Programming, Robotics, and Control for High School Students

David A. Copp, Jason T. Isaacs, and João P. Hespanha *

Abstract

Recent changes in Science, Technology, Engineering, and Mathematics (STEM) education in the United States call for an increase in the number of students pursuing degrees and careers in STEM. The Next Generation Science Standards (NGSS) integrate new science and engineering standards in an attempt to achieve these goals and are the first engineering standards for curriculum below the undergraduate level. Towards accommodating these changes, we present a project-based learning engineering program for high school students that teaches fundamental concepts in programming, robotics, and control. The students complete increasingly sophisticated projects that expose them to large solution spaces where creativity and exploration lead to unique solutions. The learning goals of the program align with the NGSS practices while also giving the students a head-start on material and key skills they will need when pursuing bachelor's degrees in engineering. We describe the projects, including learning goals and outcomes, as well as the hardware and software we used so that others can reproduce the program. Finally, we provide feedback from students as well as insights from the mentors.

Keywords: project-based learning, robotics, feedback control

1 Recent changes in STEM education

The landscape of science and engineering education in grades K-12 is changing in the United States. The concept of science, technology, engineering, and math (STEM) education was coined by the National Science Foundation in the 1990's after several decades of increased interest in innovation, technology, and engineering after the 1957 launch of the Russian satellite Sputnik and the following space race (Woodruff 2013). For the last two decades, various approaches to teaching STEM curriculum have been developed and evaluated. Many of these approaches were spurred on by national research and new national standards for K-12 STEM education. For instance, the National Science Foundation (NSF) published a "National Action Plan" in 2007 in order to address educational needs for STEM in the 21st century (NSF: National Science Board 2007). They note that addressing the STEM education needs of U.S. students is paramount in sustaining the economic success and national security of the country.

Most recently, the National Research Council (NRC) identified goals for STEM education in the United States as well as criteria and key elements of effective STEM instruction (National Research Council 2011). These goals include 1) increasing the number of students who pursue advanced degrees and careers in STEM fields, 2) expanding the STEM-capable workforce (including K-12 teachers in STEM disciplines), and 3) increasing the STEM literacy of all students, even those not pursuing STEM-related careers, in order to better prepare all citizens for a society driven by science and technology. The Next Generation Science Standards (NGSS) (NGSS Lead States 2013) are an attempt to achieve these goals. The NGSS include a framework that integrates the teaching of engineering principles with science education and identifies eight practices of science and engineering that are essential for all students to learn. This set of practices for K-12 classrooms are listed as follows:

1. Asking questions (for science) and defining problems (for engineering)
2. Developing and using models

*D. Copp and J. Hespanha are with the Center for Control, Dynamical Systems and Computation at the University of California, Santa Barbara (email: {dacopp,hespanha@ucsb.edu}). J. Isaacs is with the Computer Science Department at California State University, Channel Islands (email: {jason.isaacs@csuci.edu}). This work was partially funded by NSF Grants CNS-1329650 and EPCN-1608880.

3. Planning and carrying out investigations
4. Analyzing and interpreting data
5. Using mathematics and computational thinking
6. Constructing explanations (for science) and designing solutions (for engineering)
7. Engaging in argument from evidence
8. Obtaining, evaluating, and communicating information

Many innovative approaches have been developed and implemented for teaching students these core ideas and concepts including flipped classrooms, game-based learning, and inductive teaching approaches (Prince and Felder 2006), such as project-based learning (PBL), among others. In the next section, we describe PBL as an effective instructional approach for STEM education, in general, and teaching engineering and robotics, in particular.

2 Project-based learning

PBL is an innovative student-centered instructional approach that uses projects to motivate students to learn through inquiry. The projects are challenging problems or complex tasks in which the students learn problem solving, design, and decision making skills. The projects usually culminate in some kind of product, demonstration, or presentation. PBL focuses on the learning of concepts and problem-solving skills and less on the specific delivery of facts. The instructor's role, then, is not to lecture about particular material but, instead, to be a mentor or facilitator for learning (Frank, Lavy, and Elata 2003).

There are many advantages to using PBL as an instructional approach for STEM education. A survey of some of the initial research on PBL (Thomas 2000) shows that PBL may enhance the quality of student learning in comparison to other instructional approaches, and that PBL is a popular, beneficial, and effective instructional approach. Using PBL, students not only learn facts but also a way of thinking that can be applied to real-world problems. Students are motivated to apply the problem-solving skills, imagination, and creativity that they have learned and harnessed in the classroom to experiences with technology and society that they encounter outside of the classroom (Laboy-Rush 2011). It has been shown that students taught using PBL approaches perform at least as well as other students being taught using something other than PBL on math and science standardized tests, and they outperform those other students in solving applied and conceptual problems (Bell 2010). Students participating in PBL often work in groups, and when working in groups, students learn to negotiate among team members how best to collectively solve a problem. This requires students to define roles and accomplish their individual tasks to be sure that the team as a whole succeeds. The author of (Bell 2010) sums up these advantages well:

> "In the future, children must enter a workforce in which they will be judged on their performance. They will be evaluated not only on their outcomes, but also on their collaborative, negotiating, planning, and organizational skills. By implementing PBL, we are preparing our students to meet the twenty-first century with preparedness and a repertoire of skills they can use successfully." (p. 43)

There are also challenges with implementing PBL education programs. One challenge is overcoming existing teachers' attitudes against using newer or foreign instructional approaches as these teachers' attitudes have direct impacts on student learning (Laboy-Rush 2011). Assessing students' work can be difficult as it may be more subjective, and teachers may be most comfortable determining grades based on perceived objective methods like testing or achievement. Teachers also may be forced to learn new material that perhaps they have never been exposed to or does not come easily to them. These challenges can be overcome with teacher and professional development strategies and supportive administration and consultants who can focus on the needs of teachers when transitioning to new methods of teaching (Laboy-Rush 2011). Teachers specifically utilizing PBL may need to be flexible with project goals and more knowledgeable about specific material as unforeseen difficulties and spontaneous questions during projects may arise.

