting behaviors can then be incorporated into navigation tasks. Finally, students combine sonar sweeps with odometry in building and using 2-D maps. Students experiment with tradeoffs in the granularity of the tessellation of space and the amount of memory and processing power needed to employ the resulting data. Memory is especially limited on these platforms. In this lab, students are taught both probability-based occupancy grid methods and vector-field histogram approximations [12]. Graduate AI courses could go into more depth explaining the Bayesian equations of occupancy grids.

We questioned whether these platforms could be pushed beyond their early uses and transitioned towards achieving substantial educational and research goals.

To aid debugging, we developed a tool to quickly visualize arrays as intensity values. Arrays can be cut and pasted into this tool after being uploaded from a tethered Handy Board. More effective methods for grabbing data from the Handy Board for processing and debugging are needed. We have several projects focusing on working on such tools.

Figure 8 depicts the map-building task, and Figure 9 shows an example of a student map displayed with our visualization tool. Students then program the robot to compare a stored map with new sonar sweep data to detect changes in the world.

At this point the platform is ready for further advanced robotics and AI topics, such as path planning with obstacle maps. We have not yet implemented detailed labs on these topics. We intend to experiment with implementing both force-vector and global-path-planning algorithms, such as V*GRAPH [15] on the Handy Board.

Robot Platform Issues

The mobile robot kit we implemented for this study includes the Handy Board [8] microcontroller, nearly 1,200 LEGO construction pieces, four motors, and 14 sensors built with parts from various vendors. The total cost for this kit is less than an entry-level personal computer. A detailed inventory, including price, supplier, and part number for each component of the kits, can be found at our Web site along with links to informational and supplier Web sites. We designed labs that incrementally build a differential-drive mobile robot platform that includes paired forward-facing IR and photosensors, a ground-facing reflectance sensor, multiple bump switches, wheel encoders for odometry, and a servo-mounted sonar module and photosensor. In building this platform, students manage to consume seven out of seven analog sensor ports, six out of nine digital sensor ports, as well as the serial peripheral interface and power expansion ports of the Handy Board.

The two most nonstandard parts in our kit are the wheel encoders and a servo-mounted sonar module. These parts add less than 10% to the total cost of the kits. This cost is well justified in that they greatly expand the educational and research goals possible with this platform. This added capacity does come, however, with the added cost of significant complexity

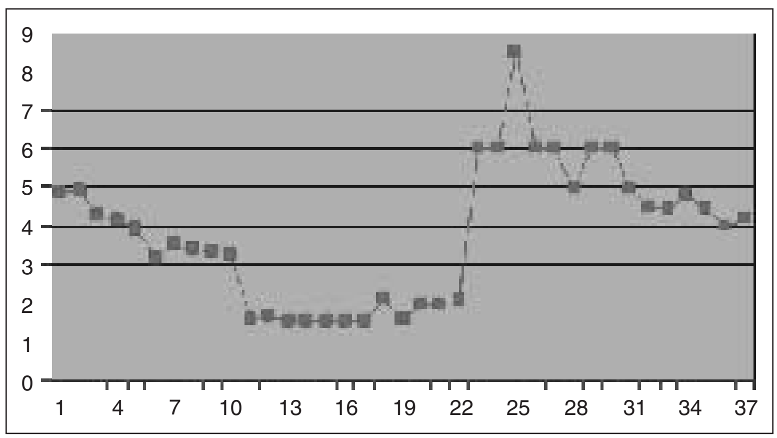


Figure 6. *Example sonar samples resulting from a robot facing a straight wall (centered at sample 15) and rotating the servo turret in a half-circle sweep (in 5° increments).*

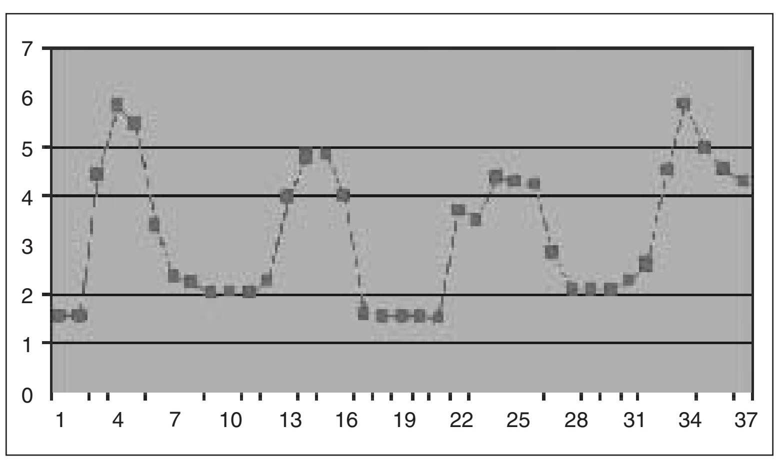


Figure 7. *Example sonar samples resulting from a robot facing a concave corner (centered at sample 15) and rotating the servo turret in a half-circle sweep (in 5° increments).*

Our primary goal was to see how far we could push the limits of this technology.

in lab assignments. These additional complexities allow hands-on lessons about robotics, engineering, and AI that are difficult to replicate with simple sensors alone.

