120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013

Paper ID #6062

Putting the Fun in Programming Fundamentals - Robots Make Programs Tangible

Dr. Todd R Hamrick, West Virginia University

Todd has over 20 years of engineering experience in industry. He has worked in process engineering, product development, product design, and technical sales. He recently returned to academia, and today he teaches and advises first year engineering students at West Virginia University. His interests include transition from high school to college, hands-on STEM education including robotics, and first year engineering curriculum development.

Dr. Robin A.M Hensel, West Virginia University

©American Society for Engineering Education, 2013

Putting the Fun in Programming Fundamentals - Robots Make Programs Tangible

Abstract

Many university first year programs seek to integrate kinesthetic projects into their fundamental programming curriculum. This work describes an innovative hands-on method developed at West Virginia University for teaching fundamental MATLAB programming through inductive learning. Low cost, re-usable robotic kits were created using Arduino controllers and OWI robots. Projects using the kits required the fundamental programming skills taught in the course. Learning outcomes met or exceeded expectations. The robotic application students had higher median and mean exam scores as compared with students who took the conventional course, with significantly more students earning 100% on the final exam. Student feedback was also very favorable, with high marks given in all categories. Written student feedback indicated that the hands-on approach was highly valued. Many indicated that making the programming tangible made the concepts more clear.

Introduction and Background

West Virginia University uses a common first year program for its Engineering curriculum, which includes two consecutive introductory courses. The second course is primarily dedicated to teaching fundamental Matlab programming as an engineering tool. Such programs are customary in common first year curricula.[1-4] The course is project based, with projects generally requiring software input and output. In the conventional version of this course, students are given engineering problems to solve and apply what they learn about Matlab programming to solve them. While many of the problems are based in real-world applications and represent a variety of engineering disciplines, they are often theoretical in nature. They do not require students to build or construct anything.

Many engineering students are kinesthetic learners who learn best by using their hands, building and manipulating things, in order to understand concepts.[5,6] It appeared that these students were underserved in the conventional version of the course, so an effort to create more sensory opportunities were sought. Robotic applications seemed a logical choice in programming, and have become a popular educational vehicle in recent years. So a low cost robotic application was sought for a pilot course taught in the summer of 2012.

Robots have been used in education for some time. Penn State began using robots of its own design in its freshman program in the mid-1990's. Typical sub-group size was kept about three students with good success.[7] Northeastern University also uses project base learning, and incorporates semi-custom kits to teach programming and electronics. The hands-on approach was met with a high degree of student approbation.[4]

Louisiana State University uses project based freshman courses that include programmable controllers and small robots.[2] Louisiana Tech University and Portland State University implemented the use of Arduino controllers as a tool for teaching C programming. In their study student feedback was positive toward the project based approach, but students were frustrated by difficulties encountered in programming the Arduino. This could have something to do with having no textbook for the course, and researchers recognized that the programming materials needed improvement.[8]

Several inexpensive robot platforms are commercially available. The OWI arm kit is a 5-axis arm costing around $50. Researchers in Thailand successfully programmed an inverse kinematic joint controller in Matlab to interface with an OWI arm.[9]

The purpose for conducting this work was to develop hands-on projects to teach fundamental programming through inductive learning. Low cost, re-usable robotic kits were developed for this version of the course to serve as a pilot. Students were asked to rate their experience in the course, and if they felt that the delivery method was an effective way to learn fundamental programming. Learning was also measured by comparing exam scores with students who experienced the conventional version of the course.

Discussion

In the conventional version of the course, theoretical concepts are reinforced using computational projects, usually three per semester or two in the summer. Typical projects cover a wide range of topics, but do not require the students to build anything or for the computer to interface with physical world.

For the robotic application version of the course the same book and curriculum was used as is used in the conventional course. The difference was that robots were used for project work. OWI robot kits were purchased, along with Arduino Uno controllers, and breadboards. Each complete kit cost under $100, with many of the components re-usable in later courses. The major components are listed below, while the arrangement can be seen in Figure 1.

- OWI Robot Arm Edge kit
- Arduino Uno
- Breadboard
- Assorted electrical components
- Mounting boards and hardware

Function files were provided that allowed Matlab to individually interface with the digital and analog input and output pins on the Arduino controller. By wiring signals from the computer through a USB port to the controllers, students were able to turn robot motors on and off by using Matlab commands.