2.1 PBL for engineering

One of the features of the NGSS that makes it different from past education reform is its inclusion of engineering practices. Teaching engineering with PBL approaches seems natural given the nature of defining problems and designing solutions in engineering. Several organizations, including the NSF and the National Academy of Engineering (NAE), noted a need for engineering education reform in their visions for engineering in the 21st century (National Science Foundation 1997; National Academy of Engineering 2004). These mainly focused on undergraduate engineering programs as, until recently, there were no national standards for teaching engineering curriculum before the undergraduate level. However, there is some literature on pedagogy for engineering curriculum before the undergraduate level; two examples in robotics include the recent development and evaluation of a robotics course for junior high school students (Barak and Assal 2018), and the textbook (Matarić 2007) provides a pedagogical introduction to robotics for K-12 teachers and students. At the undergraduate level, there is a larger body of literature that highlights the successes of PBL approaches to teaching engineering curriculum. These include the teaching of engineering design and introductory freshman courses in undergraduate engineering programs. Some examples are described in (Frank, Lavy, and Elata 2003), (Dym, Agogino, Eris, Frey, and Leifer 2005), (Hadim and Esche 2002), and (Hassan, Domínguez, Martínez, Perles, Capella, and Albaladejo 2015). The outcomes and lessons learned from these undergraduate curriculum examples can help inform changes in the curriculum for younger students.

The authors of (Mills, Treagust, et al. 2003) highlight some criticisms of undergraduate engineering programs and discuss how the accreditation process should be updated to reflect the changing requirements for students to succeed in engineering careers after graduation. These requirements include strong communication and teamwork skills, having a broader perspective on social, environmental, and economic issues, and an understanding of how to apply the fundamental knowledge of science and engineering in practice. Undergraduate engineering programs may be too content driven and lack the proper integration of technical concepts with industrial practice and opportunities for gaining design experience. PBL addresses many of these concerns. Classes utilizing PBL give students a better understanding of practical applications for the engineering curriculum that they are learning along with other complexities involved in professional engineering practice (Hadim and Esche 2002). Students are more motivated to learn when participating in PBL and often demonstrate better teamwork and communication skills after completing courses that use PBL (Mills, Treagust, et al. 2003).

The field of robotics presents a particularly attractive area to apply PBL and thereby achieve the eight NGSS science and engineering practices. In particular, the opportunity to integrate computational thinking within the context of engineering problem-solving is invaluable for students who will be pursuing careers in our increasingly digital age. A review of teaching and learning of computational thinking for K-12 proposes problem-solving learning environments for fostering computational practices and perspectives (Lye and Koh 2014). Robotics provides natural problem-solving learning environments that require programming and computational thinking. Even specialized programming tools (e.g., Pyro, a Python-based programming environment (Blank, Kumar, Meeden, and Yanco 2004)) have been developed to ease teaching challenges associated with programming robots, such as the frequent requirement for unique code for platform specific hardware, sensors, and different application programming interfaces (APIs). Robotics projects are also inherently well-suited for the practices of defining problems, planning and carrying out investigations, and designing solutions because, as the author of (Martin 2007) notes, robots are feedback systems with a dual nature "as both deterministic machines and unpredictable entities." Therefore, robotics projects often involve defining seemingly deterministic problems that ultimately require investigation and creative solutions.

The rest of this article presents and evaluates a PBL high school engineering program that addresses the eight NGSS practices through teaching programming, robotics, and control.

3 A high school PBL programming, robotics, and control program

From 2010 to 2015, a summer project-based engineering program, called the Robotics Challenge, was offered in the Electrical and Computer Engineering Department at the University of California, Santa Barbara (UCSB) that gave rising senior high school students hands-on experience with robotics and control engineering. The program was six weeks long, with the students working about thirty hours per week. From 2010 to 2014, the program was a paid internship. In 2015, the program was unpaid but was still free for the participating students. Each year, the Robotics Challenge included at least four high school students, selected from a local high school, who were mentored by graduate students and/or post doctoral researchers from the Center for Control, Dynamics, and Computation (CCDC) at UCSB. Twenty-six total high school students participated in the program, of which nine were female students and seventeen were male students.

The projects in the Robotics Challenge are designed to not only lead students to understand core concepts, but to also build a foundation of basic understanding and problem-solving techniques that can be used effectively in more difficult projects later in the students' educations and careers. As there is not only one correct way to accomplish the projects, students are faced with a large solution space that forces them to be creative and make design decisions. Therefore, the students gain a better understanding of what practicing engineers actually do on a daily basis. The curriculum and desired accomplishments of the Robotics Challenge are intentionally left loose and open-ended so that students have opportunities to pursue their own interests and creative ideas within the context of the program.

Ultimately, the objective of the Robotics Challenge is to provide the students with a learning experience that accomplishes the integrated science and engineering practices of the NGSS, as listed in Section 1, while also motivating the students to pursue further STEM education. In the projects, multiple robots are used, and students work individually, in pairs, and in groups. The projects are designed so that the students need to define the problems to be solved, plan and carry out investigations, analyze data, and design solutions, all within the context of mathematical and computational thinking. Several projects culminate with a competition between the students or pairs or groups of students. This not only motivates the students to perform as well as possible, it also allows the students to see and reflect on how several solutions may accomplish the same task but may have different strengths and weaknesses. This also allows the students to evaluate and communicate their ideas. Specific examples of these practices will be given in the project descriptions below.

