

A Guided Internship For High School Students Using iRobot Create^{*}

Jason T. Isaacs^{*} Daniel J. Klein^{*} João P. Hespanha^{*}

^{*} *Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: {jtisaacs,djklein,hespanha}@ece.ucsb.edu).*

Abstract: We present an internship program focused on project-based learning designed to introduce students to robotics engineering utilizing the iRobot Create platform. Our aim is to provide instructors and mentors with a set of projects with an appropriate level of complexity so as to stimulate the interest of the students while providing them a sense of accomplishment. We provide a detailed description of the student projects, an overview of the iRobot Create hardware, and insights gained from our experience and student feedback.

Keywords: Education; Educational aids; Robotics.

1. INTRODUCTION

With today's interest in robotics as a tool for teaching science, technology, engineering, and math (STEM), many instructors find themselves hosting high school and undergraduate research interns interested in robotics. These students typically have some level of programming experience but do not yet have the experience to participate in university level robotics research projects.

A key to a successful internship experience for the mentor, as well as the intern, is to properly match the complexity of the projects to the abilities of the intern. For mentors, this may be the most time consuming and difficult aspect of mentoring. Ideally, the intern will be challenged beyond their current abilities, thus providing motivation for growth, but care must be taken not to leave the intern feeling hopelessly lost. A second key element in mentoring a robotics intern is the selection of hardware. In most hardware intensive labs, it may not be practical to allocate existing expensive hardware platforms to high school interns. These platforms may already be in use by current research, may have a learning curve too steep for a short internship, or may be costly to repair if damaged.

It is these choices of project complexity and hardware platform that are addressed with this paper. We hope to share insights gained from our experience hosting a group of four robotics interns as part of a robotics introduction internship program. This internship program is intended for high school seniors or early stage undergraduate students with an interest in computer science, controls, and robotics. The expected prerequisite experience includes introductory courses in Java programming and calculus. The students should be comfortable with Java but are not expected to be fluent.

^{*} This work was supported by the iRobot Corporation, by the Institute for Collaborative Biotechnologies through contract No. W911NF-09-D-0001 from the U.S. Army Research Office, and by the National Science Foundation Grants No. ECCS-0725485 and No. 0720842.



Fig. 1. Student interns from Dos Pueblos Engineering Academy.

The first group of interns in this program were from the Dos Pueblos High School Engineering Academy <http://www.dpengineering.org> in Goleta, CA, see Fig. 1. The internship took place during the summer between the students' junior and senior years. The students participated in the internship to gain hands-on programming experience to prepare them to be programmers on the 2011 Dos Pueblos High School FIRST robotics team.

There are many examples of using robotics to encourage students to become interested in STEM topics. In Beer et al. [1999], the authors describe the benefits of hands-on learning in their Autonomous Robotics course at Case Western University in 1999. A robotics project course was described in Maxwell and Meeden [2000] as a bridge between undergraduate engineering curriculum and graduate and industrial robotics research. A mentor training program is described in Osborne et al. [2010], where the undergraduate students act as mentors in the outreach program RoboCupJunior. Another outreach program is described in Salamon et al. [2008] where LEGO[®] MINDSTORMS[®] NXT are used as the robot platform. These programs typically focus on building robots, but seldom focus on robotic applications or control algorithms. The iRobot Roomba was evaluated and recommended for projects focusing on computational and applied facets of robotics in Tribelhorn and Dodds [2007].



Fig. 2. iRobot Create (photo courtesy of iRobot).

In Dickinson et al. [2007], a course at Brown University is described that uses iRobot Roombas to introduce undergraduate students to autonomous robot control. Others have presented the results utilizing the iRobot Create in robotics applications, see Houston and Regli [2008] and Kuipers [2009]. With the popularity of “Hacking Roomba”, see Kurt [2006], the iRobot Roomba became a target for “hackers” who wanted to make the household appliance perform other functions. This gives working with the iRobot platforms a certain “cool” factor that seems to resonate with the high school students as well as mentors. In Matarić et al. [2007], the authors introduce the robot programming workbook to be used in conjunction with the iRobot Roomba or Create. The excellent introductory book “The Robotics Primer”, see Matarić [2007], is recommended as a reference for any students interested in robotics. In Houston [2008], the author provides a detailed course plan utilizing iRobot Creates as teaching tools for a high school introductory course in robotics. Our work here is meant to supplement these works rather than to replace them.