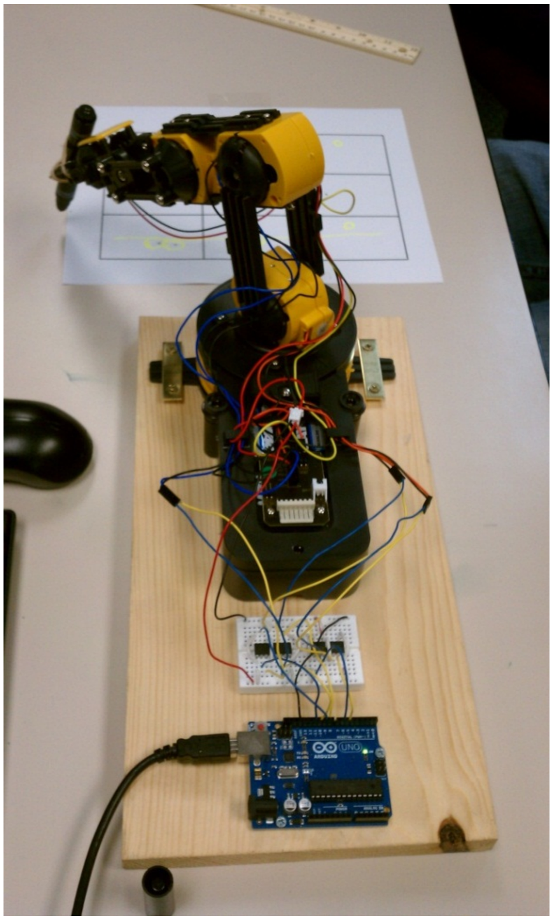

Figure 1 Assembled robot kit used in the Robotic Application version of the course. The computer is tethered by a USB cord to an Arduino Uno, which controls the robot motors.

Each kit was used for groups of three, giving each student ample hands-on opportunities. After building the kits, they collected and analyzed data on the motions of the robot arms, since DC motors do not work precisely the same in both directions. They collected precision and repeatability data on each motor, and used this information to program desired motion sequences. In order to move the robot arm from one position to another, programs had to turn on at least three different motors for various lengths of time. Retraction to the original position would require a slightly different reverse sequence taking the repeatability data into account.

Two projects were assigned in this six week summer session. The first project required all students to program the robot to play tic-tac-toe. This required them to program nine different move sequences to be called at will. Between each move the arm was required to retract to a home position in order to allow the opponent an open board on which to take their turn. Repeatability using the data that they collected in the early part of the project became a highly important part of the program. Programming skills taught up to that point in the semester were utilized in the programs.

The competitive nature of the project compelled students to boost the performance of the robots. Several teams learned programming techniques on their own in order to do more. For example, some modified the function files that they had been given in order to access more ports on the Arduino controllers. This gave them the ability to move more than three motors, giving them a higher level of precision. Others used graphical user interfaces not taught in class in order to simplify instructions given to the robots.

The second project was open-ended, allowing students to define the specific task, while meeting a set of given requirements. Because more skills had been introduced in class, more skills were required for the project. Students were required to construct a work environment for the robot that included position sensors either attached to the robot arm or to the work environment. The sensors allowed for a greater level of repeatability, and also necessitated loops and conditional statements in the programming. Again students exceeded expectations. For example, some students substituted the momentary switches provided with more advanced sensors for better precision.   Others used more complex looping structures than taught in the class to improve repeatability. The robot was required to do work in at least two areas of the work station, but most groups programmed the robot to work in more, including one that did work in nine different places in the envelope.

Examples can be seen in Figures 2 through 4. Figure 2 shows one of the open ended project displays. This project used the robot in parts sorting operation. It was designed to take an object from a dispenser and place it in one of four bins. The team’s technical poster is also on display on the screen.

Figure 2 Project display showing the team's working model and technical poster.

Figure 3 shows a different workstation display by a different team. This one was programmed to pick objects from one place in the workstation and place them in a different place. In this case small balls were used in such a way that a ball return could be used in the demonstration. This illustrated the repeatability possible using simple sensors. Both the robot motors and sensors were controlled using the programs written in MATLAB by the students. Figure 4 shows a project that was developed for an entertainment application. Again the robot motors and sensors were controlled using student written MATLAB programs. Students added artwork and supplemental lighting to demonstrate the potential use in a unique “fun house” application.

Figure 3 Workstation with ball return. The robot was programmed to pick and place balls on either end of a return, demonstrating increased repeatability through the use of sensors. The sensors and robot motors were controlled using MATLAB.

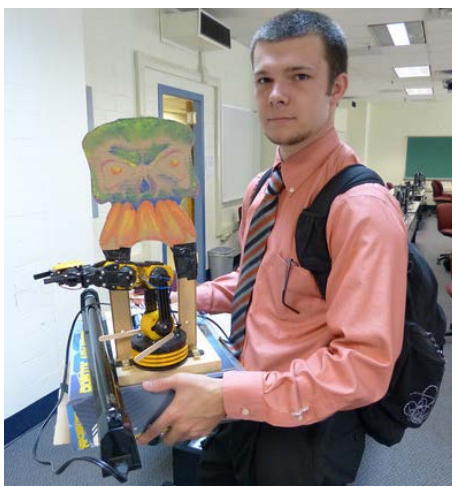

Figure 4 Student demonstrating a robot used in an entertainment application.