In the rest of this section, we present the hardware and software that was used in the Robotics Challenge, as well as the projects' learning goals, descriptions, and outcomes. We hope others find the projects and ideas presented in this paper to be useful for adapting new curriculum or extracurricular programs for high school students.

3.1 Hardware and software

The majority of the high school students who participated in the Robotics Challenge had at least some programming experience, often including an object-oriented programming class offered at their high school, so the majority of students had a working knowledge of programming in Java that greatly improved their ability to begin developing code. Because of this, we used the Java programming language in the Robotics Challenge, but other programming languages could have been used. On the other hand, the students had minimal to no experience working with hardware and software, so these robotics projects were intended to provide an introduction. The students worked on laptops running Windows 7 and developed code in the Eclipse Integrated Development Environment (IDE).

The robotic platform used for all student projects is the iRobot Create, shown in Figure 1. This platform has been used before as a teaching tool for introductory courses in robotics (Mataric, Koenig, and Feil-Seifer 2007), (Housten 2008), (Isaacs, Klein, and Hespanha 2011). On board the iRobot Create are several actuators and sensors that are accessible through a Serial Command Interface (iRobot 2006). The iRobot Create is a complete package that can be used right out of the box, but several accessories and expansion slots are available for modification.

3.1.1 Sensors and actuators

The iRobot Create is equipped with a suite of sensors that allows it to sense its environment and provide feedback for control algorithms. There is a wheel encoder located on each of the two wheels to provide odometry data. On the front right of the iRobot Create is a reflective infrared (IR) distance sensor that measures the distance from the robot to a wall. On the front top is an omni-directional infrared sensor that detects the virtual wall IR signal as well as the inputs from the optional remote control and home

Figure 1: iRobot Create platform

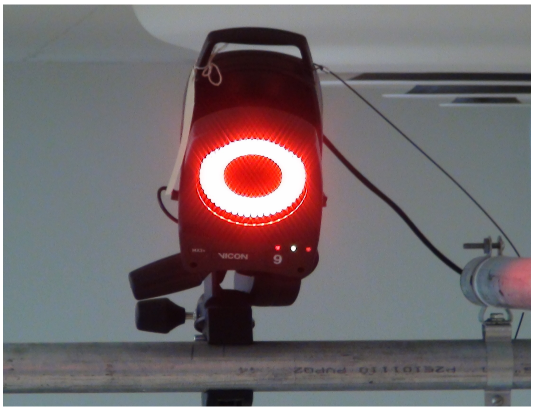

Figure 2: VICON camera

base. There are also cliff sensors to detect ledges and bump sensors to detect contact with objects. A wheel drop sensor detects when one of the wheels has dropped off a ledge.

A VICON motion capture system is used to accurately measure position and attitude of the robots, and these measurements are used as state feedback in the control algorithms. A VICON camera is shown in Figure 2. The measurements from VICON are communicated to the computer running the control algorithms via a network connection and a Java client. Control inputs are computed on the computer and sent to the robot using a Bluetooth® radio paired with a BAM wireless accessory that plugs into the cargo bay connector on the iRobot Create. The BAM wireless accessory is available for purchase from iRobot. Importantly, if a VICON motion capture system is not available, there are other solutions for localization of the robots, including less expensive camera systems, a simple web camera (Housten and Regli 2008), or something similar. Certainly global position information was crucial for some of the projects described below, but there are still many exciting projects that can be done using only on-board sensors or attachments; for some ideas, see, e.g., (Isaacs, Klein, and Hespanha 2011).

The main actuators on board the iRobot Create are the motors that drive the wheels. There are two wheels, and the iRobot Create uses a differential drive system. The Create also has three light emitting diodes (LED) located on the top of the Create that can be controlled and a speaker that can produce a wide range of tones.

3.2 Problem-based projects

In this section we describe the individual projects, the associated learning goals, and outcomes.

3.2.1 Project 1: Connect to, and communicate with, the robots

Learning Goals: The goals for project 1 include introducing students to the hardware and software that they use for the duration of the Robotics Challenge. Other goals include learning how to write basic classes and methods in Java.

Description: The students learn the process of initializing the communication between the robots and their laptops. The students write basic methods in Java that allow them to perform tasks such as reading the on board sensors, turning the LED lights on and off, and issuing drive commands to the wheels. Additional basic concepts like implementing key listeners and writing code to log data are introduced. During initial software development, the idea of laying out the code so that it is object-oriented, can be easily added to and extended for multiple robots in future projects, is stressed.

Outcomes: The students are usually eager to get started but quickly learn how frustrating the task of communicating between software and hardware can be. Once past this initial hurdle, the students enjoy seeing for the first time code that they have written affect the robots in desired ways. At this stage, some musically inclined students even write programs to have the robots play music using the on board speakers.

3.2.2 Project 2: Driving in a square

Learning Goals: The goals for project 2 include introducing the students to the robots' kinematics and feedback control. With the kinematic model of the robots, the students learn how changing quantities in the model affects the motion of the robots. Therefore, the NGSS practice of developing and using models is achieved. Other goals include learning the importance of collecting and analyzing data and making design decisions based on observations and feedback.

Description: The students learn about the kinematics of the robots and how to mathematically compute commands needed to drive and rotate the robots by changing the velocity at which the wheels rotate. Then the students try to drive the robots in a one meter by one meter square using open-loop commands such as "drive forward one meter," "rotate 90 degrees clockwise," etc. Students log and plot the data and compare the results to the observations they made while the robots were moving. Next, the students learn about feedback control and repeat driving in a square with feedback from the on board sensors. Finally, the students learn how to model the robots in the VICON motion capture system and how to query the robots' positions and orientations using a simple Java client. Then the students have the robots drive in a square again, this time with feedback from the VICON system. Again, the students log and plot the data.

