122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society
Paper ID #12520

Precision Low-Cost Robotics for Math Education Work In Progress

Dr. Ravi T. Shankar, Florida Atlantic University

Ravi Shankar has a PhD in Electrical and Computer Engineering from the University of Wisconsin, Madison, WI, and an MBA from Florida Atlantic University, Boca Raton, FL. He is currently a senior professor with the Computer and Electrical Engineering and Computer Science department at Florida Atlantic University. His current research interests are on K-12 education, engineering learning theories, and education data mining. He has been well funded by the high tech industry over the years. He has 7 US patents, of which 3 have been commercialized by the university. He has published at the 2013 conference on this topic. This work is a continuation of earlier research. We plan to leverage this in developing a state-of-the-art course on the Internet of things for our undergraduates in Spring '15.

Mr. Jean Lapaix, Florida Atlantic University

I am a senior electrical engineering undergraduate at Florida Atlantic University. I am interested in science and engineering and applying them towards math education. I am also interested in incorporating control systems to make platforms more intelligent and robust.

Charles Perry Weinthal

Currently Seeking a Master's in EE Commodore Business Machines: EE: Engineering Services for Manufacturing /Production / Test-QA / Intern Training 1989-1994, Amiga / CD-TV /PC /C64/ C65 University of Central Florida: BSE-EE 1989 UCF Walking Machine Robotics Club: 1987-1989 Founding Member: - 2nd Place U. Va Nat'l Competition 1988 A.D. Henderson University School / FAU-High: Parent Mentor 2012-2015: Makerspace, FTC-Robotics, SeaPerch, EV-GT, Formula-E School Series, Quadcopters, MATE Girl Scouts USA: Co-Leader: 2005-2015

Dr. Don Ploger, Florida Atlantic University

DON PLOGER, Ph.D., is an Associate Professor with the College of Education. He and Dr. Ravi Shankar have worked together in developing Robotics courses for high school students. Don Ploger brings two important perspectives to this collaborative research. First, from an engineering education perspective, he emphasizes the importance of communicating essential knowledge to non-engineers. The second perspective comes from the mathematics education research literature. There is a well-established paradox: students often fail to apply familiar methods when they attempt to solve novel problems. Coordinating these perspectives has facilitated the collaboration across disciplines.

Ms. Malissa Augustin, FAU
Santiago Aguerrevere

Precision Low-Cost Robotics for Math Education
Work In Progress

Abstract:
A professor and several teachers in mathematics have collaborated with an engineering professor and his students at our university since 2011 in developing robots for math education. This project was started with the clear goal to develop low cost robots that use off-the-shelf commercial grade components and are thus easy to incrementally acquire, build, maintain and repair. Further, this robot, unlike currently available commercial education robots, would be built to be transparent in exposing the underlying math, physics, engineering, and technology principles. A group of engineering undergraduates first built low cost prototypes and explored alternatives for cost effective solutions. In a following semester, Seventeen ninth grade pre-engineering students worked in teams to build their own low cost robots (an improved version), program them and use them to draw various geometric shapes[1 - 3]. This course was designed to enhance their interest in engineering and math, while providing a social context of empowerment, competition, and cooperation. The results indicate that these students benefited from the use of robots. Two papers document the research results of student interviews to evaluate the effectiveness of this course [4, 5]. This will be described further below.

In the 2103 ASEE conference paper[1], a two-boat problem was examined to demonstrate how robots can be used for solving complex math problems in an intuitive and incremental manner. The problem is visually and dynamically solved. Successive approximation is used to identify a trend and come close to a solution. After examining the problem from multiple perspectives, the students become comfortable that the result that they have gotten with the robots is near to the mathematically correct solution. Students can stop with the robotic investigation at any point and solve the problem algebraically.

Earlier student interviews demonstrated that robots help students visualize challenging real world applications and secure multiple representations of a problem. They also develop a lasting hands-on experience in a social context and a better attitude towards math education and engineering realities. Building low cost robots that schools can afford would ensure access, availability and foster mainstream instruction with robots that would prepare our next generation in math and engineering principles.

This latest paper is focused on the final phase of engineering research, to build in precision in robots so the distances traversed and angled turned are mathematically exact. Problem solving can be significantly supplemented with robotics, even if a robot is imprecise, goals that are well appreciated by high school students who are in a pre-engineering program. However, it is also essential to make the robot a tool for teaching math to all students, so interest in math and

engineering can be enhanced for all. This is required to make the robots useful in classroom environment for teaching mathematics.

Several engineering enhancements already implemented and currently undergoing implementation will be covered in detail in the paper. These significantly improve the precision of the robot; here precision implies the yield of a repeatable solution. Ten graduate and undergraduate engineering students worked during this semester to improve accuracy and incorporate methods for error correction and detection to build a robot that solves a math problem much more accurately. Thus, the focus is not only making the problem solving exercise more repeatable and precise, but also enhance the eventual accuracy of the solution. This will be presented in more detail at the conference (as it is underway at present).

Concurrent to this engineering effort, research is also underway to develop math lessons that can be incorporated in a class environment. This will be covered briefly in this paper. This will also be covered in more detail in the presentation. The presentation will also be supplemented with student and teacher surveys (contribution of the fourth author, an undergraduate engineering student with interest in education research).

