Appendices For:
Robotics: Enhancing Pre-College Mathematics Learning with Real-world Examples
Submitted to 2013 Annual ASEE Conference
(A complete paper with the appendices will be posted after the paper is accepted).

Appendix A:

Technical Details of our Existing Courses:

As indicated above, the impetus for this came from a local school that had a robotic club, but no robots. To address the issues of affordability and self-repairing ability, the authors offered a course in Fall ’10 in which engineering students explored ways to build low cost robotic systems that were designed to be robust and reliable. The goal was that any hardware or software module that either did not function properly or was not meeting their goals, could be easily substituted with another one that was compatible at the level of interfacing (this is not difficult to do today, but still needs to be planned in; it cannot be an afterthought). The focus was on the design of hardware and software components to reach that goal. Most of the projects are fully documented along with videos[4]Our robot comprises of a DF Robot 2WD mobile platform, 2 DF Robot wheel encoders, Parallax Ping Ultrasonic (US) range sensor, Robotics DMS infrared (IR) short distance sensor, Arduino Uno microcontroller board, Arduino Protoshield, 2 H-bridge motor drivers, and a few other miscellaneous items, all secured from two suppliers [www.robotshop.com, wwwe.sparkfun.com]. Arduino is an open source initiative that provides significant hardware and software support for embedded system designers[14]. Arduino’s Sketch language and APIs provide programming support for the underlying microcontroller without exposing the user to the extremely complex and confusing register level details of the ATMega 328 microcontroller from Atmel[18]. The Arduino platform was developed primarily to aid artists (to control the stage props, applause, music, etc., for example) without overwhelming them with the underlying electronics. Extrapolating, it seemed like a good fit for both our engineering undergraduate students (from computer science, electrical engineering, and computer engineering), so they could focus on systems issues, and for high school students who have limited or no programming and electronics skills. See Figure 2 (in the paper) for the two robots that were built. There were differences, as described later; experiences with high school students’ class has allowed us to specify a $100 robot (without IR and US sensors) that is adequate for use in math classes at the 9th grade level. The robot can be made more sophisticated and useful for other applications, such as interactive gaming, with the aid of add-on components (such as IR and US sensors, as well as building blocks for wireless communication such as Bluetooth and Xbee).

Engineering pertinent experiences from our undergraduate level course: This class, offered in Fall 2011, had an equal number of students from computer science, computer engineering, and electrical engineering. Each team was typically comprised of one student member from each of these three disciplines. Such teaming helped them harness different strengths and also appreciate and accommodate different perspectives. Seven teams were formed. The students used a 1 m x 1m floor space with reflector walls encircling the space. The student blog sites and videos of their robotic art are available at our website[4]. Two teams focused on creating simple polygon patterns (a rectangle and a triangle); See videos of groups 1 and 6[4, 21]; this required effective use of the IR receivers and the ultrasound sensor for distance measurements and corrections for interference and potential avoidance of ‘drunken sailor’ behavior of the robot because of the IR receiver characteristic in near field (more on this later). A third team (group 3) created intricate polygon pattern that was repeated a large number of times[21]. See Figure 3 (in the paper). This required effective use of optical encoders and interrupts. The former did not yield clean ten pulses per turn, as would be expected. The electromechanical bouncing was evident even here and led to thousands of pulses. The students used back-to-back Nand gates and software delays to overcome the problem. Prioritization of interrupts had to be carefully thought through. Finally, while some teams complained of loss of precision in robotic movement when the batteries were somewhat drained, this group managed to create intricate patterns without losing control. Group 7 specialized in building a pen holder and the reflector walls that were used by other teams. In terms of lessons learned, the pen we chose was too bulky and the pend holder with pen ended up being a drag on the performance of the robot. Further, the use of four wheels caused large circles when the robot turned. It appears that other teams suffered because of lack of communication and coordination. However, we also recognized the need to help them plan and strategize, and also address issues in a piece-meal fashion. Experiences, both positive and negative, helped us identify a better kit and a focused syllabus when this material was used to teach a mechatronics course to high school students the following semester (Spring 2012).