Outcomes: This is often the first time students have learned about kinematics, and they are usually eager to do this project and have high expectations of success. The results, however, are not what the students expect. Often the pattern that the robots drive hardly looks like a square, and looking at plots of the data helps the students understand what happened. When asked why the results were so different from their expectations, the students thoughtfully respond with answers such as "the wheels were losing grip on the floor and causing inaccuracies in the wheel encoder data" and "the sensors are not very accurate." This project motivates the importance of accurate position information and feedback in order to effectively control the robots and naturally leads to the introduction of the VICON motion capture system.

3.2.3 Project 3: Parking (waypoint control)

Learning Goals: This project introduces students to proportional-integral-derivative (PID) control as well as how to design algorithms to achieve tasks with multiple steps. In this way, students learn how to construct, carry-out, and analyze the results of experiments by noting the procedure they are going to follow, their expected results, the adjustments they need to make, and the final results and lessons learned. This achieves the NGSS practices of planning and carrying out investigations and designing solutions.

Description: The students learn about how to design and tune PID controllers as well as how PID controllers are used by engineers throughout industry. Then the students design, implement, and tune two PID controllers: one for controlling the orientation of the robot, and the other for controlling the forward velocity of the robots. With these two controllers, and feedback from the VICON system, the students develop algorithms to park the robots at given locations, with millimeter accuracy, and with given orientations, with sub-degree accuracy. These tasks require applications of mathematics that the students have previously learned in school, such as geometry and trigonometry. This project can be posed as a competition in which students record how long it takes them to park at given points as well as how close they are to parking at the exact locations. Then, the student whose robot completes the parking maneuvers in the fastest time with the lowest error wins.

Outcomes: The students are excited to apply mathematical principles that they have learned in school to solving exciting, hands-on engineering problems and to compete against one another. The students comprehend the concept of PID fairly easily but can get frustrated when tuning the controller gains. Several principles to guide this process are helpful, such as starting with tuning just a proportional controller and logging and plotting the response, then adding an integral or derivative term, and repeating the process. For these tasks, often a proportional controller alone is sufficient to achieve acceptable performance. It is important that this project is completed well because the corresponding code is re-used throughout the Robotics Challenge. Furthermore, this project is the first to really lend itself to creative engineering solutions. For example, some students realize that driving in reverse may allow them to reach a parking location faster than first turning around and then driving to the parking location.

3.2.4 Project 4: Cyclic pursuit

Learning Goals: This project introduces the students to multiagent control and coordination. They learn to use the methods that they have developed so far in new and more advanced ways.

Description: Rather than parking at a stationary point, the students are now asked to park at a moving location, specifically, the location of a neighboring robot. This is a fairly simple extension of the parking project. Three or more robots are spaced equally apart, and each robot obtains the location of its closest clockwise neighbor and drives there. The positions of the robots are constantly updated as they are driving so that the robots are "pursuing" each other.

Outcomes: The students enjoy seeing the robots spiral toward each other and collide. They are also satisfied to see a new application leveraging existing code. Often this project provides a segue to introducing more advanced robot coordination, such as flocking, depending on the students' skills and motivations. Flocking is described later as an "advanced project."

3.2.5 Project 5: Circle (reference) tracking

Learning Goals: This project introduces reference tracking problems, which are ubiquitous problems in the field of control engineering. The students also learn to analyze control system performance by analyzing data and designing better solutions based on their analysis. This achieves the NGSS practices of analyzing and interpreting data, designing solutions, and engaging in argument from evidence.

Description: The students are tasked with driving in a one meter radius circle centered at a given point. This requires skills the students previously learned regarding the kinematics of the robot and PID control to design an algorithm that automatically adjusts the wheel velocities to follow the perimeter of the circle. The robots start a little bit off of the perimeter of the circle so that the robot must drive to the circle and then follow the perimeter. The students log their position data and plot a step response of how quickly they converge to driving on the circle. An example of this is shown in Figure 3. Then, the control engineering concepts of settling time, overshoot, and steady-state error can be introduced to quantify the performance of the students' controllers. This project can also be posed as a competition in which the criterion for winning is a combination of minimal settling time, overshoot, and steady-state error.

Outcomes: The students learn how to analyze the performance of their algorithms and how to design better algorithms based on experience. The students enjoy seeing a quantitative visualization of their performance, as the step response plots provide.

3.2.6 Project 6: Robotic transport

Learning Goals: This project gives the students a chance to learn and experience the different steps involved in an engineering design process. The students are faced with a high-level objective and are tasked with designing creative solutions that involve several open-ended steps. Throughout the project, the students test, evaluate, and modify their conceptual solutions.

Description: The students are asked to autonomously push a coffee can that is fixed on caster wheels into a circular goal. First, the students perform this task using a single robot, as shown in Figure 4. The robot and the coffee can are both cylindrical, so it is difficult to keep the robot in contact with the can. It is also difficult to stop pushing the can so that it remains in the circular goal without pushing it too far through the goal. The students split into pairs and brainstorm ideas on how to approach the problem. After brainstorming, the students meet with the mentors in order to describe the approach they think is best. The mentors assist by asking questions that the students may not have thought of and by refining the approach so that it is practical and implementable. Once the task is accomplished with a single robot, the students perform the same task but with two robots, as shown in Figure 5. Now the difficulty is not in keeping the can going in the right direction but instead keeping the robots in the correct positions to cooperatively push the can into the goal.

