PRINCIPLES AND EXPERIENCES IN USING LEGOS TO TEACH BEHAVIORAL ROBOTICS

Aaron Gage[1] and Robin R. Murphy[2]

Abstract - *This paper describes the application of Lego Mindstorms and Vision Command kits as a cost- and time-effective means of reinforcing behavioral robotics principles to students of different disciplines with limited programming skills. As part of a course in robotics, senior undergraduate and first year graduate students in computer science, engineering, and psychology have worked in small groups building and programming robots to perform a variety of tasks, ultimately developing robots for a mock search and rescue operation. This paper discusses the pedagogical principles, the exercises, student reactions, shortcomings, and lessons learned. The laboratory exercises were used to teach students in two locations (Tampa, Florida and Reykjavik, Iceland) with positive student reviews. The laboratory manual is available to teachers by request, along with the instructor's guide to* Introduction to AI Robotics*. Based on our experiences, we recommend their use.*

Index Terms – artificial intelligence, computer vision, Legos, mobile robots

INTRODUCTION

Behavioral robotics is a challenging area that is of increasing interest to students in computer science, engineering, psychology, and other disciplines. It is necessary to complement classroom lectures with hands-on laboratory exercises to reinforce the material, but providing robot hardware to a large group of students may be prohibitive due to cost and the time needed to learn programing techniques. This paper describes the application of Lego Mindstorms and Vision Command kits as a cost- and time-effective means of reinforcing behavioral robotics principles to students of different disciplines with limited programming skills.

Senior undergraduate and first year graduate students at the University of South Florida and Reykjavik University have taken a laboratory course accompanying a robotics class in which they built Lego robots to perform a variety of tasks to illustrate concepts learned in class. The laboratory experiments were designed to complement chapters in the textbook *Introduction to AI Robotics* [1], and ranged from simple wall-following or visual-servoing robots to a rescue robot in a mock disaster site.

The laboratory exercises used the Lego Robotics Invention Set (RIS) 2.0, which will be referred to below as the Lego Mindstorms kit, and the Lego Vision Command set. The Lego Mindstorms 2.0 kit, which costs approximately $200, provides a programmable microcontroller called the *RCX brick* and enough parts to build a variety of mechanical designs. The RCX provides three motor outputs, three sensor inputs, and an IR transceiver for communication with a computer or other RCX bricks. The RCX can be programmed using a simple language developed by Lego that provides access to its features in a graphical manner, but is limited to relatively simple programs.

The Lego Vision Command kit, which costs an additional $50, provides a small USB camera (similar to a Logitech Quickcam) and software. The Vision Command software can perform basic image operations, including color and motion detection, and can be used to write simple programs to control the RCX brick from a Mindstorms kit. Unfortunately, the Vision Command kit suffers from two distinct weaknesses: first, it was designed to work with outdated versions of the Windows operating system and is difficult to install on current versions; and second, the programming language provided with Vision Command implements only a subset of the language provided with the Mindstorms kit. The Vision Command programs run entirely on a host computer (issuing commands to the RCX through an infrared interface), and are not intended to run in parallel with Mindstorms programs on the RCX. For this reason, implementing complex behaviors on the RCX that incorporate vision is difficult and requires a work-around.

Lego kits have been used to teach robotics at graduate and undergraduate levels elsewhere, with good results [2-3]. These courses may use a competition to encourage student creativity [4-5]. Due to the limitations of the RCX and the Lego programming language, other approaches use the MIT Handy Board instead of the RCX [6] or alternate programming environments for the RCX such as Labview, Java, or Not Quite C [7-8]. Lego kits have also been used to teach robotics in a K-12 setting [9-11].

PEDAGOGICAL PRINCIPLES

The design of the laboratory exercises considered three pedagogical objectives and other secondary objectives. These are described below.