Background:

Mathematics plays an important role in high school education as it helps students develop the skill of problem solving. Problem solving is a useful and necessary skill in STEM fields which high school students may have interest in pursuing in college and as a career. But there is a dichotomy - mathematics is a precise science, and any problem solving engineering paradigm provides an optimal (or near optimal) solution. Anyone with an engineering perspective learns to appreciate this and continue to combine the two skills advantageously. However, not all students significantly develop this skill when learning math in their curriculum as they may not see the connection between the theoretical concepts in the subject and the practical problems associated with STEM fields. This lack of a connection could negatively affect the students' performance and interest in STEM. Our initial focus was to develop the robot as a tool for problem solving [1-3]. We also made sure that it is low cost and reliable so schools can afford to buy and repair. However, it soon became clear that the robot also should be precise, and accurate, for it to be useful as an educational platform to teach mathematics.

The motivation for undertaking this paper's research project thus stemmed from the desire to enhance high school students' retention and interest in Mathematics. Such qualities would significantly improve their performance in STEM (Science, Technology, Engineering, and Mathematics) career fields and education in general. Our exploration showed that much research has already been performed by other researchers to facilitate high schools in fostering STEM interest with robots [6-10]. However, such robots have tended to be expensive (~$300) by standards of a typical high school in the US and elsewhere. Further, their role in exploring fundamental principles of math is limited. Our experimentation with commercial educational robots showed that programming, and transparency into the underlying behavior, of such a robot is limited, with the results that are mathematically imprecise (and inaccurate). Our results on this are unpublished at present; however some related work is published in a graduate student's blog [11]. While there is universal agreement on the potential of robots to enhance STEM interest, the enthusiasm seems to be have been stymied by the non-availability of low cost and mathematically accurate robots. Perhaps the lure of commercial potential of high end educational robots has kept companies from exploring this option. There is still a large barrier for their use routinely in a class environment, because of their cost and sophistication, and the inability of a typical school to support teachers and students in their routine use. We believe that robots built with low cost and off-the-shelf components can reduce this barrier immensely. If, further, they are made (precise and) accurate, they will gain a major role in teaching math. This is based on the feedback we have received from all types of stakeholders, viz., parents, students, teachers, and administrators. A definite attempt will be made to quantify this with a survey in the near future.

Our extended research group has parallel activities underway to facilitate (middle and) high schools in using robots to enhance Math education. This collaborative research project involves several faculty members from engineering, mathematics, education, and K-12. It has led to the development of several iteratively improved platforms. Our goal has been to increase interest in STEM-related fields in high school students as well as teaching system-level design and integration issues to engineering undergraduate students. Using low-cost, imprecise components, a group of undergraduate students first built functional platforms (four years ago); this highlighted several issues associated with the components. A group of high school students (in a pre-engineering program), with no prior programming and electronics knowledge, then built and used an improved platform to draw mathematical shapes on a large 6' x 6' canvas. These experiences led to continued research to further develop the platform's precision and robustness. All of this is documented in earlier papers presented at education conferences and at our websites[1-5].

Our low cost (under $100) robotic platform [1-3] allows for hands-on demonstration of mathematical topics, such as Geometry and Trigonometry, which are taught in the classroom. Our initial demos to math teachers and pre-engineering students demonstrated the problem solving capability of the robots, especially for visualizing and exploring multiple perspectives. Understandably, this meets well with the needs of pre-engineering students in high school, who also now understand the errors in real world representations of mathematical concepts [4-5]. But a typical student in a middle or high school math class needs an accurate representation of mathematical concepts, whether it is done with a graph sheet, a computer simulation, or a physical robotic simulation.

Our low-cost robot is made out of off-the-shelf commercial components to keep the cost low; these components in turn have poor tolerance. Such poor tolerance leads to the solutions being imprecise during the robotic platform's execution. In this work, we document our solution to compensate for the poor precision in distance traveled and angle turned; for this, we use engineering approaches, viz., feedback control algorithms and custom (automated) calibration, to achieve this, while still keeping the cost low and the platform simple to utilize. This work by a DIS (Directed Independent Study) student (3rd author) has provided significant improvement; further improvement is being actively pursued at the algorithmic and robotic level by two graduate students (last two authors).

Concurrently, we are also exploring the use a network infrastructure to guide the robot to correct mathematical solution, using error detection and correction schemes. Such schemes have been used in engineering for other applications; however, our approach to apply it to teaching/ learning math is unique. This work is the result of a current engineering course that involves ten undergraduate/graduate students.

In summary, the robotic platform is being improved further in order to help demonstrate the concepts in the students' math curriculum in a more precise and accurate way. While there are already many robotic platforms designed to help educate students, the goal is to create one that is affordable for many middle and high schools while making it more robust, precise, and accurate for demonstration of a wide range of mathematical concepts. Not only will the platform perform such demonstrations in a cost-effective way but will also be used to teach engineering and programming concepts to the students as they learn how the platform operates.