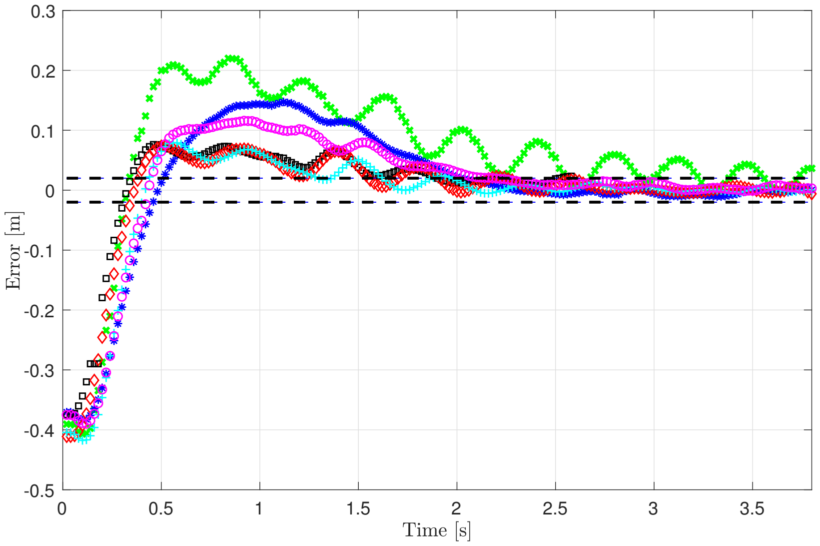

Figure 3: Experimental data showing the step responses of differently tuned PID controllers for circle tracking. The dashed lines show the desired error margin (±2 cm, in this case), where the error is computed as the 2-norm of the distance between the robot's position and the perimeter of the circle.

Outcomes: This is a prime example of an open-ended project with many possible solutions, so the students are faced with engineering design decisions; it is interesting to see what solutions the students come up with. One successful method involves lining up the robot on the opposite side of the can from the goal and pushing it straight to the goal until the can has spun off a significant distant from the front of the robot. Then the robot is realigned and the method is repeated until the can makes it into the goal. The most successful solutions involve dribbling the coffee can like a soccer ball, meaning, the robot is lined up behind the can and then pushes the can while constantly realigning to keep the can between it and the goal. In this way, the can usually never loses contact with the robot until it is successfully within the goal.

When performing the task with two robots, students are introduced to the concepts of cooperative, distributed, coordinated, and multi-agent control. In particular, students often implement a "leader-follower" algorithm in order to accomplish the task. In this approach, one robot, the "leader," knows the position of the can and the goal while the second robot, the "follower," only knows the position of the first robot. The first robot acts as a leader, or supervisor, and the second robot bases its actions off of the actions of the leader (e.g., "stay a fixed distance to the right of the leader"). Figure 5 shows this strategy in action. After learning how to coordinate the movement of the two robots, the students find the task relatively easier to complete than with a single robot.

Figure 4: Robotic transport

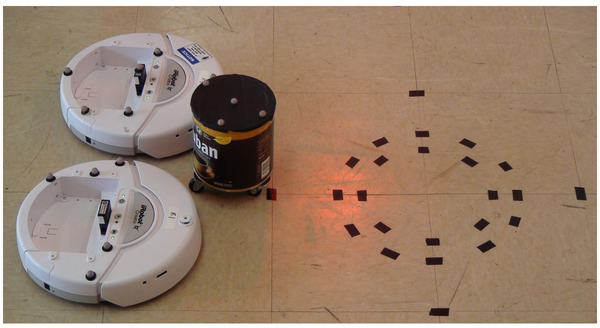

Figure 5: Cooperative robotic transport

3.2.7 Advanced projects

By week five, the students excel at programming the robots and often think of side projects that they are interested in working on individually or in teams. The mentors gauge these interests and encourage the students to pursue creative projects that they can accomplish in the remaining week or two. Two examples of these projects include flocking robots and the development of a graphical user interface (GUI) to control the robots from the console of the laptop.

Flocking: The objective of flocking is to get the robots moving randomly around the court at full speed, while avoiding collisions with each other, and eventually all moving together like a school of fish. One student read about flocking algorithms and wrote a simulation of particles flocking using the algorithms described in the paper. Then the students, with some guidance from the mentors, collectively figured out how to implement the same algorithms using six robots.

The solution concept started with one robot driving randomly around the court and bouncing off of the virtual boundary like a billiard ball. Then, other robots were added and used the same behavior to avoid colliding with each other as used when encountering the virtual boundary. Eventually, code was included that caused the robots to align their headings when they got within some zone of attraction of each other so that they all flocked together. This was very impressive and showed that the students were extremely motivated to try more complex projects based on the knowledge and skills that they had learned from the previous projects.

GUI for multiple robot waypoint control: Another year, a student was interested in developing a GUI that allowed a user to click in a window on the laptop to control the movement of multiple robots by calling several of the methods developed throughout the Robotics Challenge. A screenshot of this GUI is shown in Figure 6. This GUI allowed users to drive multiple robots around and also to try to get the robots to push the coffee can into the goal, all while virtually watching on the screen what was physically happening. This was also very impressive and showcased what the students could accomplish when they were motivated and provided the resources to explore their interests.