First, the laboratory sessions allowed students to explore issues in behavioral robotics and apply what was presented

[1] Aaron Gage, Dept. of Computer Science and Engineering, University of South Florida, Tampa, Florida. Email: agage@csee.usf.edu
[2] Robin R. Murphy, Dept. of Computer Science and Engineering, University of South Florida, Tampa, Florida. Email: murphy@csee.usf.edu

in the lectures and reading. For example, sensors on a mobile robot are subject to noise and may produce faulty readings. A demonstration of the impact of sensor error is more compelling than anecdotal accounts. The laboratory exercises also reinforced the material by requiring students to revisit the issues presented in class.

Next, the lab sessions encouraged groups of students to solve difficult problems in creative ways, often going beyond what the Lego kits were intended to do. Students generally built new robots from scratch for each assignment, producing a new mechanical design to complement the software they would write. Further, the Lego kits are intended for a younger audience, and their capabilities are limited. Each RCX brick can control up to three motors and can read from three sensors (plus send and receive infrared messages), but much of this functionality is lost when attempting to use the Vision Command camera. The students were encouraged to develop an understanding of how the RCX and Vision Command interact, and in two exercises, to use Vision Command to trigger complex behaviors programmed into the RCX.

Third, they allowed the students to have fun in the class, gaining hands-on experience with building and programming robots in an informal environment. An underlying objective for the lab was that the material should be engaging and enjoyable, so that students would not only reach a deeper understanding of the material by applying it, but would also extend the designs of their robots to do more than what was asked.

Finally, the exercises were conducted with kits that were relatively inexpensive at $250 (compared to a research robot platform, such as the Khepera, that starts around $4,000 including a camera), and without the need for dedicated facilities. Since the kits were affordable to many of the students, they were able to work on their own time and at their own pace. Kits were made available for students that chose not to purchase them, and a general-use computer lab was reserved for four hours once a week to allow students to use the kits. Since the Lego kits were recovered at the end of each lab session, there was no need for a secure facility in which the students could work. Each of the exercises fit within a single lab period and students could keep partial assemblies in their kits between sessions. Cost and availability are not a pedagogical goal, but these allow the labs to be put into practice with a small investment in terms of both time and cost.

LEGO EXERCISES

The Lego laboratory exercises were developed while teaching classes in the United States and Iceland, and a lab manual was written based on the results and observations from these classes. The manual contains a total of six laboratory exercises and one short project for the students to complete. For each exercise, the students were assigned a pre-lab worksheet that tested knowledge from the relevant chapter of the text to encourage students to read the material in advance. Thus, the significance of the lab would be better understood when the students began their work. The textbook chapters associated with each lab session are shown in Table I. Another worksheet was assigned to the students at the end of each lab to test the concepts that the lab was intended to convey. Students worked in groups of two or three, but were required to individually complete all of their pre- and post-lab worksheets (even though the results for their entire group were usually identical) to ensure that each student spent some time thinking about the material. Each of the exercises was intended for a single four-hour lab session, and the project was intended for two or three sessions. In total, the exercises and worksheets were worth 40% of each student's grade in the course. In many of the exercises there were opportunities for extra credit, usually exploring further issues and requiring that the groups build a more complex variant of the required robot. The lab manual containing the exercises, worksheets, and answer keys is available from the authors by request. A description of the exercises follows.

Introduction

The first exercise, designed to introduce students to the Lego Mindstorms and Vision Command kits, was split into two parts. The first part required the students to build a simple skid-steering robot base using the RCX and to devise a simple control strategy for it using two buttons (*touch sensors)* to drive and steer the robot. Once built, the students would attempt to maneuver *(teleoperate)* the robot up a ramp to the top of a small docking platform that was only slightly wider than the robot itself. This was a relatively simple task as the students could see the robot and the ramp directly. The hardest part of this task was making the robot's center of balance low enough that it would not tip while climbing the ramp. For the second part of the first exercise, the students attached the Vision Command camera to the robot built in the first part and attempted the same task using only the robot's camera view and the Vision Command control interface (which runs the motors at full speed for short intervals, and tends to oversteer). This part of the exercise was intended to demonstrate *telepresense*, and in particular, how a single camera view does not offer much information (including peripheral vision or an external frame of reference) to aid the navigation task. As a result, the second part of the assignment was considerably more difficult for the students to perform.