The remainder of this paper is organized as follows. Section II contains a detailed description of the iRobot Create including summary of costs. The details of the student projects and evaluation methods are provided in Section III. Section IV discusses the outcomes of this work, provides results of the student evaluations, and mentor insights and experiences. Finally, conclusions and directions for future improvement of the program are presented in Section V.

2. IROBOT CREATE

2.1 Overview

The iRobot Create, shown in Fig. 2, is an inexpensive mobile robot platform with several sensors and actuators that are accessible through a Serial Command Interface, see iRobot [2006]. The Create is a complete package that is ready for projects right out of the box without the need for further modification. However, it does offer accessories and expansion slots to enable modification.

2.2 Sensors

The iRobot Create is equipped with a suite of sensors and inputs that allows it to sense its environment and provide

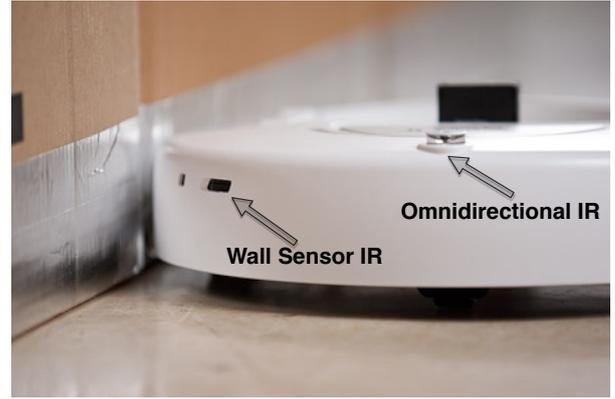


Fig. 3. iRobot Create with wall sensor and omni directional IR sensor.

Table 1. iRobot Create Sensors and Inputs

Sensor / Input	Number Available
Wheel Encoder	2
Wall Sensor IR	1
Omnidirectional IR	1
Cliff Sensor	4
Bump Sensor	2
Wheel Drop Sensors	3
Buttons	2
Digital Input	4
Analog Input	1

feedback for control algorithms. A list of these sensors can be found in Table 1. There is an encoder located on the left and right wheels to provide odometry data. On the right front of the robot one can find a reflective infrared (IR) distance sensor that measures the distance from the robot to a wall. The output of the wall sensor IR is an unsigned 16-bit integer representing this distance (See Fig. 3). Fig. 3 also shows the omni directional infrared sensor, which detects the virtual wall signal as well as inputs from the remote control and home base. The cliff sensors can detect if the robot is near a ledge. These sensors can also be used to detect landmarks on floors. When the front of the robot touches an object, the bump sensors will detect it. The Create can also detect when one of the wheels has dropped off a ledge. The start and power buttons can provide inputs that can be used to start algorithms. Access to four external digital signals as well as a 10-bit analog signal are provided by the cargo bay connector.

2.3 Actuators

The iRobot Create is equipped with a set of actuators that allow it to effect its environment. The key actuators are the left and right drive motors attached to the wheels, which provide a differential drive system. The three light emitting diodes (LED) located on the top of the Create can be controlled. The advance and play LED can be turned on and off, and the intensity of the power LED can be controlled. Finally, the Create is equipped with a speaker that can produce a wide range of tones.

Table 2. iRobot Create Cost Summary

Component	Price (USD)
iRobot Create	\$129.00
iRobot Create w/ Battery	\$219.00
iRobot Create w/ Premium Development Package	\$299.00
BAM Wireless Accessory	\$59.99
USB Bluetooth Radio	\$39.95
Battery	\$69.99
Battery Charger	\$39.99
Command Module	\$59.99

2.4 Communication

The Serial Command Interface specification describes how to access sensor information and command the actuators via a serial connection between the control computer and the robot, see iRobot [2006]. There are three ways to establish this serial connection with the Create. The first is through a proprietary serial cable that ships with the Create. This option requires a serial port on the computer or a USB to serial converter dongle. This method has the advantage of being a guaranteed connection, while it has the disadvantage of requiring a tether between the control computer and the robot. The second method is to use a Bluetooth[®] radio on the computer paired with the BAM wireless accessory available for purchase from iRobot. This device plugs into the cargo bay connector and allows wireless control of the robot. The third option is to program the 8-bit, 18MHz microcontroller in the Command Module directly using C.