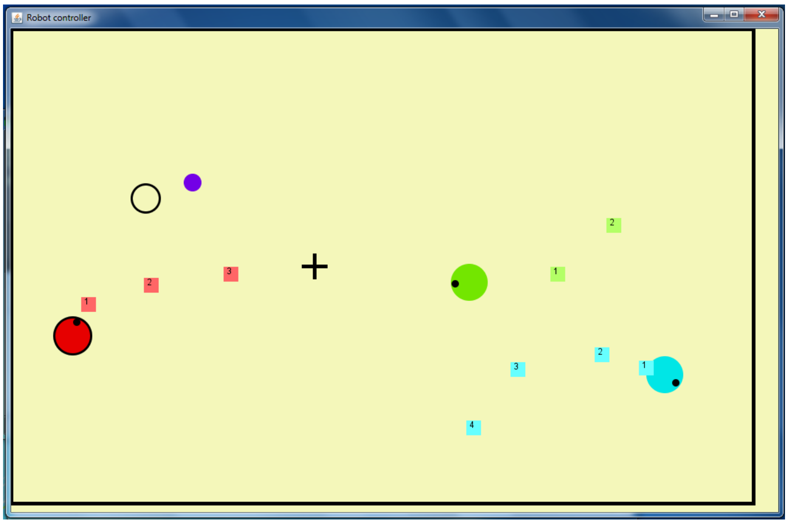

Figure 6: GUI for multiple robot waypoint control. The crosshairs are the cursor, the red, green, and blue circles are the robots, the smaller purple circle is the coffee can, the black circle is the goal, and the numbered squares are waypoints that the user has clicked for the robot with the same color to follow. The black dots on the robots indicate their orientation.

4 Program Outcomes

Besides giving the students an exciting hands-on engineering experience in general, the Robotics Challenge has several additional outcomes, and they are discussed in this section.

4.1 Final demonstration and presentation

At the conclusion of the Robotics Challenge, the students participate in a final demonstration and presentation to showcase what they learned and accomplished in the previous six weeks. In this way, the students accomplish the eighth practice of the NGSS by evaluating and communicating information. The students collectively decide how to run the presentation and alternate between presenting slides and demonstrating the robots moving in the lab. The students take turns explaining the projects and how they approached them. This gives the students a chance to reflect on and master the content that they have been exposed to during the Robotics Challenge. From the mentors' perspectives, it is satisfying to see the students take ownership of their individual and collective accomplishments when presenting to a general audience. The audience usually consists of about 20 people including graduate students and faculty and staff from UCSB, a few staff from the students' high school, as well as some of the students' family and friends. Often the faculty members and graduate students at UCSB are amazed at the abilities of the high school students to accomplish several projects in six weeks that would be considered advanced projects for undergraduate students. Several videos of some of the students' projects can be found here: `https://www.youtube.com/user/icbRobotics`

4.2 Teamwork and leadership

The only way the students were able to accomplish so much in just six weeks was through effective teamwork. The students learned to work together while individually leading certain parts of the projects, just as teams of engineers would. Some of the students were more comfortable and capable writing and developing code, while others excelled at designing algorithms. Some students enjoyed both. The students could focus on the aspects that they were most interested in individually and then work together on translating their solutions into working demonstrations, thereby leveraging the strengths of each team member. These skills will be invaluable as the students pursue careers in science and engineering where multidisciplinary teams are the norm.

4.3 Software development

In the context of object-oriented programming, the projects were designed to build on one another in order to also provide a natural way for classes and methods to be written and re-used throughout the Robotics Challenge. In this way, the students' code was re-used, and they learned how to build a software project from scratch. Additionally, since the students often worked together, they were introduced to version control software in order to maintain clean, working copies of their code. After each project, the students work together to decide which code to commit to their repository, deal with merging and resolving conflicts, and then move on to the next project ensuring they are all working with the same code base. We used a Mercurial repository, but a different repository, such as Subverion or Git, can just as easily be used. Methodically creating classes and methods and learning about version control will be useful for the students who go on to work on teams building software projects in the future.

4.4 Preparation for robotics competitions

Some of the students in the Robotics Challenge used what they learned in a robotics competition during their senior year of high school. The competition was the national *FIRST* Robotics Competition (`www.usfirst.org`), where *FIRST* is an acronym meaning "For Inspiration and Recognition of Science and Technology." The mission statement of *FIRST* can be found on their website, and it reads:

> "Our mission is to show students of every age that science, technology, and problem-solving are not only fun and rewarding, but are proven paths to successful careers and a bright future for us all."

This mission statement is well-aligned with the goals for STEM education described by the NRC and the practices of the NGSS, and the students who participated in the Robotics Challenge reported that it prepared them well for the *FIRST* Robotics Competition.

5 Student feedback and mentor insights

5.1 Student feedback

In 2012 and 2014, surveys were emailed to the students who participated in the Robotics Challenge in the prior two Summers. In total, eighteen students who participated in the program between 2011 and 2014 were emailed surveys, and thirteen of them responded. Some of the students who responded had just completed the program, while others completed the program a year earlier and had just begun college. This provided multiple perspectives on the program. All responses were kept anonymous. Some students may have chosen not to respond, and others may have not received the survey due to inactive email accounts. The survey included the following ten questions:

1. If you are in college, what school and what is your major? If not, where do you plan to apply, what major? Did your experience in the internship influence your decision on choosing that major?
2. If you could go back in time, would you do this internship again?
3. Would you recommend this program to your friends?
4. Please rate your experience.
5. Do you plan to take a control systems class in college?
6. What did you enjoy most about the experience? and the least?
7. What concept from the program did you understand the most? and the least?
8. What did you learn (added skills) that may help you with the rest of high school and college? Please describe.
9. How much and in what way did the hands on projects help you understand the concept of feedback control?
10. What can we do to improve the program for students next year?

Some of the quantitative results are shown in Figures 7-9, and quotes from some of the students' responses are given below.

Figures 7 and 8 show that all of the students who responded were "very likely" or "extremely likely" to recommend the program to their friends, and they all rated their experience in the program as "very good" or "excellent."