One immediate drawback of fully loading the Handy Board with sensors is battery capacity. Toward the final labs, most students could not complete the designated tasks without being tethered to a power outlet.

The sensors used in the lab assignments are all fairly straightforward to assemble and mount on LEGO pieces, although time consuming. Care must be taken with hot glue, solder, and epoxy as some of the sensors are fragile. Additionally, sensors regularly detach from LEGO pieces and need reassembly. Circuits and assembly instructions for each of the sensors used in these labs can also be found at our Web site.

Building and rebuilding the LEGO structure of the robots proved to be the most time consuming and difficult part of the labs. From other documented experiences and hints [9], [10], we anticipated that the students would not understand the importance of good design just by lecture and would need to experience it themselves. We designed the labs so that the students would be forced to rebuild their bases from scratch three times. Furthermore, we forced the students to complete one lab using someone else's base—and then tear that base down—to further understand the relationship between alternative base designs and robot behavior.

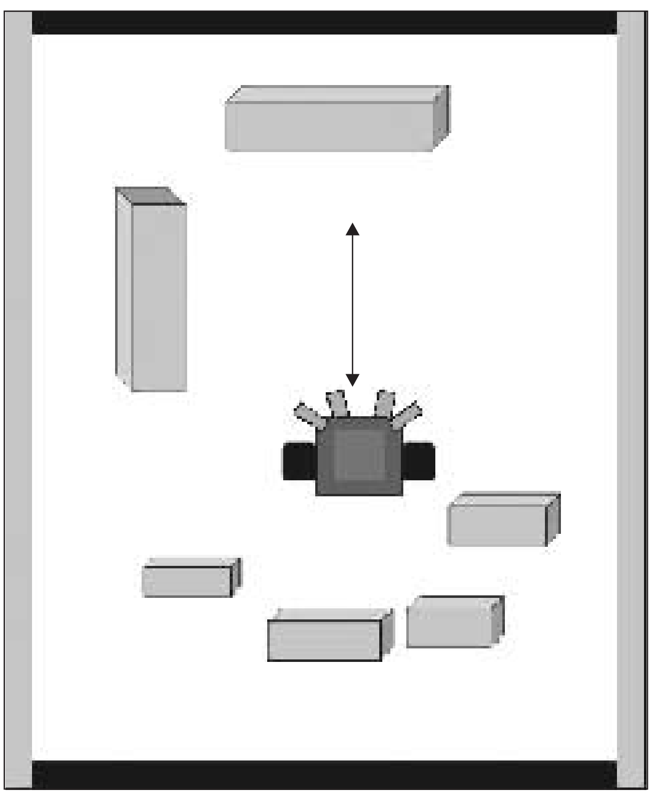


Figure 8. A diagram of the mapping task. Students program their robots to explore and map an obstacle field, combining sonar readings and odometry.

The LEGO Mindstorms RCX Alternative

An alternative yet similar platform to the Handy Board is available at an even lower cost. The LEGO Group markets a consumer robot building kit called the Robotics Invention System. This kit centers around the LEGO Mindstorms RCX, a microcontroller inspired by the Handy Board. The RCX differs from the Handy Board most significantly in that it is limited to three sensor inputs and three motor outputs. The Robotics Invention System kit comes with over 700 LEGO pieces and includes two motors, two touch sensors, and one light sensor. Having been designed for use with children, the RCX is more durable than the Handy Board. There are no exposed electronics, and its case is designed to withstand rough treatment.

We chose the Handy Board for our lab assignments primarily due to the limited number of sensor inputs available on the alternative RCX. Many tricks and circuits have been developed to expand the number of inputs and outputs on the RCX, ranging from putting a light sensor and a touch sensor on the same port to elaborate six-way input multiplexing circuits. We have recently received and are experimenting with the use of an RCX backpack unit, developed by John Barnes of Sensor Applications Inc., that multiplexes the existing RCX ports to provide more sensor inputs and motor outputs. In ad-