2.5 Cost

The low cost of the Create is a primary advantage for this device. The costs of several relevant items from iRobot are provided in Table 2. The iRobot Create can itself be purchased for only \$129.00, but requires 12 AA batteries to operate. To purchase the high capacity rechargeable battery independent of the Create costs an extra \$69.99 for the battery and \$39.99 for the charger. Another option is to purchase the Premium Development Package which comes with a battery, charger, home base, remote control, two virtual walls, and a command module for \$299.00. To accommodate the four student interns, five units were purchased, allowing an extra unit in case of failure. This purchase included one Create Premium Development Package and four Creates with the battery packs. To avoid the need to tether the controlling computer to the robot, the BAM wireless accessory and USB Bluetooth[®] radios were purchased for \$59.99 and \$39.95, respectively for each unit. The total cost incurred to support four students came to \$1574.76. This cost summary assumes that each student has access to a computer with Bluetooth[®] or USB. If computers must be purchased as well, netbooks could potentially be purchased for around \$200.00.

3. STUDENT PROJECTS

The six week internship was divided into three projects, each taking approximately two weeks. To help the interns organize their time, a target schedule was provided. Each project and relevant concepts were introduced by a mentor on Monday morning during a three hour interactive



Fig. 4. Project 1: The robots are shown in initial formation with virtual wall used to synchronize the beginning of the ballet program.

session. On Tuesday and Thursday mornings the mentors were available for a one hour question and answer session. The interns were expected to use the remainder of the time on these days to work as a team. On Wednesday and Friday the mentors worked with the students to provide guidance where necessary. On the second Friday after introducing the project, the students were expected to demonstrate their solution. While the projects had very specific objectives, they were designed to confront the students with a large solution space, forcing them to make design decisions and then validate these decisions through experimentation and testing. As the internship progressed, the complexity of the projects increased and each new project required skills and knowledge gained in previous projects.

3.1 Project 1: Robo-Ballet

Learning Goals The goal of the first project was to get the students familiar with the robots and the serial command interface, while introducing the important concepts of path planning, synchronization, and the difficulties associated with open loop control. As Martin points out in Martin [2007], “real robots don’t drive straight.” This project drives this point home, and thus motivates the concept of feedback that is essential to the two subsequent projects.

Description The first project, called Robo-Ballet, required students to pick a song of their choice and program the speaker actuators of a team of four robots to play the song in a four-part harmony while moving in a choreographed dance to the music. Each robot can play only one tone at a time, so each student programmed their robot to play a different part (bass, melody, etc.) of the song, and each performed the same dance. The first step required the students to translate the sheet music into serial commands to be interpreted by the Create. The next step was to choreograph a robot dance that fits the music. The students then had to figure out a way to synchronize both the songs and the dance movements. The start of the program was triggered by the omni-directional IR sensors using an IR pulse produced by the virtual wall, see Fig. 4. This allowed all four robots to start at the same time. The final challenge to overcome was to pick the proper initial conditions such that the robots would avoid

bumping into one another as they got farther and farther off the planned path due to differences in motors, batteries, and tire frictions. These problems served to motivate the use of feedback in the second project.

Outcomes The students successfully demonstrated the Robo-Ballet with Ludwig van Beethoven’s Für Elise as the song of their choice. To succeed in just two weeks the interns accomplished many important subgoals. This project divided into sub projects nicely. One student worked on the wireless control interface, while other students worked on converting the sheet music to serial commands in java, and another student programmed the dance movements. In the end, all four robots were triggered to start at the same time and played the song correctly, but typically at least one robot would get out of formation and a collision would occur due to the open loop nature of the path planning.

3.2 Project 2: Robo-Race

Learning Goals The goal of the second project was to introduce the concept of PID feedback control while also emphasizing the need to include logic to check and handle sensor and actuator operating constraints.