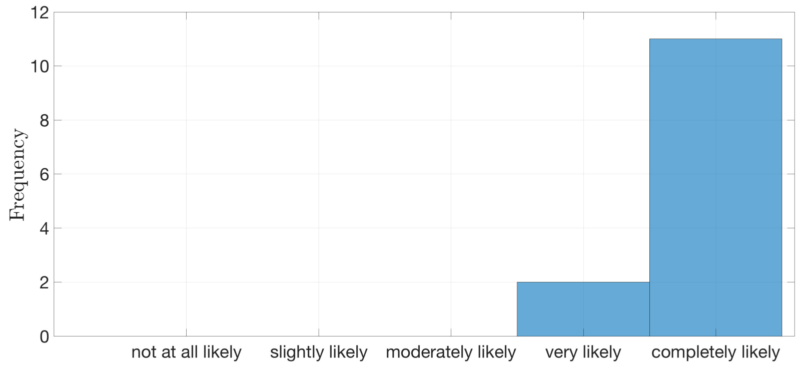

Figure 7: Student survey responses: Would you recommend this program to your friends?

Students responded favorably towards the PBL approach and the specific projects involved in the internship:

- "I really enjoyed the opportunity to develop advanced projects of our own design. There was great freedom and flexibility in what projects we worked on, particularly toward the end of the six weeks."

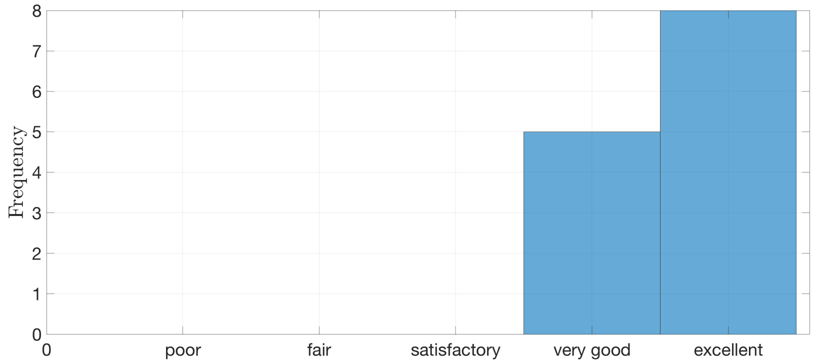

Figure 8: Student survey responses: Please rate your experience.

- "I enjoyed the variety of projects–it's much more interesting when projects have some element of creativity unrelated to the central crafting of code itself. I also really enjoyed giving presentations, as it made the projects feel worthwhile on another level."
- "I loved the hands on learning experience. The vast majority of our time in school is spent reading textbooks and taking tests. Textbooks and tests have their time and place however I believe they are overdone in school so it's nice to get a break from that. The internship also showed me that learning really can be fun."

These responses highlight the advantages of PBL including motivating students to learn and providing them with space to design and test creative solutions.

The students' responses also highlight the teamwork and collaborative elements of the program:

- "This [program] was my first experience with programming in groups, and I learned how challenging yet rewarding this can be. My ability to explain myself and my ideas also improved."
- "Learning what it takes to work with a group on technical projects like this, though, will be incredibly valuable to me, both during build season [for the *FIRST* robotics competition] and going into college. I suspect that I will have to work in a group setting frequently, and this experience will help me transition into this mindset more easily."
- "In addition to all the programming concepts I learned, I learned how to work cooperatively with other programmers and how to meet deadlines."

It is clear from these responses that the students who participated in the Robotics Challenge learned several of the important skills needed for engineering careers in the 21st century including collaborating on a team as well as communication and presentation skills.

Figure 9 shows that all students who responded stated that they were at least "slightly likely" to take a control systems class in college. One student wrote:

- "This [program] helped convince me that computer science was something I was interested in pursuing even further."

This shows that the Robotics Challenge motivates students to learn more about computer science, robotics, and control. In fact, the majority of students who participated in the program decided to pursue higher education in computer science and engineering fields, some even at top institutions such as MIT, Stanford, UC Berkeley, Columbia, UCLA, and others.

The students leave the Robotics Challenge with a more complete idea of what engineering is and what it would be like to pursue a career in engineering or computer science. One student wrote:

- "The [program] gave me a better sense of the kind of work and challenges I might face in such a field [as computer science]."

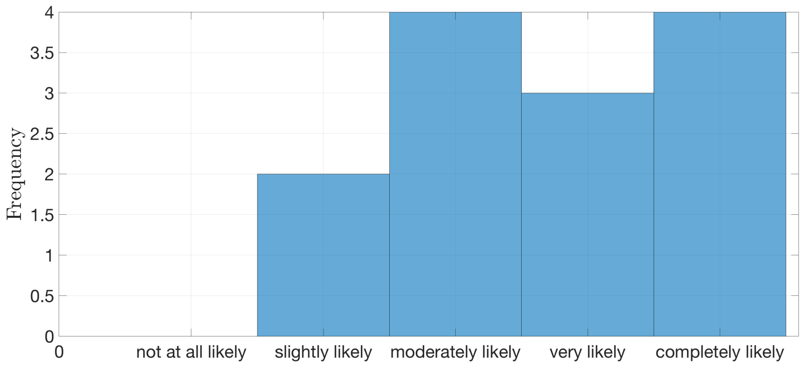

Figure 9: Student survey responses: Do you plan to take a control systems class in college?