This section discusses the transition from the first version of our robot to its second version (See Figure 2). The four wheel (first) version was used by the undergraduate students; the three wheel (second) version, shown on the right side, was used by the high school teams. The high school students worked in teams of three and used a large canvas (80 cm by 80 cm sheet with greeting card thickness) to draw (or plot) their art on. They used a color pen that could be mounted in the center of the robotic platform when needed (in the previous version, we used a larger pen at the back, which was too heavy and impacted the drawing; see above). The robots now had three wheels (this facilitated the drawing of smaller circles and reduced power consumption), with two motors that are driven (through an H bridge and a power amplifier) from the PWM outputs of the Arduino Uno microcontroller. The robot also had one US ping sensor mounted in the front and two IR sensors mounted on the two sides. The sensors were used to ensure that the robot traveled on a straight line; this is achieved with reflector walls placed to make a 1 m x 1 m square fence. The wheel encoders provided 10 positive going pulses per turn. Student teams used Sketch language to program their robots[12]. Undergraduate engineering students used interrupts and achieved intricate geometric patterns[6]. We intentionally restricted our 9th grade high school students to draw simpler patterns that repeated in some simple way, so we could show how an algorithm can be reused with changes in parameters. The goal then is to make sure the algorithm is rugged and that it leads to repeatable results and then use this algorithm in different contexts to create their desired pattern. For example, a triangle involves two algorithms repeatedly used, viz., to draw a line, and to turn an angle. The first one will need two parameters: distance and angle (the angle parameter is used to ensure that the robot does not drift from its intended path), while the latter one will need one parameter: angle, with the angles referring to a standard coordinate system.

Engineering pertinent experiences from our high school level course: This course attracted 17 ninth graders at the Henderson School, FAU’s University School on our Boca Raton campus. These students are in their pre-engineering program. This was the second engineering course that they had taken. The students comprised of 11 boys and 6 girls, with strong aptitude for mathematics and engineering. Five teams of 3 or 4 students were formed and were asked to choose a geometric pattern of medium complexity to draw with their robots. Each team built their own robot and developed algorithms for drawing lines and rotating by given angles. This required calibration of their robot's wheels for the distance traveled (at different duty ratios of pulse width modulation, or PWM) in a given time period, and the length of the PWM train to make a complete circle (again, at different PWM duty ratios). To make such a tight circle, the outer wheel was subject to this PWM train, while the inner wheel was held stationary. One could make fairly precise angular turns by controlling the duration of the PWM train. Different groups needed angles of 45, 60, 72, or 90 degrees depending upon the mathematical shape they were creating. Our initial goal here was to have the student teams use the reflector walls along the fence as guideposts; but the IR receiver characteristic has a negative slope in the near field. So, any simple algorithm that checks for a large reading from the near side IR sensor to keep the robot away from that wall ended up making the robot end up at the wall. It was difficult to convey the concept of a more sophisticated algorithm to the student teams because of the time pressure to get a working prototype completed. Another way would have been to use ultrasound sensor to measure the distance from the front wall and use that to correct the robot's path. Time pressure precluded us from incorporating this either. Thus the final student implementations ended up being 'deaf' to the surroundings that executed their motions purely based on control of the motors with PWM trains and different duty ratios. The five teams implemented four different patterns (Trinity Force of Courage, Butterfly, our university logo, and Star). Links to their blogs and demos are provided at two sites [4, 21].