The first part of this exercise is intended to take 2-2.5 hours for the students to complete and requires approximately 30 minutes of preparation time for the teacher or teaching assistant (to build the ramp that the robots must attempt to climb). Students are graded based on completion of the task and their answers to the questions in the worksheets, and can earn additional points by developing a more sophisticated control scheme for their robots. For the second part, only 1.5-2 hours of student time is needed,

TABLE I
MAPPING OF LABORATORY EXPERIMENTS TO CHAPTERS IN THE TEXTBOOK

Lab	Chapter	Topic
1a	1	Introduction to Lego kits
1b	1	Introduction to teleoperation
2	3	Affordances, biologically inspired behaviors
3	4	Schema theory, affordances
4	5	Sequencing of behaviors
5	6	Computer vision
6	8	Multi-agent systems
Project	1-6+	Search and Rescue

primarily because the robots from the first part can be reused. Very little additional time is required by the teaching assistant.

Affordances

The second exercise was developed to reinforce the idea of perceptual affordances (perceivable potentialities in the environment for an action), biologically inspired behaviors, and the need for sensors for navigation as opposed to *dead-reckoning*. For this exercise, groups of students built robots that navigated an area that was white with black dots, detecting the dots ("food") using a light sensor pointed at the ground. The robots explored the area using two strategies: a raster-scan (straight lines across the region with turns at each end) and completely random motion. The robot stopped each time it found "food" (a black dot), played a sound, then continued on while students monitored how many dots the robot found over time. Through this exercise, the students made three discoveries: first, performing a raster-scan was susceptible to error as the robot's wheels slipped, so going in a straight line was essentially impossible. Next, making the robot remember where it stopped (at a dot) and continuing on with the remainder of the scan was challenging. Finally, students found that different search strategies yielded different results depending on the environment (including the distribution of dots and the size of the search area). A robot built to complete the second exercise is shown in Figure 1.

This exercise required two to three hours for the students to complete. The most time-consuming aspect of the assignment is programming the robot, as it takes significant trial and error to make the robot perform turns at specific angles (90° and 180°), and to resume its search after stopping without losing track of its position. Approximately fifteen minutes is required for the teaching assistant to construct the area that the robots will search (by taping down pages printed with patterns of dots). Grading is based on student responses on the worksheets.