5.2 Insights from mentors

One of the most exciting and challenging things being a mentor for this, or any other high school program, is that each year there are new students and, therefore, it is a completely new experience. What works well one year may not work with a different group of students. Therefore, being flexible with the projects and how they are implemented is important for a successful program. This may involve having pairs of students work together to accomplish a task or having students work individually, having students describe the big ideas of their approach to the mentors to see if they are on track, making the projects a competition between students to further motivate them, and knowing how much time to allow for the students to explore their own ideas before needing to implement something that works and move on to other projects. From a mentoring perspective, this kind of year to year flexibility makes teaching the material and mentoring the students exciting and new. Another challenge is that the students often do not ask for help and can lose focus if they get stuck. This requires the mentors to be proactive in offering help and advice to make sure the students stay on track. Ultimately, it is challenging to constantly update approaches and expectations depending on the successes, personalities, skills, and interests of the students, but it is very rewarding to see how proud the students are of what they accomplished when they do the final presentation and demonstration.

6 Conclusion

In this article, we discussed recent changes in STEM education in the United States and the particular benefits of using project-based learning for the teaching of engineering curriculum. We described a project-based learning high school program in which students learn about programming, robotics, and control with the help of mentors at the University of California, Santa Barbara. It was shown that this program incorporates the eight integrated science and engineering practices of the new Next Generation Science Standards. Student feedback shows that the internship is both fun and motivating. The students described the knowledge and skills that they learned as helpful to understanding what computer scientists and engineers do on a daily basis, and the majority of students who participated in the program decided to pursue higher education in computer science and engineering fields.

Acknowledgment

The authors would like to thank Professor Julie Bianchini in the Department of Education at UCSB for her insights and helpful comments on an early version of this article. The authors also thank, and acknowledge the hard work of, the other mentors who participated in the program including Daniel Klein, Steven Quintero, Justin Pearson, Hari Sivakumar, and Henrique Ferraz.

References

Barak, M. and M. Assal (2018). Robotics and STEM learning: students' achievements in assignments according to the P3 task taxonomy - practice, problem solving, and projects. *International Journal of Technology and Design Education 28*(1), 121–144.

Bell, S. (2010). Project-based learning for the 21st century: Skills for the future. *The Clearing House 83*(2), 39–43.

Blank, D., D. Kumar, L. Meeden, and H. Yanco (2004). Pyro: A python-based versatile programming environment for teaching robotics. *Journal on Educational Resources in Computing (JERIC) 4*(3), 3.

Dym, C. L., A. M. Agogino, O. Eris, D. D. Frey, and L. J. Leifer (2005). Engineering design thinking, teaching, and learning. *Journal of Engineering Education 94*(1), 103–120.

Frank, M., I. Lavy, and D. Elata (2003). Implementing the project-based learning approach in an academic engineering course. *International Journal of Technology and Design Education 13*(3), 273–288.

Hadim, H. A. and S. K. Esche (2002). Enhancing the engineering curriculum through project-based learning. In *Frontiers in Education, 2002. FIE 2002. 32nd Annual*, Volume 2. IEEE.

Hassan, H., C. Domínguez, J.-M. Martínez, A. Perles, J.-V. Capella, and J. Albaladejo (2015). A multi-disciplinary pbl robot control project in automation and electronic engineering. *IEEE Transactions on Education 58*(3), 167–172.

Housten, D. and W. Regli (2008). Low-cost localization for educational robotic platforms via an external fixed-position camera. In *AAAI AI Education Colloquium.*

Housten, D. T. (2008). *Robotics toolkit for pre-college engineering education.* Ph. D. thesis, Drexel University.

iRobot (2006). irobot create open interface. *www.irobot.com*.

Isaacs, J. T., D. J. Klein, and J. P. Hespanha (2011). A guided internship for high school students using irobot create. *IFAC Proceedings Volumes 44*(1), 5195–5200.

Laboy-Rush, D. (2011). Integrated STEM education through project-based learning. *Learning.com, http://www.rondout.k12.ny.us/common/pages/DisplayFile.aspx*.

Lye, S. Y. and J. H. L. Koh (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior 41*, 51–61.

Martin, F. G. (2007). Real robots don't drive straight. *Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*.

Matarić, M. J. (2007). *The Robotics Primer.* The MIT Press.

Mataric, M. J., N. P. Koenig, and D. Feil-Seifer (2007). Materials for enabling hands-on robotics and stem education. In *AAAI spring symposium: Semantic scientific knowledge integration*, pp. 99–102.

Mills, J. E., D. F. Treagust, et al. (2003). Engineering education - is problem-based or project-based learning the answer? *Australasian Journal of Engineering Education 3*, 2–16.

National Academy of Engineering (2004). *The Engineer of 2020: Visions of Engineering in the New Century.* Washington, DC: The National Academies Press.

National Research Council (2011). *Successful K-12 STEM Education: Identifying Effective Approaches in Science, Technology, Engineering, and Mathematics.* Washington, DC: The National Academies Press.

National Science Foundation (1997). Systemic engineering education reform: An action agenda. *NSF98-27,www.nsf.gov/cgi-bin/getpub? nsf9827*.

NGSS Lead States (2013). *Next generation science standards: For states, by states.* Retrieved from http://www.nextgenscience.org/next-generation-science-standards. (Appendix F).

NSF: National Science Board (2007). A national action plan for addressing the critical needs of the U.S. science, technology, engineering, and mathematics education system. *NSB-07-114, www.nsf.gov/nsb/documents/2007/stem_action.pdf*.

Prince, M. J. and R. M. Felder (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of engineering education 95*(2), 123–138.

Thomas, J. W. (2000). A review of research on project-based learning.

Woodruff, K. (2013). A history of STEM - reigniting the challenge with NGSS and CCSS. *U.S. Satellite Laboratory, www.us-satellite.net/STEMblog/?p=31*.