Description The second project was a wall following race in which the students had to utilize the sensors on the Create to navigate a circuit with walls made of real walls, virtual walls, and tape landmarks on the floor. Fig. 5 shows an example course. A successful algorithm would navigate the entire circuit with no human intervention. This task naturally breaks down into sub projects. The first being implementing and tuning a PID loop to achieve wall following. The second involves designing a higher-level logic-based controller that looks for virtual walls, cliff sensor input from the tape on the floor, and takes appropriate actions when the wall sensor is out of range, such as corners.

The wall following portion of the project provided a perfect opportunity to motivate the use of PID control. The discussion of wall following in Chapter 10 of Mataric et al. [2007] was used to introduce the students to PID control. In a brainstorming session led by a mentor, the students were asked how they would follow a wall if they were given use of a sensor to provide a measurement of distance to the wall. A student blurted out “if the robot is far from the wall turn right, or if it is close to the wall turn left.” The student was asked to draw the trajectory the robot would take relative to the wall. The student drew the robot zig zagging about some desired distance. The students were then asked how they might determine how much to turn and whether the robot should always turn the same amount regardless of how far it would be from the wall. Another student proclaimed that it should turn more when it is far from the desired distance and turn little when it is close. The students were building an intuitive feel for what feedback control means before ever seeing an equation describing it. At this time the formal discussion of PID control was given, including motivation of Integral and Derivative terms.

Outcomes This project was intended to be a race between two teams of two, but the PID tuning and the debugging

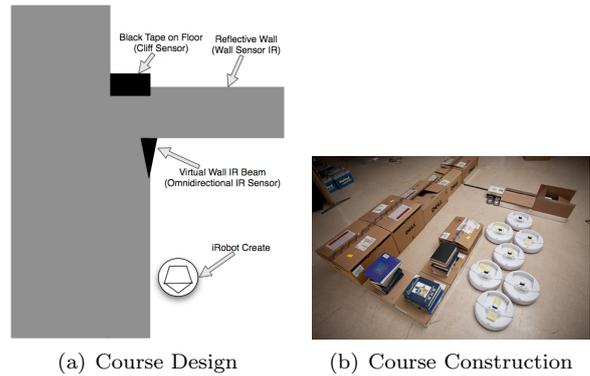


Fig. 5. Project 2: Complete a race course using PID feedback control for wall following while using logic for landmark navigation.

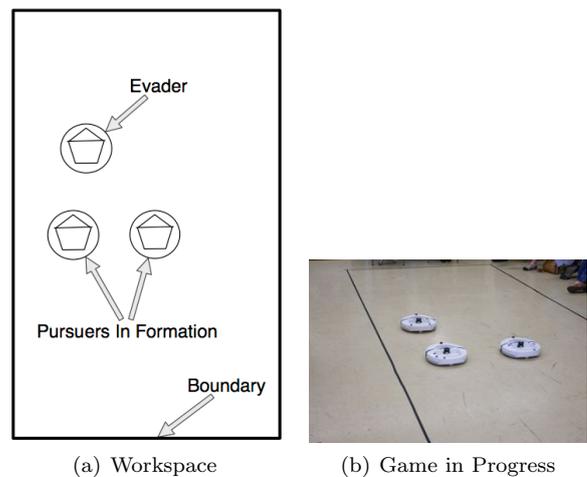


Fig. 6. Project 3: Pursuit evasion game in which autonomous pursuit vehicles were required to bump into a single human-controlled evader. The workspace is shown in (a) and a game in progress is shown in (b).

of the logic loop took longer than anticipated. In the middle of this project it was decided to combine the teams and make the objective to complete the circuit at least once without human input. This goal was achieved after considerable effort on the part of the interns. The students settled on using a proportional derivative (PD) controller in place of the PID controller and spent the rest of their time writing logic loops to navigate the corners and landmarks.

3.3 Project 3: Robo-Marco Polo

Learning Goals Project 3 intended to introduce the students to multiagent control and coordination. In order to achieve this goal the project incorporated concepts of coordinate systems and localization.

Description The third project was a pursuit evasion game in which a small team of autonomous pursuit robots were required to capture a single human-controlled evader robot. A successful capture occurs when any one of the pursuit robots bumps into the evader. None of the robots were allowed to leave a 2.1m by 6.3m rectangular workspace whose perimeter was marked by black tape



(a) Demonstration of wall following in Project 2. (b) Demonstration of coordinated pursuit in Project 3.