Methods:

We document here the engineering enhancements that we had to undertake to improve the precision of the robotic platform, while reducing the cost and power dissipation (the robot had to be operational longer; we also found that the motors response is poorer when the battery power is low). Essentially, these changes involved removing ultrasound and infrared sensors, and improving the performance of the only remaining sensor, viz., the optical encoders on the two wheels. The microcontroller sends electrical pulses to the motors to turn the wheels; the encoder yields 10 optical pulses as the wheel makes one rotation, thus giving us the ability to track fractional rotations, whether in distance traversed or angle turned. Successful utilization of this required us to address certain interference issues so there are no extraneous optical pulses (as with contact debouncing and shielding), and incorporation of an engineering algorithm to adjust the number of electrical pulses sent to the motors based on the feedback from the number of optical pulses received. The algorithm's robustness is enhanced by considering the displacement, and its derivative and integral. Details are provided below; one may skip this section and go the 'Results' section if the engineering details are not of interest.

Algorithmic and Robotic Level Improvements: The objective of this project phase included reconstructing and redeveloping the robotic platform with improved precision while maintaining the affordability and simplicity of the system already built. Reconstruction consisted of analyzing past accomplishments and issues with the platform as highlighted from the research in order to maximize the effectiveness of the components associated with the platform. The next step was to implement methods of utilizing optical encoders to display data on speed, angle, and distance the platform has traveled in order to come up with novel solutions to increase the precision of the said platform. The final step was to incorporate PID (Proportional-Integral-Derivative) algorithm [12] to further increase the (distance and angular) precision of the platform in order to achieve an error value of less than one percent. All of the data and measurements are to be recorded wirelessly via Bluetooth communication on a monitor that the students can observe in easy-to-understand formats. Exposure to underlying algorithmic manipulation, which may be exposed with a button press, is also being implemented; this may help advanced students become involved further and explore other options for further optimization.

The expected outcome of the overall project is to convince several high schools that the robotic platform will reinforce their students' math education at an affordable price. With the incorporation of control algorithms, the platform has become precise enough to be an effective tool for demonstration of mathematical concepts. The students themselves will attain a deeper understanding of the concepts through the utilization of this platform, and thus gain more interest in STEM. However, exact mathematical solution is the ideal solution. This is the work underway in a class this semester and this will be briefly covered in the discussion section (details will be presented at the conference).

Reconstruction (For Improved Design): The first part of the project has already been completed. The robotic platform was reconstructed using most of the major components (used by the previous group of high school students) such as the Turtle Mobile Robot Platform by DFRobot and the Arduino microcontroller [13]. These components have proven to be sufficient in the hands of high school students for plotting artistic renditions, such as the Star of David or a Butterfly, with reuse of fundamental mathematical concepts[5]. Essentially, we removed the ultrasound and infrared sensors, which reduced the cost of the platform further, but kept the platform functional. However, this left the optical encoders as the only sensor for measuring distance. Also, the breadboard and additional Arduino shields (or boards) for wiring the components were removed to reduce the degree of messy wiring as well as the potential for students to unintentionally undo the wiring resulting in malfunctions. In their place was the inclusion of a wire wrap tool to wrap the wires to the components, hence eliminating the need for a breadboard and additional soldering. With this tool, students will be able to add their own components with ease to further enhance their use of the platform.

Optical Encoders: After construction, the platform was programmed to perform basic functions with a smaller degree of error while using fewer and more cost-effective parts such as the optical encoders. The encoders are each composed of an optical sensor aligned to a small, segmented disk and a light source. When the light source is interrupted by the disk, a digital signal is generated by the encoder. When attached to the motors of the platform, the encoders rotate along with the rotation of the motors. This allows the encoders to measure how much the wheels have rotated based on the number of times a pulse signal is generated in order to determine the distance the platform travels. They can then be used to determine how long the motors should be turned on so that the platform can travel a specific distance. They can also be used to have the platform turn a certain angle. For instance, the circumference of the platform's wheels is measured to be 8.05 inches. Since there are 10 segments on the encoders' disks, each time a segment interrupts the light source, a distance of 0.805 inch will have been traveled by the platform. Also, it takes about 2 revolutions to turn 360° which means each segment passed results in a turn of 18°.

The resolution of the encoders can be increased even further, resulting in 20 signal changes during one revolution of the platform's wheels which would allow for more precise distance and angle measurements. However, the encoder readings were initially found to be both inaccurate and imprecise in their measurements (more below). When the platform is given a high acceleration, the encoders generate a signal that bounces in between a high state and a low state hundreds of times within a few milliseconds even when the platform is not moving. This is known as contact bouncing. Also, the encoders change states during motion due to noise in the signal. For example, each time a segment of the encoder's disk interrupts the light source, an interrupt is sent to the Arduino microcontroller. The interrupt stops the main program and increments a variable to display how many interrupts occurred while the platform was in motion. The platform travels about 1 foot per second which is the length of about 15 segments. Therefore, the number of interrupts that should occur in one second of the platform's motion is 15. However, the Arduino displayed a range of 500 to 600 interrupts in one second. With these issues, accurate measurements of distance and angle were not possible.