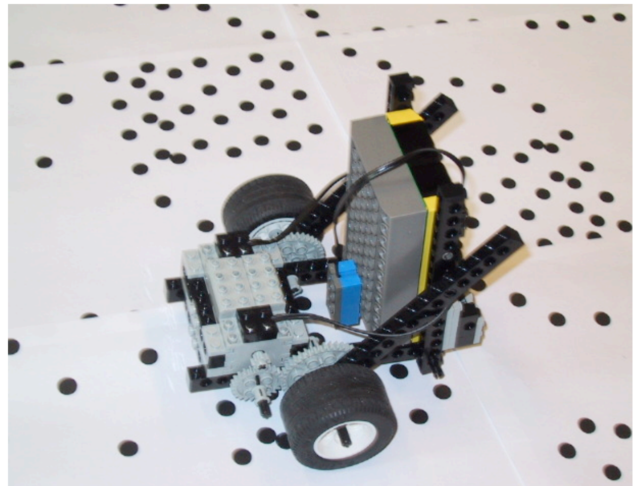

FIGURE 1
A ROBOT BUILT TO COMPLETE THE SECOND EXERCISE DEMONSTRATING SIMPLE BIOLOGICALLY INSPIRED BEHAVIORS

Schema Theory

The third exercise explored schema theory. It demonstrated that a biologically inspired behavior, composed of perceptual and motor schemas, allowed different perceptual schemas to be paired with a single motor schema to perform the same kind of task with different sensors. The students built robots that followed a wall using an antenna (using a touch sensor to detect the wall) and a dark line on the ground (using a light sensor) with the same motor schema. This exercise also exposed a crucial fact when dealing with simple robots and animals: complete knowledge of the environment was not necessary for intelligent behavior, but rather, the robot needed only a simple stimulus to perform its task. In this case, the robot would follow anything that it encountered, whether it was a wall, a shoe, or a trash can; there was no reasoning about what it was that the robot followed. A robot from this exercise is shown in Figure 2.

This exercise requires approximately two hours for students to complete, and requires no special preparation by the teacher or teacher's assistant. Grading is based on responses on the worksheets.

Manipulation

The fourth exercise challenged the students to build a robotic gripper that would grasp soda cans. The purpose of this exercise was to reinforce the notion of affordances, to introduce sensor fusion, and to practice the design of a stateful reactive implementation. The gripper sat in an idle state until it detected a soda can (using the light sensor), at which point it attempted to grasp the can. If the attempt failed (determined by other sensors on the gripper), the robot repeated its attempt three more times. If at any point the gripper succeeded in grabbing the can, it held the can for several seconds, set it back down, and returned to its initial state. In a more challenging variant of this exercise, students

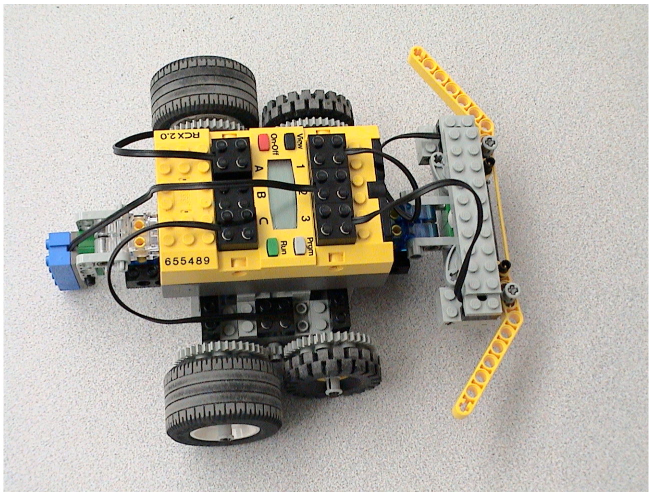

FIGURE 2
A ROBOT THAT CAN FOLLOW WALLS USING ITS ANTENNAE OR A DARK LINE ON THE GROUND USING A LIGHT SENSOR

used the Vision Command camera to detect the can based on its color and trigger the gripper's behavior. This approach was more difficult because the Vision Command and RCX programs are not compatible, and the Vision Command alone cannot be programmed to do the relatively complex behavior required for this exercise. It was necessary for students to learn what messages Vision Command sent to the RCX brick to trigger certain actions, and to have a program on the RCX interpret those commands accordingly. A robot gripper for this task is shown in Figure 3.

This exercise requires approximately two hours to complete, but the variant (using Vision Command) can take three or more hours. No special preparation is required by the teacher or teacher's assistant, though the students should be provided a number of empty soda cans for their robots to grasp. Grading is based on student worksheets, with additional points awarded for completing the Vision Command variant.

Computer Vision

The fifth laboratory exercise provided an introduction to issues in computer vision. The students had experience with the Vision Command system by this point, but had not been exposed to a number of its strengths and weaknesses. The students were asked to build a motorized pan-tilt unit for the camera and to program it to track targets using vision. The targets included brightly colored paper (red, blue, and green), the students' faces, and objects that were white or black. The students found that solid colors could be tracked easily, but that other colors (especially black) could not be tracked at all using Vision Command's color segmentation, and that others (especially white) were highly susceptible to false-positives.