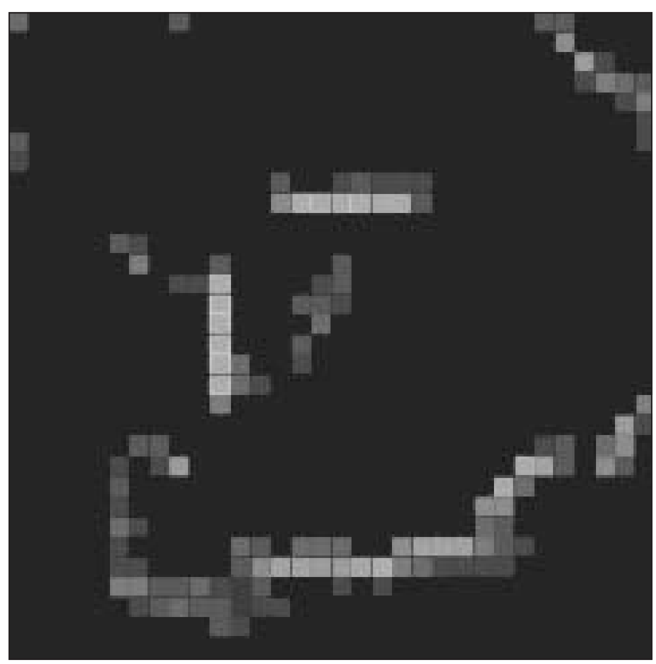


Figure 9. A view of a sonar map generated by one of the students' robots. The obstacle field is the same as that depicted in Figure 8.

dition to having fewer inputs, the RCX sensor interface is also more complicated than the Handy Board sensor inputs. This complicates the circuitry needed to construct even the simplest powered sensors, such as an IR emitter/detector pair. The LEGO Group does sell a variety of well-designed and useful sensors, including light, temperature, rotation, and touch sensors. However, they tend to be comparatively expensive.

On our Web site, we provide a detailed account of the impact of substituting the LEGO Mindstorms RCX for the Handy Board on each lab assignment. In summary, many of the lab assignments should be feasible to implement using the RCX. In some cases, the RCX provides more options (e.g., with respect to programming environments) than the Handy Board. In order to provide runtime user-input, an alternative RCX firmware offering is used since the default does not provide this mechanism. Nearly all of the languages and firmware available for the RCX provide the multitasking features needed in many of our lab assignments.

The light sensor available with the Robotics Invention System can also be used as a ground-reflectance sensor since it incorporates a small LED. However, the use of this sensor as an IR proximity sensor in general is limited by its lack of range.

Servo motors can be interfaced to the RCX through one of the motor ports with some additional circuitry. Angular control over the servo motor can be obtained by using the alternative firmware/language system pbForth. The default firmware only has enough control over the motor ports to flip the servo from one extreme to another.

After adding two light sensors, a ground reflectance sensor, and two touch sensors, the sparse sensor inputs of the RCX begin to become an issue. Tricks to multiplex multiple sensors on the same port take significant assembly effort and hinder the study of sensor fusion, as in our obstacle orbiting lab assignment. Odometry can be accomplished with available LEGO rotation sensors or wheel encoders, however, combining odometry with other sensor-based behaviors does not seem practical without having some sort of general-purpose multiplexer capable of handling multiple light and reflectance sensors.

Sonar attachment and map building are technically possible, though the effort required to construct sonar for the RCX may be prohibitive in the quantities needed for a class. Proposed circuits for sonar include a small microcontroller of its own that has to be programmed.

Discussion

Our efforts to push the limits of small, low-cost, mobile robot platforms rely heavily on the work of the Handy Board community. Our contribution is in finding ways to combine these various hardware and software innovations toward achieving educational and research goals.

We are currently administering this course a second time for graduate and undergraduate students. Significant modifications include drawing heavily on code and detailed knowledge about the Handy Board presented in [6]. We have introduced more exposure to serial communications and off-board processing. This has been made possible by tools developed in student projects over the course of the last year (available on our Web pages). We are also involving the undergraduates more in the graduate student end of term projects. These projects explore some of the issues presented throughout the course in more depth.

These platforms can aid the integration of robotics with more general computer-science educational and research goals.

The set of lab assignments we have developed can be used in various combinations to support a variety of courses. The low cost and small size of these platforms can bring robotics to small schools, undergraduates, and researchers with limited space and funds for large-scale robotics. We have implemented a subset of these labs for high school students as well as federal civilian and military employees. Material from the course has also been used in presentations for high-school teachers interested in robotics as well as middle-school students. Environments are also easier and cheaper to construct and duplicate at this scale. While we have presented this work as an examination of low-cost, small-size mobile robot kits, several of the labs described in this article can be modified for courses that employ more expensive and expansive robot platforms.

Prior to investing in a commercial robot platform, it is possible to support initial research experimentation using these kits to build a differential-drive robot with a variety of sensors, including sonar and wheel encoders and a full set of debugged behaviors. We have developed research projects that use these platforms to explore IR communications networks, teleoperation, and real-time control of robots over public networks.