Implementation Issues:

Errors involved in line-drawing with robots: Our goal is to use low cost robots in our work. Low cost robots will need to use mass manufactured parts in their kits. That brings in the issue of manufacturing variability. A wheel may not be precisely 2.5 cm in radius, as an example, and might have a tolerance of ± 5% in the radius. If one were to determine the distance traveled as being equivalent to a number of full and fractional turns, this tolerance will carry over to the distance traversed as well. A fractional or full turn can be estimated with the help of optical encoders. Such encoders will generate a pulse for a certain number of degrees of rotation of the wheel. However, there is finite resolution associated with this. As a consequence, there might be errors involved in quantifying fractional turns, which will add further ambiguity to the distance traversed. Also, in our elucidation of the Pythagorean Theorem, there will be need for physical turns of the robot by 90 degrees and 45 degrees. Such turns can also benefit from optical encoders, since it is a differential drive to the two sides of wheels that will bring about the physical turn. There could be errors in creating physical turns by these angles, furthering the ambiguity. These three types of errors may be considered to be systematic errors, but random with respect to each other. Thus, the total error from these errors is not an arithmetic sum, but will be based on the rms (root mean square) sum, and will be less. Our estimate is that for our given robotic kit, the error in estimation of the distance traversed will be slightly higher than ± 10%. This, however, is strictly based on the error compensation one can achieve with hardware alone. However, this can be improved in two distinct ways: (1) positional adjustment once the destination is reached, with the aid of an ultrasound range sensor; and (2) use of predictive algorithms in software to estimate fractional turns to a better accuracy than is feasible with feedback pulses from the optical encoder itself.

Course pedagogy issues:

1. We have taught project oriented courses for a number of years. The emphasis, before this spring semester, had been to give lectures, and institute a few quizzes, but essentially allow the teams to develop the project on their own. However, unlike the robotics course, the earlier groups (in non-robotics oriented courses) had the benefit of projects completed by earlier groups of students; that facilitated the thinking of the latest term's students. However, this was missing for the Fall 2011 students taking our robotics course, and some of their projects suffered. We changed our approach to teaching such project oriented courses in spring 2012. The first half of the semester was focused on the theory during the lecture hours, while they built the robots during the lab hours. Several quizzes and a mid-term exam evaluated the students on the lecture and lab material. This ensured that the students had an acceptable minimum level of understanding and competence when they started the project in the second half of the semester. The students also had a clearer idea of how they were faring in the class. These younger high school students may also have needed reinforcement of the ideas. Either way, it appears that we were able to help them in successfully completing their projects. This was a significant improvement over the results from the Fall 2011 semester with our undergraduate students from three different disciplines.
2. The high school students, under time pressure, discarded our recommended approach for use of infrared and ultrasound sensors and designed their algorithms merely based on control of distance traveled and angle turned. Successful completion of their projects is proof that high school projects can be undertaken with a simpler robotic kit. This brings down the cost from about $160 per kit (currently) to about $100, our targeted price point that should be met for schools to afford the robots.
3. The high school students seemed to have had problems with soldering, and thus had to use breadboards to connect wires from motors (and sensors) on to the Arduino Photo shield and the Arduino microcontroller board. It appears that we will use, in future course offerings, wire-wrap technology to make semi-permanent connections and avoid the potential for one of the bread-board wires to pop off and cause malfunctioning of the system.
4. Battery drainage was blamed by one student group for not drawing their star with proper angles, thus leading to a gap at the end. This group used 100% duty ratio and tried a large size star. Another team that drew a smaller sized star completed without any problems.
5. All these issues provide enough research material for an undergraduate robotics course in engineering at a later date. We expect to focus on building components or subsystems that can aid expansion of robot's role in STEM education for high school students. Possible examples of this are: robust sensor sub-systems for tracking; optical encoder-based interrupt driven sub-system for angular turns; and graceful slowdown based on battery voltage fall—off characteristics.

Appendix B:

Interview:

Five students who participated in the robotics course volunteered for a clinical interview so that we could document student learning in response to the robotics initiative.
Interview questions:
Students were first told the following:
"Let's start off with a simple problem. Please explain how you would solve it. How would you explain it to someone who is just learning the topic."

Once these initial instructions were understood, they were given the following problem:

Tamara leaves school and walks 6 blocks east. Then she turns left and walks 8 blocks north. How far did Tamara walk?