This exercise should take approximately two hours to complete and requires no additional setup time. Grading is based on the worksheets and extra credit is available to

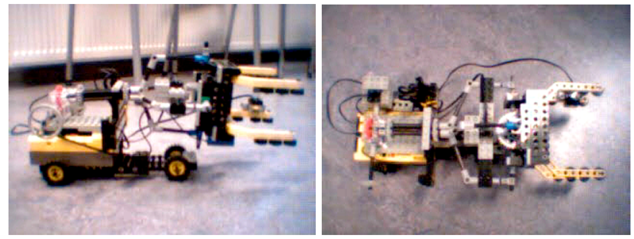

FIGURE 3
A ROBOT GRIPPER THAT PERFORMS A SEQUENCE OF ACTIONS TO GRAB A SODA CAN AND TEST FOR SUCCESS

students that are willing to make changes to the robot and answer optional questions.

Multi-Robots

In the final laboratory exercise, students applied concepts in robot teams. A robot built by the students along with one supplied by the teacher were programmed to cooperatively forage for a target (a large dark spot on the floor, detected using a light sensor). Each robot was to explore independently, and either detected the target and began guiding the other robot to its location (using its infrared transmitter as a beacon), or received a message from the other robot and followed it to its source. This exercise was challenging because the infrared signal emitted by a robot could reflect off of the walls, and the robot it was guiding could easily become confused. Students compensated for this difficulty by developing behaviors that allowed a robot to reacquire the infrared signal if it were lost, and mechanical designs that blocked indirect infrared signals.

This assignment requires three or more hours to complete as the programming can be challenging and the students will likely need to build their robots from scratch. The teacher or teaching assistant will need to spend approximately one hour building a spare robot for the students to use. That is, each group will borrow this extra robot, load their code onto it, and test it with their own robot. If desired, two groups of students can take turns with each other's robot instead. Grading is based on the worksheets, with additional points available for making the robot behaviors more sophisticated.

Project

After the six laboratory exercises, the students were asked to complete a project which was modeled after a simulated search and rescue event (for another class taught with Legos that used this domain, see [8]). For this project, the students designed robots that satisfied a number of criteria: they had to be small, rugged, able to navigate uneven terrain, and equipped with sensors to aid in their exploration of the test area shown in Figure 4. As the robots explored, they visually recognized disaster victims using color and gained points for successfully bringing a “medical probe” to the victim. The course was fragile, and teams lost points for causing further collapse. During the exploration phase of

