

Implementing a Robotic Programming Project in a First Semester “Programming for Engineers” Course.

William Schleter and Amy Biegalski

The University of Tennessee - College of Engineering - Jerry E. Stoneking *engage* Engineering Fundamentals Program
wschleter@utk.edu, biegalask@utk.edu

Abstract - Programmable robots have been added to the curriculum in the University of Tennessee’s Engineering Fundamentals Division as part of an introductory “Computer Solutions of Engineering Problems” course. The focus of the course is to gain proficiency in general programming and problem solving skills using MATLAB. The addition of a robotic programming project was intended to help students understand and apply the abstract programming concepts they have learned during the semester to a hands-on, fun, real-world team project. The use of a physical robot helps introduce concepts such as measurement errors, communications issues, and random variability associated with practical applications. iRobot Create2 robots were selected for the project based on their robustness, their familiarity as a relatively common household appliance, and because of the availability of an open-source MATLAB based toolbox and simulator. Capabilities of the robots include circular and straight line movement, the ability to play musical notes, an LCD display, physical “bump” sensors, reflectivity sensors, “light bumper” sensors, and an IR sensor. Onboard Raspberry Pi’s and video cameras were incorporated to add wireless communication and elementary image acquisition capabilities. Students were tasked with creating programs that enable the robots to complete three independent tasks – maze navigation, line-following, and an open-ended creative task of their own specification. This paper will discuss the implementation details, describe the lessons learned, provide a summary of student work, provide updated toolbox and simulator code, and discuss plans for improvements and changes for future classes.

Index Terms – MATLAB, Programming, Project Based Learning, Robotics.

IMPLEMENTATION METHODOLOGY

Most University of Tennessee (UT) Engineering students take a 2 credit hour course “EF 230 - Computer Solutions of Engineering Problems” as part of the Engineering Fundamentals curriculum. Students meet for 75 minutes twice a week in a computer lab to work through a series of labs [5] led by graduate teaching assistants. The course uses

MATLAB as the programming environment and is designed to give students familiarity with technical computing software and basic programming skills to solve engineering application problems. The software allows students to tackle realistic and complicated engineering problems, and most UT students will continue to use MATLAB in their upper level engineering courses.

There was a desire to implement programmable robots into the course curriculum as an interactive and fun tool that students could use to apply their programming skills and concepts learned in the classroom to a real world application, i.e. project based learning [4]. A pre-built robotic platform was sought so students could walk into a lab and instantly apply their programming skills to control a robot without build and prep time. iRobot Create2 robots were selected for the project based on their accessibility, robustness, their familiarity as a relatively common household appliance, and because of the availability of an existing open-source MATLAB based iRobot toolbox [1] and simulator [2]. The iRobot MATLAB simulator enables students to visualize the robots’ movements, outputs, and responses to programmed commands and environments in the MATLAB user interface.

Capabilities of the robots include circular and straight line movement, the ability to play musical notes, an LCD display, physical “bump” sensors, reflectivity sensors, “light bumper” sensors, and an IR sensor. To further enhance the robots’ capabilities, onboard Raspberry Pi’s and video cameras were incorporated to add wireless communication, live programming, and elementary image acquisition capabilities. The adaptability and minimal cost of Raspberry Pi’s made them the ideal choice for an onboard microcontroller. A “cargo bay” for each Create was 3D printed by UT’s Innovation and Collaboration Studio (ICS) to replace of the vacuum’s dust bin. The design for the bin was based on the iRobot provided downloadable code with custom openings added for cable routing. The Pi’s were then installed in the robot’s cargo bay with a mini USB Wireless Network Adapter and powered by a wired connection to the robots battery (Figure 1). Inexpensive video cameras were affixed to the top of the robots with the video output wired to the Pi. A video website was set up to show 24 hour live feed from all of the robots.

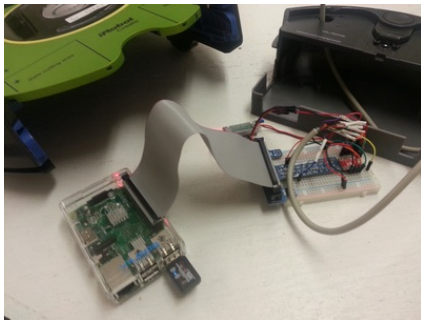


FIGURE 1
ENLARGED VIEW OF CREATE2 AND RASPBERRY PI



FIGURE 2
“FIGURE 8” TASK

STUDENT DESIGN TASKS

The simulator environment was introduced to the students first as a simple introductory team project with the goal to program the robot in the simulator to move in a square path. The students were able to type a few lines of simple code and readily view the motion in the simulator. Immediately they were able to visualize the concept of randomness errors, because the virtual robot does not make a precise 90-degree turn, and with each turn the angle of rotation differs. It was later learned that this same first exercise was utilized in the Cornell course where the simulator was first designed and implemented [3].