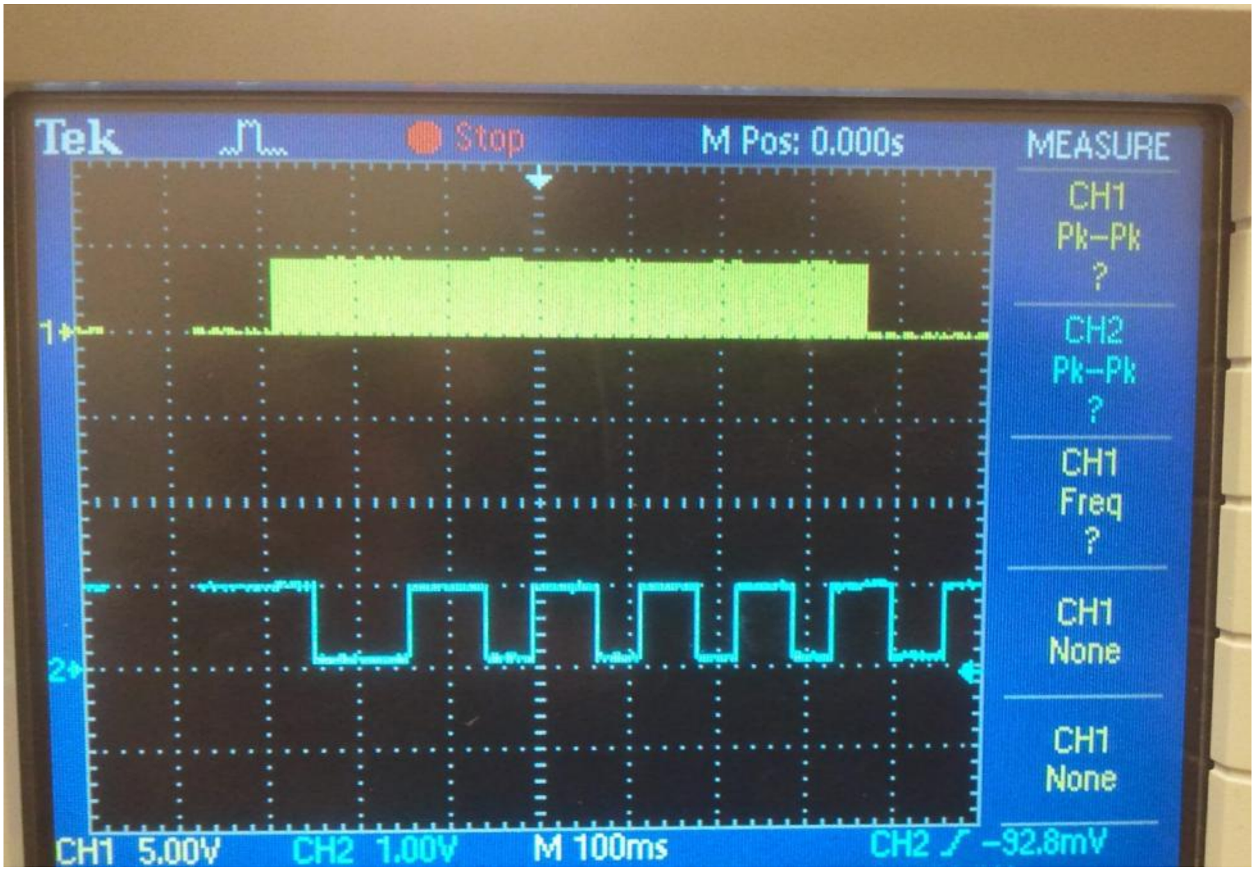
In order to mitigate the contact bouncing, the use of logic gates to smooth out the signal was used. This solution was discovered by one of the student groups in the undergraduate course (it is a standard hardware solution for PC keyboard debouncing). With a combination of NAND gates, the signal input to the interrupt line will maintain a high state or a low state regardless of any bouncing in the signal. As for the noisy signal, a software solution was also discovered by another student group; it was implemented to ignore incorrect interrupts. Using Arduino code, whenever an interrupt occurs, the program was designed to have a software delay loop that ignored any other interrupt for a certain fixed interval (this is also a standard, albeit a software, solution in keyboard debouncing). The length of the interval was determined by the speed of the platform. If the platform was to achieve 15 interrupts in one second, then each interrupt should occur every 67 milliseconds. Therefore, a good software delay to ignore spurious pulses would be 60 milliseconds. If the speed of the platform were to double however, the interval will have to be less than 34 milliseconds to avoid ignoring actual interrupts as well. So, there was a tradeoff that was difficult to resolve.

We have solved this problem with yet another, time-honored solution of shielded cables, as described below under ‘Results.’ Our current implementation does provide the correct number of pulses. More work may be needed to integrate all of the above solutions in a holistic manner.

PID (Proportional-Integral-Derivative) Algorithm [12]: The next part of the project was to utilize control algorithms in order to lower the degree of error in precision as well as to allow for more robustness in the demonstration of mathematical concepts. PID (Proportional-Integral-Derivative) algorithms are feedback mechanisms widely used in industrial control systems. The platform uses a simple process of tuning the parameters of the algorithm to produce a desired output. What the algorithm does is measure the error value between the recorded output and the desired output (called the set point) and attempt to minimize the error using a combination algorithm that uses a weighted sum of the error, its derivative, and its integral, as defined over certain interval. Along with PID, we also had to incorporate robust control loops essentially to control for unknown situations such as the platform heading in the wrong direction or being disturbed.