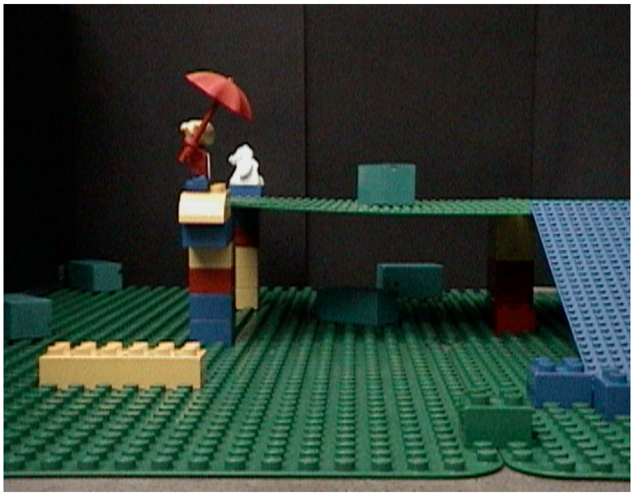

FIGURE 4
A SIMULATED SEARCH AND RESCUE ENVIRONMENT FOR ROBOTS TO EXPLORE FOR THE PROJECT

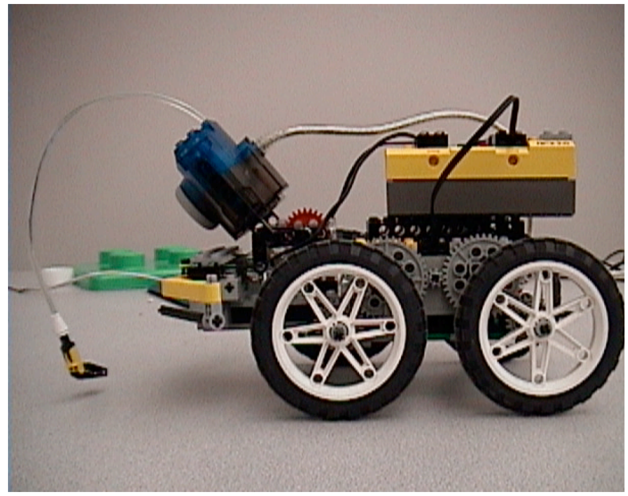

FIGURE 5
A SEARCH AND RESCUE ROBOT BUILT BY STUDENTS FOR THE PROJECT

each trial, the students allowed their robots to run an autonomous wall-following algorithm until a victim was detected using Vision Command. At this point, students took control and teleoperated to the detected victim. At the end of each test run, teams attempted to bring their robot back to its starting point, which may not have been possible due to damage or instability in the robot. For additional points, the students could build a fully-autonomous robot that would not only detect the victim but also navigate to it unaided. The project tested the concepts studied in earlier exercises and provided the students an opportunity to present their own designs to solve an open problem. A robot built to complete the project is shown in Figure 5.

The project requires 12-15 hours for the students to complete and can be split across a number of lab sessions. It will be necessary for the teacher or teacher's assistant to spend approximately thirty minutes per session to build the "disaster area" that the robots will explore. Grading is based primarily on a group report, though the performance of the robot in the search and rescue task may also contribute.

DISCUSSION

The laboratory exercises detailed above have been used to teach two introductory classes in behavioral robotics (in Tampa, Florida, and Reykjavik, Iceland). The students taking the classes had varied backgrounds, including computer science, engineering, and psychology. Many students had little or no previous programming experience. The labs developed using the Lego kits were accessible to all of the students, and they were able to complete the assignments with a minimum of supervision.

Shortcomings

There are a few shortcomings of the Lego kits for this application that should be noted. First and foremost, it is very difficult to install the Vision Command software on any Microsoft operating system after Windows 98 and the program is often unstable. Vision Command also does not integrate well with the Mindstorms 2.0 software, making it difficult to write programs using the full range of features from both. When a Vision Command program runs, it installs a small program on the RCX that interprets commands from Vision Command; if Vision Command detects that the RCX program it installed is no longer running, it will stop. This effectively prevents the use of programs written for the RCX to be used in conjunction with Vision Command. However, it is possible to use these together, but doing so requires either modifying a code template that Vision Command installs on the RCX (which is beyond the skill of most users) or quickly changing programs on the RCX so that the Vision Command does not detect the change, which is failure-prone.

Next, the Lego kits contain a large number of small pieces that are easily lost and difficult to keep organized. After their first use, the Lego kits made available to the students were missing pieces, and one kit had to be set aside for spare parts. Further, after each course for which the kits are used, they must be sorted and reorganized (as parts have a tendency to migrate between kits when students work together), and this can be a tedious process.

Finally, the Lego Mindstorms programming language is reasonably powerful, but for anything other than simple tasks it becomes unwieldy. The language allows for different threads of control within the program, such that each thread can be triggered by a particular sensor or state change event. However, if too many of these threads are used, the RCX may not respond reliably to them. Many of the lab assignments pushed the limits of what the Lego programming language can do. Despite these shortcomings, the exercises met the pedagogical objectives outlined above. Students were able to further explore issues introduced in the lectures, building robots of their own design to perform a variety of tasks, and finally applying what they had learned to an open-ended project. Student reactions and reviews were positive, and student performance on the laboratory questions indicates that the students understood the purpose of the exercises and were able to apply what they had learned.