Fig. 7. Demonstrations of Projects 2 and 3.

to trigger the cliff sensors, see Fig. 6. The evader was controlled using the RoombaComm GUI, available at Kurt et al. [2009].

To locate the position of the individual pursuit vehicle and of the evader, the workspace was equipped with a VICON motion capture system. This system, often found in the movie industry, is capable of tracking all robots with millimeter accuracy. A simple Java client allowed the students to obtain real-time data from the system. If a VICON motion capture system is not available, an inexpensive web camera can be used for localization as in Houston and Regli [2008].

Outcomes A successful pursuit strategy requires a high level of autonomous planning, feedback, and coordination. After introducing the project, the students were given a few days to develop a strategy. The students devised a pursuit strategy in which the pursuit vehicles would “mow the lawn,” moving down the length of the workspace in a line formation. Given four properly spaced pursuit vehicles, they determined that there was no way for the evader to escape.

However, the strategy devised by the students was overly simple and did not utilize any of the feedback control techniques from the previous modules. To encourage further thought, the students were challenged to modify the lawn mower strategy to work with just two pursuit robots. With time running out, the mentors worked closely with the students to devise a “modified lawnmower strategy” in which two pursuit vehicles used advanced formation control algorithms to align their formation center with the evader while continually marching up and down the workspace. The formation control algorithms implemented by the students came from the technical academic paper Lalish et al. [2006], which the mentors helped the students to understand.

4. RESULTS

4.1 Final Demonstration and Presentation

At the conclusion of the program the students demonstrated each project in front of an audience of approximately 20 people including parents, professors from UCSB, and members of the Dos Pueblos Engineering Academy. Fig. 7 shows the demonstrations of wall following and



(a) Student explaining goals of Project 3. (b) Student explaining the propagation of the evader.

Fig. 8. Student presentations during final demonstration.

coordinated pursuit. As part of the demonstration, each student described on a white-board how they contributed to each project (See Fig. 8). This was an excellent way to build confidence and presentation skills. For the first time the students took time to reflect on how much they had accomplished in the past six weeks. This exercise also helped the students to better understand the projects through the act of explaining them to others.

4.2 Student Feedback

At the end of the six week program the students were asked to take a voluntary survey to help gauge the quality of their experience and to provide feedback on ways the program can be improved for future students. Three of the four students chose to participate in the survey.

When asked what aspect of the experience they enjoyed most, the students responded with these remarks:

“I enjoyed seeing how what I knew about programming related to programming robots. It was fun to try and tell the robot something to do and have it do the complete opposite. Then it would get really exciting trying to figure out what on earth went wrong and how to make it right.”

“I enjoyed the collaboration that was necessary to accomplish our projects, and the overall experience of amassing new knowledge.”

“The “team bonding” was cool, and I felt like I had done something when we saw what the robots did together in the end.”

When asked how best to improve the program, the students provided valuable feedback. First, it is important for all students to be in the lab at the same time. Each student mentioned that collaboration could have been improved if they were able to spend more time together in the lab. In the future, more structure in the schedule and the project durations will be provided. The students also became frustrated by issues associated with connecting to the robots via Bluetooth[®]. This may be alleviated in the next iteration of the internship by using the code written by this year’s interns as a template that manages the serial connection.

4.3 Insights

This is the first year for this robotics internship program at UCSB. Over the course of the program the mentors gained valuable insight in how to manage such a program.

The overall theme for mentors working with high school students is to remain patient and flexible.

At the beginning of the program the mentors developed a project plan describing the goals, the requirements, and the schedule for each project. As more was learned about the individual students, the plan was adjusted according to the students' interests and skills. While interviewing the students, it was discovered that all of them had an interest in music, and one of them had experience with ballet. This provided the idea to incorporate music and dance into Project 1. Later one of the students expressed an interest in developing video games. This led to the alteration of the plan for Project 3 to include the VICON motion capture system providing a connection to the video game industry.