Tamara has a pet carrier pigeon that is trained to fly directly from Tamara's house to the school. How far does the pigeon fly?

Now imagine that this is going to be simulated by a robot moving on a map of Tamara's town. The robot is going to trace the path that the carrier pigeon flies. What distance do you expect the robot to travel?

Suppose the robot travels 9.3 units.
What would you say?

Suppose the robot travels 17.1 units.
What would you say?

Prompts
Do you expect the robot to travel exactly 10.0 units?
    If the robot travels 9.3 units, is something seriously wrong?

How would you define real-world error?

If the robot travels 17.1 units, is something seriously wrong?
    How would you define an error in mathematical understanding?

Results:

Evan and Karl worked as a group. The problem was read and then Evan was asked to solve the first part of the problem.

The following is a verbatim transcript of the interview. When necessary, clarifications are added to clarify the meaning, and these clarifications appear in italics.

Interviewer: Tamara leaves school and walks 6 blocks east. Then she turns left and walks 8 blocks north. How far did Tamara walk?

Evan: 10
Evan gave the direct distance.

Interviewer: OK [Pause]

Evan: 14
I thought that was what you were looking for.

Interviewer: You answered the next question.
The Interviewer is noting that Evan first gave the direct distance, 10, and then gave the walk distance, 14.

Evan: She walked 14 blocks

Interviewer: Correct. You answered the next question.
Tamara has a pet carrier pigeon that is trained to fly directly from Tamara's house to the school. How far does the pigeon fly?

Evan: 10 Blocks

Interviewer: Great.
Thank you Evan, that's wonderful.
Now..
That's the math problem.

Imagine that the robot is going to simulate this.

Karl: In blocks?

Interviewer: We will call the length of a block, 'one unit'
So 10 units.

Karl: Ideally, it should go 10 blocks.

Interviewer: Now, suppose the robots odometer reads 9.3 units.
Is there something wrong? What would you say?

Karl: Figure out the percentage?

Interviewer: Yes, please.

Karl: OK.
93…. Minus 7 per cent.

NOTE: Karl is considering 10 units to be 100 per cent, and 9.3 units is 93 percent.

Interviewer: Does that indicate a serious problem?
Is something seriously wrong?

Karl: I don’t believe seriously.
Slightly.
NOTE: Karl is responding to the question, “Is something seriously wrong?“.

Interviewer: Good.
Now, suppose the robot’s odometer reads 17.1.

Karl: It is seriously.

Interviewer: And if someone checked that and said it’s OK, that person is missing a step, right?

Karl: (nods)

Interviewer: Do you expect the robot in the real world … The answer to the math problem is 10, you got that right away.
In fact, you got it before I asked it!
But do you expect the robot to travel exactly 10.0 units?
With robots from commercial parts.
The goal of the project is to get this out as affordably as possible.
So under those constraints, do you expect it to be exactly 10.0 units?

Karl: No.

Interviewer: Great. And if it travels 9.3, is something seriously wrong?

Karl: No.

Interviewer: Good
How do you define real world error?

Karl: There are always certain factors that would affect the robot movement.
For example, the temperature of the air would affect ultra-sound.

The error is acceptable, but it depends upon what you are doing.

Interviewer: Now if someone checked it and said the robot traveled 17.1 units, is something seriously wrong?

Karl: Yes.

Interviewer: Can you explain that?

Karl: Because it is almost double the distance.

Interviewer: You shouldn't have that much error?

Karl: Yes.

Interviewer: So you’re saying: 10 percent error, that’s OK, but 70 percent error..

Karl: (nods)

Interviewer: Is there another way…You did the Pythagorean..
Please write down the three sides of the triangle.

Karl: Triples.

Interviewer: Can you say something else about them?

Karl: Well, they do follow the Pythagorean Theorem.

Interviewer: Do they share a common factor?

Karl: 3, 4, 5
Karl noted that the triples were 6, 8, 10. He implicitly realized the common factor was 2, and then stated that this is a 3, 4, 5 triangle.