There were additional lessons learned from teaching the robotics course twice with the Lego kits. It appears that the ideal group consists of three students, which ensures that there will be some diversity in their background and an opportunity to divide work among building and programming tasks. This group size also keeps the number of Lego kits necessary for the class to a reasonable level; our robotics classes average around thirty students so only ten kits are needed. It was also found that students working alone could not always finish the exercises in the four hours allotted.

CONCLUSIONS

Lego Mindstorms and Vision Command offer an inexpensive option for supplementing a course in behavioral mobile robotics using the book *Introduction to AI Robotics*. These kits have been used for six labs and a project as part of two courses, and student reactions were positive. The lab manual (including assignments and answer keys) is available upon request.

Overall, we recommend the use of the Lego kits for teaching. There are a number of limitations: the Lego programming language is too simple to perform complex tasks elegantly, the Vision Command software does not install easily on current operating systems, Vision Command and the Mindstorms kits do not interoperate well, and keeping the kits complete and sorted is difficult. Despite these issues, the Lego kits are accessible to most students and can be used almost immediately, and do not require prior programming experience. Using these kits, students have been able to complete non-trivial tasks that directly reinforce the material taught in class.

ACKNOWLEDGMENTS

The authors would like to thank the following students for providing pictures of their robots included in this paper: Kári Halldórsson, Bjarni Gu_mundur Jónsson, Arnar Freyr Björnsson, Úlfur Kristjánsson, Freyr Gu_mundsson, and Haukur _ór Lú_viksson. Thanks also to Jen Carlson and Seema Patel for suggestions and proofreading.

REFERENCES

[1] Murphy, R. R., *Introduction to AI Robotics*. Cambridge, Massachusetts: The MIT Press, 2000.

[2] Williams, A. B., "The qualitative impact of using lego mindstorms robots to teach computer engineering," *IEEE Transactions on Education*, vol. 46, no. 1, p. 206, February 2003.

[3] Schumacher, J., Welch, D., and Raymond, D., "Teaching introductory programming, problem solving and information technology with robots at west point," in *31st ASEE/IEEE Frontiers in Education Conference*, 2001, pp. F1B-2-7.

[4] Lund, H. H. and Pagliarini, L., "Robocup jr. with lego mindstorms," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 813-819.

[5] Murphy, R. R., ""Competing" for a robotics education," *IEEE Robotics and Automation Magazine*, pp 44-55, June 2001

[6] Rosenblatt, M., and Choset, H., "Designing and implementing hands-on robotics labs," *IEEE Intelligent Systems*, vol. 15, no. 6, pp. 32-39, Nov/Dec 2000.

[7] Wang, E., "Teaching freshmen design, creativity and programming with legos and labview," in *31st ASEE/IEEE Frontiers in Education Conference*, 2001, pp. F3G-11-15.

[8] Garcia, M. A., and Patterson-McNeill, H., "Learn how to develop software using the toy lego mindstorms," in *32nd ASEE/IEEE Frontiers in Education Conference*, 2002, pp. S4D-7-10.

[9] Oppliger, D., "Using the first lego league to enhance engineering education and to increase the pool of future engineering students," in *32nd ASEE/IEEE Frontiers in Education Conference*, 2002, pp. S4D-11-15.

[10] Wang, E., and Wang, R., "Using legos and robolab (labview) with elementary school children," in *31st ASEE/IEEE Frontiers in Education Conference*, 2001, pp. T2E-11.

[11] Caci, B., and D'Amico, A., "Children's cognitive abilities in construction and programming robots," in *Proceedings of the 2002 IEEE Int. Workshop on Robot and Human Interactive Communication*, 2002, pp. 189-191.