With the initial programming skillset the students were then able to explore the capabilities of the robots further and use the simulator in combination with the robots to build and develop their programs for the final project. The final project incorporated programming the robot to perform three independent tasks, a figure 8 line following exercise, completion of a random “maze”, and then the principal objective, performance of an open ended task of their own design. For the line following task, students utilized the robot’s reflectivity sensor to distinguish between a dark Figure 8 on the white course (Figure 2). For the maze, students utilized either the physical bump sensors or the “light” bump sensors which quantified distance from an obstacle, or a combination thereof (Figure 3). The final maze configuration was unknown until each class’s presentation day. For the project presentations, the maze and Figure 8 tasks were structured in a competition type environment, where student teams were pinned against each other with each task timed and judged for accuracy. Thus students had to examine the tradeoff between speed and accuracy. For the self-designed task students were encouraged to pursue tasks of high complexity that incorporated the knowledge base from their coursework, with grading based on the complexity and creativity of the task. A final component of the project was to develop a short self-running PowerPoint presentation exhibiting implementation of their creative task.



FIGURE 3
RANDOM MAZE TASK

USING A ROBOT TO PROVIDE DEVELOPMENT AS AN ENGINEER

The robots enabled the students to apply and build upon their programming knowledge through use of loops, functions, image processing, conditionals, logic, etc. The use of a physical robot helped introduce to students concepts many did not anticipate such as measurement errors and random variability associated with practical applications. For example, use of different robots, varying lighting conditions, and even repetition of the same command resulted in responses with discernable differences. Thus students were challenged to build adaptable programs to address the variability of the real world device. Further adaptability and innovation was required to tackle the maze challenge where the maze configuration was unknown until the presentation day. The open-ended task gave the students limitless freedom to use their creativity, design, and teamwork skills to both create and solve an engineering problem, and added entertainment value which helped motivate and encourage the students to succeed.

CHALLENGES (FURTHER DEVELOPMENT OF PROBLEM SOLVING SKILLS), PLANS FOR IMPROVEMENTS AND CHANGES, AND LESSONS LEARNED

As a work in progress and a new addition to the curriculum, a number of challenges presented themselves during implementation of the robots. First, the Create2 robot was released just a month before the start of the semester, on December 10, 2014, therefore the body of knowledge on the robot and its record of successful implementation in the classroom was limited. Due to the Create2’s additions and

modifications as compared to the iRobot's 2007 Create, many of the functionalities of the original MATLAB simulator and toolbox were not operational and therefore required substantial updating. The simulator code was built such that the simulator utilized separate programs for individual commands instead of the simulator "interpreting" the same code used on the robot. Therefore for each functionality modification, both the simulator and robot code needed to be reviewed and revamped. In trial tests, students were frustrated with the lack of compatibility between the simulator and the robot (apart from the randomness issues, such as marked speed and distance dissimilarities). To facilitate the students' experience, the simulator and toolbox were improved and expanded upon to provide more seamless use between the simulator environment and the robots themselves. Also a handful of student volunteers were selected to work ahead of the class to utilize and troubleshoot the robots and provided instructions, help with the camera and Pi functionality, and document their findings, in lieu of participating in the final testing day competition.

Unexpected issues occurred when many students worked simultaneously for extended hours on the robots, but proved to add to the real world engineering experience of troubleshooting and design adaptability. The wireless network in the testing room would often drop connections. The connectivity issues were dealt with through troubleshooting, use of a hardwired cable to laptops, and relocating the testing room to an area with a more reliable wireless network. With repeated, extended use, the robot batteries would lose charge. Battery issues were solved by being more attentive to rotating robots in and out of use, and using the robots while plugged into their charger.

Another obstacle was building access issues for students during the weekend preceding the presentation day when the building was not unlocked as scheduled. This proved to be a good lesson as the department transitions to another building on campus, and the decision was made that a future priority will be to enable all engineering fundamentals students all day card access to the Engineering Fundamentals laboratories, to encourage them to spend more time working together on creative and innovative endeavors. A lesson learned occurred due to the fact that the project testing days were scheduled during finals week. For some students, their lack of good time management skills, and more commonly for other students, the fascination and appeal of a robotics project, likely detracted from their studying and performance on other finals for other courses. To address this issue, it was determined that for future semesters the robotics project will be implemented earlier in the semester, which also adds the benefit of seeing programming codes in action and sparking student interest earlier in the semester.