It worked well if time was spent at the beginning of each project to explain the problem to be solved and the expectations. The students were then allowed some amount of unsupervised time to formulate a plan and begin working on the solution. It was necessary to help the students with distribution of work to help parallelize their efforts. The students had a tendency to work serially, slowing down the progress. When the pace slowed, the students became distracted. While this is to be expected, it is important for mentors to recognize when guidance is needed to get everyone back on track. This leads to another key mentor responsibility, being proactive. Even though the students were in the same lab as the mentors, they seemed reluctant to ask for help even if they were truly stuck. It is up to the mentor to check in periodically and to ask the appropriate questions to make sure everyone is progressing.

5. CONCLUSION

A project based robotics internship program was presented. This program used the iRobot Create platform as the primary tool and provided a set of projects found to be of appropriate complexity for advanced high school or even early undergraduate students.

Due to the success of this program, the Dos Pueblos High School and UCSB have decided to offer this program again next summer. Prior to the next iteration, there is room for improvement. First, potential interns will be identified earlier in the year to allow them to plan their summer activities with the internship schedule in mind. Next, the projects will be evaluated to identify any potential improvements. With more time, Project 1 could be revisited to include some basic form of feedback. With feedback, the students could implement a collision avoidance algorithm in conjunction with the formation control for the dance routine. The second project could have benefited from a better constructed race course. In the first iteration old computer boxes filled with books were used to make up the walls of the course. Next year, the plan is to construct modular walls that will make it easy to change the course configuration. A formal lesson on working in teams and the importance of breaking larger projects into smaller tasks that can be worked on independently will be provided. This should help to address the issues encountered on parallelizing the work.

REFERENCES

- R. D. Beer, H. J. Chiel, and R. F. Drushel. Using autonomous robotics to teach science and engineering. *Communications of the ACM*, 42(6):85–92, June 1999.
- B.C. Dickinson, O.C. Jenkins, M. Mosely, D. Bloom, and D. Hartmann. Roomba pac-man: Teaching autonomous robotics through embodied gaming. In *Proceedings of the AAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*, pages 35–39, Stanford, CA, March 2007.
- D. T. Houston. *Robotics toolkit for pre-college engineering education*. Msc. in computer science, Drexel University, August 2008.
- D. T. Houston and W. Regli. Low-cost localization for educational robotic platforms via an external fixed-position camera. AAI AI Education Colloquium, 2008.
- iRobot. irobot create open interface. [Online], 2006. Available: <http://www.irobot.com>.
- M. Kuipers. Localization with the irobot create. In *ACM-SE 47: Proceedings of the 47th Annual Southeast Regional Conference*, pages 1–3, 2009.
- T. E. Kurt. *Hacking Roomba: ExtremeTech*. John Wiley & Sons, Inc., New York, NY, USA, 2006.
- T. E. Kurt, P. Bouchier, and J. Pitts. Roombacomm java library. [Online], 2009. Available: <http://www.dprg.org/projects/2009-07a>.
- E. Lalish, K.A. Morgansen, and T. Tsukamaki. Formation tracking control using virtual structures and deconfliction. In *45th IEEE Conference on Decision and Control*, pages 5699–5705, Dec. 2006.
- F. G. Martin. Real robots don't drive straight. In *Proceedings of the AAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*, Stanford, CA, March 2007.
- M. J. Matarić. *The Robotics Primer*. The MIT Press, 2007.
- M. J. Matarić, N. Koenig, and D. Feil-seifer. Materials for enabling hands-on robotics and STEM education. In *Proceedings of the AAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*, Stanford, CA, March 2007.
- B.A. Maxwell and L.A. Meeden. Integrating robotics research with undergraduate education. *IEEE Intelligent Systems and Their Applications*, 15(6):22–27, Nov./Dec. 2000.
- R. B. Osborne, A. J. Thomas, and J. R. N. Forbes. Teaching with robots: a service-learning approach to mentor training. In *SIGCSE '10: Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 172–176, 2010.
- A. Salamon, S. Kupersmith, and D. T. Houston. Inspiring future young engineers through robotics outreach. GCER, 2008.
- B. Tribelhorn and Z. Dodds. Evaluating the roomba: A low-cost, ubiquitous platform for robotics research and education. In *IEEE International Conference on Robotics and Automation*, pages 1393–1399, March 2007.