These platforms can aid the integration of robotics with more general computer-science educational and research goals. For example, as a natural complement to research in controlling robots over networks, we have introduced robot-themed programming exercises into a computer networking course. In one exercise, students program a simple robot server to execute commands issued by a remote client. In another exercise, the students locate servers and clients at different points across the world and across different network media and experiment with the effects of delay and congestion on the robot control application.

Better development tools are needed to realize the potential of these platforms. We are working on extending the array-visualization tool discussed earlier, a sound-based sensor-debugging tool, and remote robot control and simulation tools in

Java and C combining serial communications and remote networked control using both simulated and real networks. We readily disseminate all our research and education materials at our Web site (http://itcsl.cs.drexel.edu).

Acknowledgments

We thank the students in Drexel's Robot Building Lab course for their efforts and insights as well as John Barnes for his multiplexing RCX backpack. This research is sponsored in part by a National Science Foundation (NSF) Instrumentation Award under grant CISE-9986105.

Keywords

Mobile robotics, undergraduate education, robot building labs.

References

[1] R.D. Beer, H.J. Chiel, and R.F. Drushel, "Using autonomous robotics to teach science and engineering," *Commun. ACM*, vol. 42, no. 6, pp. 85-92, June 1999.

[2] D. Kumar, "Curriculum descant, teaching about embedded agents," *SIGART Bull.*, vol. 9, no. 1, p. 7, 1998.

[3] D. Kumar and L. Meeden, "A robot laboratory for teaching artificial intelligence," *SIGCSE Bull.*, vol. 30, no. 1, pp. 341-344, Feb. 25-Mar. 1 1998.

[4] D.P. Miller and C. Stein, "'So that's what Pi is for!' and other educational epiphanies from hands-on robotics," in *Robots for Kids: Exploring New Technologies for Learning Expriences,* A. Druin and J. Hendler, Eds. San Mateo, CA: Morgan Kaufmann, 2000.

[5] F. Turbak and R. Berg, "Robot-based explorations in a liberal arts environment, final report for NSF grant DUE-9650969," Wellesley College, Wellesley, MA, 1999.

[6] F.G. Martin, *Robotic Explorations: A Hands-on Introduction to Engineering*. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[7] T.M. Morse, T.C. Ferree, and S.R. Lockery, "Robust spatial navigation in a robot inspired by chemotaxis in c. elegans," *Adaptive Behavior*, vol. 6, pp. 393-410, 1998.

[8] F.G. Martin, "The Handy Board technical reference," MIT, Cambridge, MA, 1999.

[9] F.G. Martin, "The art of LEGO design," *The Robotics Practitioner: The Journal for Robot Builders*, vol. 1, no. 2, 1995.

[10] J.K. Bennett, "Introduction to engineering design, Elec 201 course notes," Rice University, Houston, TX, 1998.

[11] J. Borenstein, H.R. Everett, and L. Feng, "'Where am I?' Sensors and methods for mobile robot positioning," Univ. Michigan, Ann Arbor, MI, 1996.

[12] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 278-288, June 1991.

[13] J. Borenstein and L. Feng, "Correction of systematic odometry errors in mobile robots," in *Proc. Int. Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, 1995, pp. 569-574.

[14] S. Thrun, A. Buecken, W. Burgard, D. Fox, T. Froehlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt, "Map learning and high-speed navigation in RHINO," Univ. of Bonn, Bonn, Germany, Tech. Rep. IAI-TR-96-3, 1996.

[15] C. Alexopoulos and P.M. Griffin, "Path planning for a mobile robot," *IEEE Trans. Syst. Man Cybern.*, vol. 22, pp. 318-322, Mar./Apr. 1992.

[16] R.A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Automat.*, vol. 2, pp. 14-23, Feb. 1986.

Lloyd Greenwald is an assistant professor of CS and director of the Intelligent Time-Critical Systems Lab at Drexel University. He received a B.S.E. in CS and engineering from the University of Pennsylvania and a Sc.M. and Ph.D. in CS from Brown University. His research interests include real-time planning and scheduling under uncertainty, mobile robotics, resource-bounded reasoning, anytime algorithms, sequential decision making, reinforcement learning, ad-hoc networks, sensor networks, and medical informatics.

Joseph Kopena is an undergraduate student in the Computer Science Department at Drexel University. He has assisted with several robotics and artificial intelligence courses. His research interests include electromechanical assembly representation and reasoning and process specification languages.

Address for Correspondence: Lloyd Greenwald, Department of Computer Science, Drexel University, Philadelphia, PA 19104-2875 USA. E-mail: lgreenwald@mcs.drexel.edu. URL: http://www.cs. drexel.edu/~lgreenwa.