These algorithms were used to allow the platform to move precise distances and angles. The first step to do was to measure the platform’s speed and angle precisely using the optical encoders (as detailed below). Next was to calculate the error and implement the PID algorithm using Arduino code to interpolate and bring the measured speed and angle to the desired values. For example, if the set point was 15 interrupts or 1 foot, the algorithm should adjust the speed of the platform to reduce the error below a given threshold; the error in this case is the difference between the set point and the optical encoder output (which equals the number of interrupts that have occurred at Arduino), which in turn corresponds to the distance traveled or angle turned. The algorithm will increase/decrease the speed so that the vehicle stops exactly when the 15th interrupt occurs. Once that step is completed, generic algorithms can be used with the optical encoders to account for unknown situations that might occur while the PID algorithms maintains speed and angle when driving and turning.

Further Improvements: As described above, the use of logic gates to smooth out the signal decreased the amount of interrupts from 500-600 to 13-20 interrupts. This resulted in about a 97% increase in accuracy but still remained imprecise in its measurements. With the implementation of a software delay loop to limit the number of interrupts, the Arduino displayed 15-17 interrupts resulting in a further increase in accuracy. While this was definitely an improvement, the number of interrupts displayed for each second the platform is on is not precise enough to reliably travel a set distance and angle. The reason for this imprecision is illustrated in the following image:



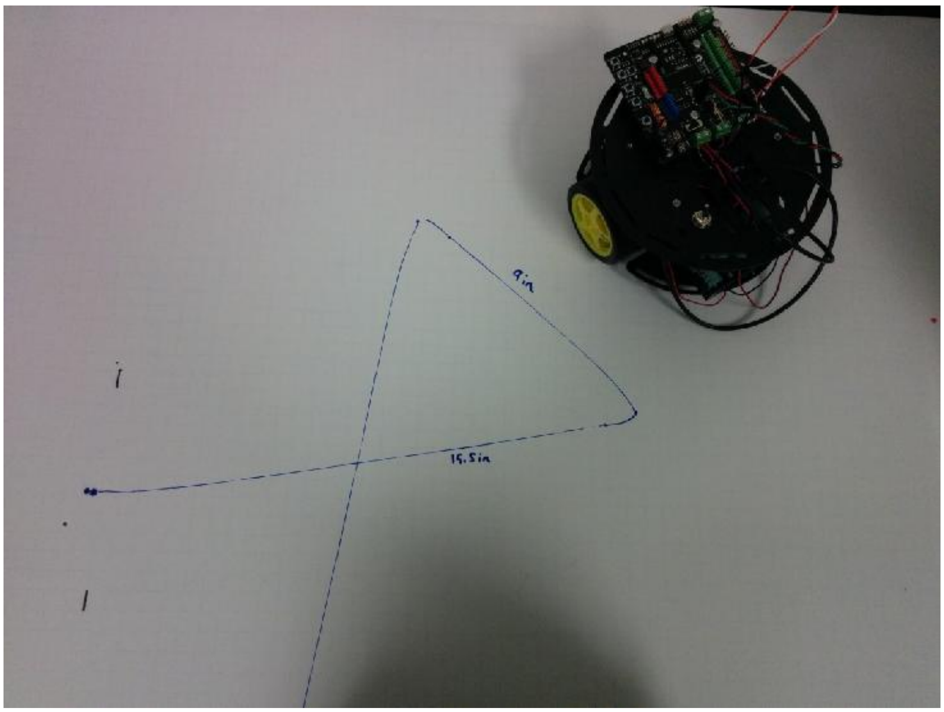
Using an oscilloscope, a comparison was made between the voltage supplied to the motors of the platform and the signal generated by one of the encoders (the upper and lower plots, respectively). At the start of the program, the motors were turned on by the Arduino microcontroller and remained on until the number of interrupts that had occurred reached 6 (count the number of high states). However, one can see a seventh high state in the optical encoder output. Thus, the platform continued to move and did not fully stop once the motors were turned off. This meant that the platform traveled further than desired. This required yet another solution to improve precision in distance and angle measurements.

Implementing PID algorithm on top of an unreliable 'sensor' further exacerbates the situation by varying the speed of the platform. Therefore, a better sensing solution was sought to reliably increase the precision of the robotic platform. To further understand the issue with the encoder readings, an oscilloscope was used to analyze the signal coming from the encoders while the motors were running. The signal produced by the oscilloscope measured a very clean signal from the encoders. This led to the belief that the issue was not related to the encoders. This prompted another look at the motors as the source of the issue.

After testing the motors for their effect on the encoder readings, it was discovered that the noise from the motors was being coupled into the encoder wires to the interrupt pins on the Arduino.