Additional future work also includes a plan to develop a test suite that can be implemented on individual robots in order to document their response to specific commands and sensors. Then students will be able to have a tangible record

of what real world randomness entails and be able to customize their programs based on the data set for each specific robot.

A significant source of frustration was the 10-second dual button push required to reset the robots before and after use and whenever connectivity was lost or when robots would "run away" or repeatedly crash into barriers during program building and debugging. The reset operation on a "rogue" robot basically requires three hands, two hands to hold the robot horizontally in the air while the wheels continue to spin and one hand to press and hold down the two different buttons simultaneously on the top center of the robot for ten full seconds. This was addressed by adding a stop command to the toolbox, and students could also add a stop button to their MATLAB toolbar, but when connectivity issues exist the MATLAB functionality does not work, therefore a redesign of the reset will be recommended to the manufacturer.

Many improvements to the simulator are planned to enhance its effectiveness, such as adding the capability of multiple robots, incorporating tracking, adding a faux camera with image processing capabilities, and redesigning the simulator environment to be more robust such that it would "interpret" the code utilized in the robot without having to make behind the box changes in the simulator with code modifications.

PRELIMINARY RESULTS

An end of semester survey indicated that the students were pleased with the addition of the practical application of programming and valuable real world skills and found the project to be both fun and challenging. Students reported that the revamped course was beneficial to their engineering curriculum but that the workload was heavy for a 2 hour course.

The random variability of the robots, the implementation of a random maze configuration, plus the connectivity and battery issues frustrated many students, particularly those that waited until the final days to utilize the robots and programmed solely in the simulator environment upfront. Student groups who took the initiative to start early on the actual robots, experiment with different robots, and customize programs to deal with effects of randomness had much greater success on the presentation (testing) day. For future courses, this learning experience will be shared with all students upfront in the project in order for them to realize that there will be both expected and unanticipated challenges involved and encourage them to be willing to manage their time effectively, leave the comfort of the computer simulator and get into the hands-on environment of the robots in the testing lab, problem solve and improve their design, and develop more adaptable programs.

Many of the student groups chose a singing robot for their open-ended design task, some added choreographed dancing, which was not exceptionally innovative or creative, but the students did enjoy doing the work and the

entertainment value of the results. Other groups used the onboard cameras and image processing skills learned in the course for robot responses based on color recognition. Other tasks included drawing, knocking over or pushing objects, and robot “pets”. Due to the time limitations of the students, a limited frame of reference for what makes a good project, and the aforementioned challenges of launching the new project, the small steps in creativity and complexity are understandable and hopefully will improve with future implementation and planned improvements to the tools and resources. We hope that in the future students will utilize the enhanced toolbox to experiment with other capabilities such as image recognition, infrared, mapping, and calculations.

MATLAB TOOLBOX AND SIMULATOR CODE

The updated MATLAB Toolbox and simulator code can be downloaded from our course website:
<http://ef.engr.utk.edu/ef230-2015-01/projects/roomba/>.

SUMMARY

This was a hands-on, challenging, and fun project for the students that encouraged them to develop their problem solving and teamwork skills and helped build design skills and self-directed learning abilities. The use of programmable robots in introductory engineering coursework offers the opportunity to spark a lifelong interest in innovation.

REFERENCES

- [1] Esposito, Joel, MATLAB Toolbox for the iRobot Create, <http://www.usna.edu/Users/weapsys/esposito/roomba.matlab/>
- [2] Salzberger, Cameron, K-Y Daisy Fan and Hadas Kress-Gazit, MATLAB-based Simulator for the iRobot Create, <http://verifiablerobotics.com/CreateMATLABsimulator/createsimulator.html>
- [3] Fan K.-Y. Daisy, “Teaching Computer Programming to First-Year Undergraduates with a MATLAB Based Robot Simulator”, Mathworks Newsletter, MathWorks, Inc, 2012
- [4] Prince, M. J., & Felder, R. M. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education -Washington*, 95(2), 123.
- [5] Schleter, W. & Biegalski, A., EF 230 Labs, <http://ef.engr.utk.edu/ef230-2015-01/modules/index.php>