In both projects the programming was made tangible by creating physical requirements for the program. In the first project students generated data using a program to command robot motions. They wrote a different Matlab program to do statistical analysis on that data. They used that analysis to optimize the robot motion program to achieve the best possible repeatability without using sensors. In the second project, they made extensive use of loops and conditional statements. For example in order to make a motor run until the arm hit a sensor, a motor would be turned on with a Matlab command, and the program would enter an infinite while loop in order to sample the input from the sensor using conditional statements. When the sensor was triggered, the condition would be met to turn the motor off and exit the loop. Such physical requirements of the program made theoretical concepts much more concrete in the minds of the students.

Results

The results of a student exit questionnaire can be found in Table 1. Students rated their own learning in the course quite high. Specific to Matlab it was an average of 4.54, and specific to problem solving it was 4.5. They gave very high marks to their impression of the course as compared with what they knew about the non-robotic version, with an average of 4.75. They also overwhelmingly recommended continuing to teach the course in this way. In written responses students described the projects as fun, interesting, rewarding, and instructive. Because of the tangible nature of the projects, they reported that they were better able to understand how to apply aspects of programming to solve real problems. Many reported that they worked harder in the class because of the robotic projects.

Table 1 Results of Exit Questionnaire for Robotic Version of ENGR 102

		High				Low
Question		5	4	3	2	1
1	Concerning Matlab programming, rate your learning experience by taking the robotic version of this course	13	11			
2	Concerning Engineering Problem Solving, rate your learning by taking the robotic version of this course	13	10	1		
3	Rate your thoughts about the usefulness of the information you learned in the course outside of this class.	11	8	5		
4	Compared with what you know about the non-robotic version of this course, rate how effective you found this course	16	5	3		
5	Compared with your expectations, rate your overall experience in the course	12	8	4		
6	Rate your recommendations on whether the robotic version of Engr 102 should be continued	14	9	1		

Learning outcomes were confirmed in the final exam. A substantively identical comprehensive final exam was given to the summer students as had been given to students to previous semester in the conventional course. The summer students had a far higher percentage of students who earned 100% on the final, with 15% of the class who aced the final, verses 2% of the conventional course. The class also had a higher median and average on the final.

Conclusions and Further Work

The use of tangible projects is an effective tool for teaching theoretical concepts in fundamental programming. Students found the hands-on projects rewarding and informative, and improved test scores confirmed the educational value. A similar project that used the Arduino controllers but not the robots was implemented in another fundamental Matlab class with comparable results. The course will be repeated in the summer of 2013, with plans to make the class a regular alternative starting in the following semester. Plans are underway to create more hands-on projects for the fundamental Matlab course. These may or may not require the tethered Arduino controllers, but will require students to build models based on computational analysis.

References

[1] Dunne, B., Blauch, A., Sterian, A. (2005). The Case for Computer Programming Instruction for All Engineering Disciplines. *Proceedings of the 2005 American Society for Engineering Education Annual Conference &Exposition*

[2] Hall, D., H. Hegab, and J. Nelson. (2008) Living WITH the Lab - A Freshman Curriculum to Boost Hands-on Learning, Student Confidence and Innovation. *ASEE/IEEE Frontiers in Education Conference. Saratoga. Springs, NY. October 22-25, 2008, IEEE.*

[3] McKnight, S., Tadmor, G., Ruane, M., Rodriguez-Solis, R., & Saulnier, G. (2002). Creating 'High-Tech Tools and Toys Lab' Learning Environments at Four Universities. *ASEE/IEEE 32nd Frontiers in Education . Conference, 2:* S3E7-6.

[4] Jaeger, B., Freeman, S., Whalen, R., (2012) Programming is Invisible or Is It? How To Bring A First Year Programming Course To Life. *American Society of Engineering Education Annual Conference*, San Antonio, TX

[5] Felder R., Silverman, L., (1988) Learning and Teaching Styles In Engineering Education. *Engr. Education, 78(7), 674–681*

[6] Lumsdaine, L., Lumsdaine, E., (1995). Thinking Preferences of Engineering Students: Implications for Curriculum Restructuring. *American Society for Engineering Education, Journal of Engineering Education Vol 84, No. 2.*

[7] Avanzato, R. (1998). Collaborative Mobile Robot Design in an Introductory programming course for Engineers. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition,* Seattle, WA.

[8] Recktenwald G. & Hall D. (2011). Using Arduino as a Platform for Programming, Design, and Measurement in a Freshman Engineering Course. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition,* Vancouver, British Columbia, Canada.

[9] Prempraneerach, P., Kulvanit, P., (2010). Implementation of Resolved Motion Rate Controller with 5-Axis Robot Manipulator Arm. *The First TSME International Conference on Mechanical Engineering 20-22 October, 2010, Ubon Ratchathani*