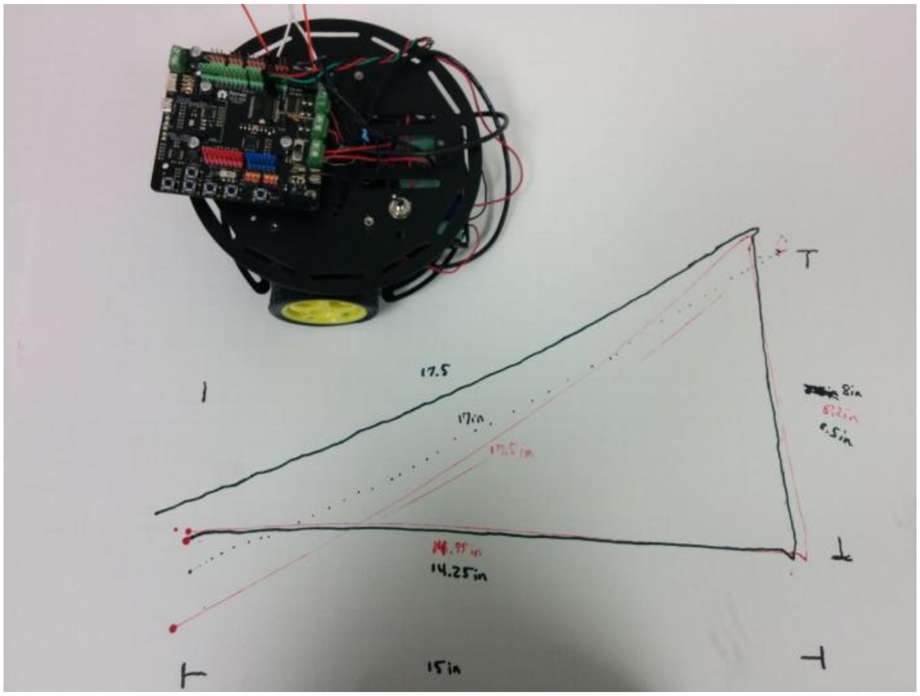
To eliminate this motor noise, non-polarized capacitors ranging from 1 to 50 micro Farads were wired in parallel with the motor inputs. This had a small but noticeable improvement in the encoder readings. Shielded cables were then used and the noise reduction now was much more significant. The encoder readings for the interrupts had no apparent errors, even without the software solutions previously implemented. This new found solution (with no software delay, but with shielded cables and NAND-based debouncing) allowed the PID and angle calculations to be much more reliable. We could now set aside more time between interrupts for PID and other relevant computations.

Results:



We have successfully incorporated the PID algorithm. The effectiveness of PID was demonstrated by programming the platform to draw right triangles on a 6 ft by 6 ft canvas using a mounted marker (pen at the center of the platform). It was programmed to travel 15", 8", and 17" to draw the sides and perform 90° and 60° left turns. The first triangle (shown above) was drawn with PID control disabled resulting in large errors when turning a particular angle as well as driving forward.

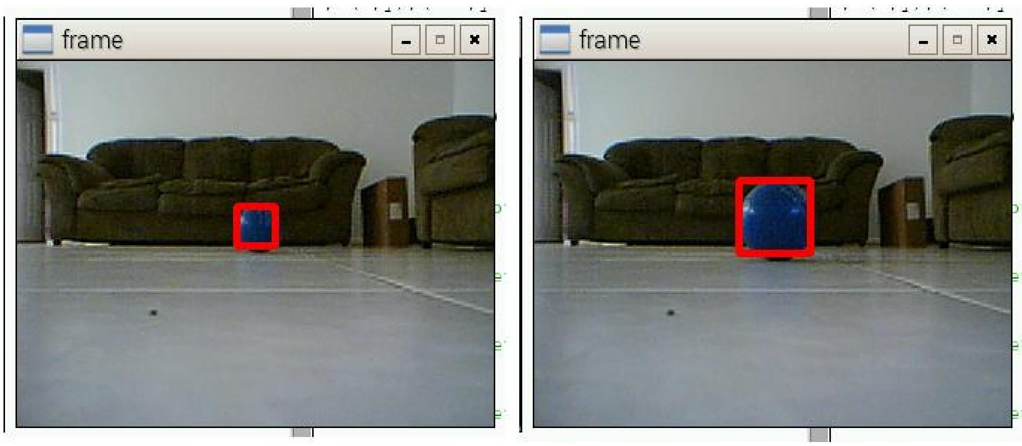
After enabling the PID controller, the platform was able to draw the same triangle significantly better (see below). The red and green markings represent the precision of the encoder readings when performing a turn. Further work is underway to improve the precision. This appears related to battery power drainage and consequent impact on the motor's functionality.



Additional Elements to Enhance the Platform's Usefulness and Accuracy: During the current semester, a group of ten engineering students are working on engineering issues related to real world robotics: a dashboard for robot's programmed control and to monitor its movement; and significant enhancement of accuracy and repeatability. These are detailed here in the Results section, though at present this work is not complete; we included it here to separate the engineering and math aspects to the extent possible.

A Dashboard for Controlling and Monitoring the Robot: While the platform is in motion, relevant data such as the distance, angle, and speed of the platform will need to be wirelessly transmitted to the PC for students to get a better sense of how the platform works. This can be done using Bluetooth communication between the Arduino and a Bluetooth-enabled device such as a Tablet or Laptop. A Bluetooth module has been integrated into the platform and it receives data from, and transmits data to, a Windows Tablet or PC. Using Python, a graphical user interface (GUI) has been developed to send a sequence of commands to the robot so it can execute a math problem's solution in an autonomous manner. Another GUI will be developed to display robot's transmitted data in a simple and intuitive way. It will also be able to display other information such as error values and the current processes the platform is undergoing. These results can be used to help students see how the platform is applying mathematical concepts for a more in-depth view of the robotic platform's actions. This will help the students to develop more of a connection to the concepts that they are learning. This will be demonstrated at the conference and will be documented at our website [4].

Error Detection and Correction (EDC) Scheme: For this, we use four Raspberry Pi miniature computer platforms[15], priced at $35 each, in four corners of a typical robotic floor space of about 15’ x 15’. These are fixed in the room space at an elevation using camera stands and thus can be used to track the motion of a robot on the floor. We use inexpensive USB-based cameras (at a cost of $10 each) with these Raspberry Pis to capture the images of one or more robots on the robotic floor and use that information to determine the error in their positions and use that to guide them to the correct location. We estimate the error in our current platform to be less than ± 10% when only the PID algorithm is implemented; with this additional error detection and correction (EDC) scheme, the error is expected to be reduced to ± 1%.



The EDC scheme has multiple elements that work together in synergy: the four Pi platforms procure their perspective of the robot’s position and then relay the same to a central PC, which acts to poll these four platforms and obtain distance and angle as obtained from each of these platforms. The server then uses triangulation[16] to compute the position of the robot with respect to the origin, typically the Pi platform at the bottom left corner. Once its position is known precisely, the server can send commands autonomously to the two appropriate Pi platforms to guide the robot to the position that is accurate by math standards.

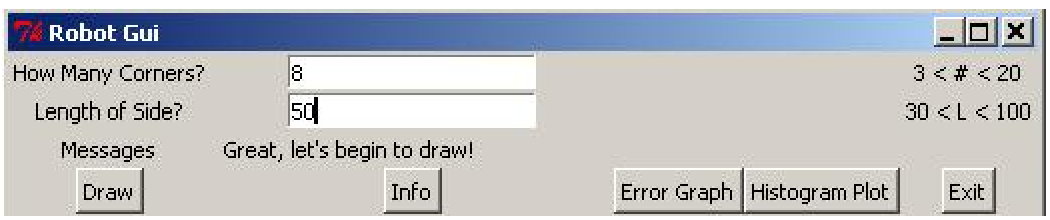
The distance and angle estimation algorithm, implemented at each Raspberry Pi station, uses a color identification method [17]. In our implementation, a blue-colored-object is attached on the top of the robot, and an USB-based PlayStation 3 eye camera (cost: $10) is used. It has a better resolution and depth of field as compared to the original Raspberry Pi’s camera. Each camera captures a real time video stream from which the object is identified by setting upper and lower limits to the blue color that we want to isolate and identify in our captured frame. The implemented Raspberry Pi code isolates the object, contours the object, blurs everything else and draws a red square on our desired object (see above). The square that is drawn is utilized to estimate the distance from the camera to the object. Programmatically, this edge detection algorithm is implemented using the Python OpenCV[18] library.

There are other engineering details pertinent to angle measurement, multi-threaded communication infrastructure, and algorithm to compute the x and y coordinates with respect to the origin. All these will be presented briefly at the conference.

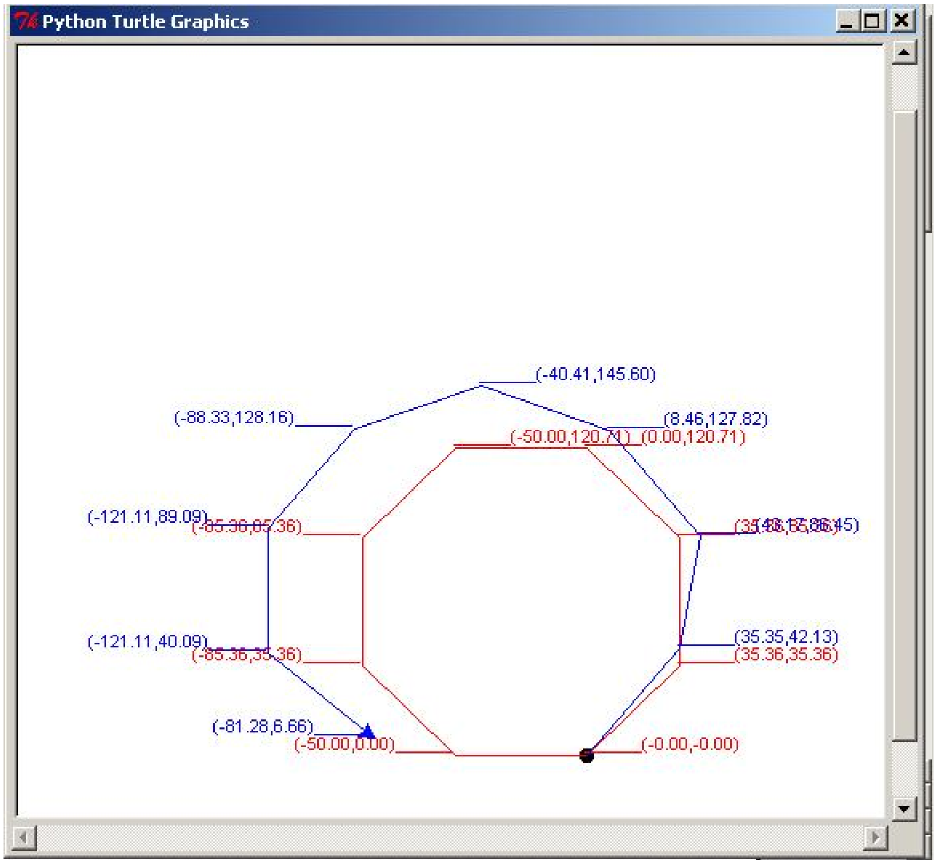
Note that the same four stationary Raspberry Pi platforms (along with their associated cameras), at a total cost of $45 each, are useful whether we have one or several robots on the floor at any given point, thus allowing many student teams to share the same background infrastructure for their math lessons and beyond.

Showcasing the Real-World Error: At a different level of communicating the mismatch between the precise math and the 'raw' imprecise engineering implementation, we have set up computer simulations to show case the effect of a $\pm$ 10% random variation in a robot's motion, as would be the case if only PID algorithm was implemented.

A Graphical User Interface (GUI) helps to achieve a clear and simplified discourse of this. We use familiar "Windows" data entry boxes and click buttons. This was coded using Python 2.7. This can be displayed at the main PC which acts also as a server.



The GUI shown above is used to demonstrate how the platform may move with & without the error detection and correction (EDC) scheme, by demonstrating both the corrected & uncorrected movement path (in red and blue, respectively), as shown below. We only show one group's work here; however, we have 5 creative solutions for the GUI and the error display, from the 5 class groups. Better ideas will be distilled and incorporated in our final flow. A standardized GUI interface has been developed and implemented. It will be presented at the conference.



Discussion:
This section details our earlier and current work, all focused on adapting our work to teach math in middle and high schools and research associated with it. It will end with a brief outline of our plans for integrating it all in the near future. An engineering section on how to further enhance the accuracy of the robot using a network infrastructure is included above under 'Results' though it more appropriately belongs here. We did that to separate engineering and math aspects of the project.

Earlier Research Results from Student Interviews: We have conducted two case studies that show that some students are interested in the application of mathematics to real world problems. Following the spirit of the Common Core Curriculum [19, 20], we emphasize "solving real world and mathematical problems." In the classroom, many students do not see the point of mathematical problems. The advantage of robotics is that these problems can be acted out visually, and controlled in a hands-on manner. Furthermore, we will examine the effect of real world error. We are making the problem more accessible to students, but not by over-simplification. In fact, we are asking students to explore the problem in greater depth. There was some initial confusion, but students found the problems interesting, and they came up with solutions that showed a deep understanding of the mathematical and engineering issues.

In one study[2], we investigated how undergraduate engineering students constructed the robots. One student provided insight into his own problem solving process. It is one thing to use the principles of trigonometry in a class when there are hints that the principles are needed. In this example, one student recognized that something was needed, and then realized he was already familiar with it. What makes this an interesting case, from the perspective of mathematics education, is that the student took time to describe the mathematical knowledge that he already knew, and then showed how he applied it to the particular real-world problems.

In a second study[3], the robots developed by undergraduate engineering majors were then used in a semester long elective course was offered to 9th grade students. Under the school-wide STEM initiative, the goal of the course was to enhance students' exposure to, and comfort level with STEM disciplines so they can migrate towards a curriculum that allows them to pursue their passions while taking advantage of STEM advances and opportunities. Seventeen ninth grade students in groups of three (two groups were larger at 4 members each) assembled and programmed robots to draw geometric art on a large canvas. The use of open source software and hardware helped high school students focus on the experience, rather than dealing with lower level engineering details. At the end of the course, groups presented their geometric art in class.

Two students who participated in the robotics course volunteered for an interview. Both students interpreted the question as a mathematical problem, which required the Pythagorean Theorem. In each case, the student gave the correct solution. When discussing a robotic simulation of that problem, they all recognized that if the robot odometer error less than 10%, this could be explained by real world error. However, if the odometer error was substantially greater than 50%, this could not be explained simply by real world error. Students were readily able to understand the difference between conceptual error and real world error. They knew how to solve the mathematical problems, and how to examine the simulations of the robots. They were able to coordinate both types of knowledge into a meaningful pattern (of relating exact solutions from Math and errors prevalent in real world situations).

The students were also able, with prompts, to realize that their work could be used as an instructional tool to a wider range of students. They became aware that they had formed a group where there was much tacit knowledge – shared by the group, but not common to the average high school student. There were ways to communicate this to a broader audience and they became interested in doing so.

Note that these students were pre-engineering students and could internalize the differences between mathematical solutions and real world solutions. As explained earlier, our new platform has been made precise, and work is underway to make it accurate. With these changes, we believe the platform will be able to address the needs of a typical math student, of providing visualization and physical proof of mathematical concepts covered in the class, to a high degree of accuracy. The following work discusses our efforts in developing robotic interpretations for common math problems.

Robot-Based Math Lessons (Under Development): At present, another engineering undergraduate student (the fourth author) is working on developing robot-based math lessons for a set of pre-calculus problems. These are from a well-known Mathematics book[14] as well as from some lessons that cover some of the standards for high school outlined in the Common Core State Standards for Mathematics. These standards are learning goals that outline what a student should know and be able to do at the end of each grade[19, 20]. This work is in collaboration with a high school Math teacher teaching in the Broward County School District. Below is a table of eight topics under development that involve the use of the robot. Column one covers the general topic that question aims to address. Column two describes the possible implementation and application the question would be applied to. Finally, column three describes any additional ideas the question might cover that would help students solve the question or extends their understanding further. There are more topics planned, but for right now these are the three topic ideas under exploration.

Topic	Question / Application	Additional Ideas
Trigonometric Ratios	Surveying used in Civil Engineering to find height from the angle of elevation and the horizontal distance from the base	Scaling (robot implements a scaled version of the surveyed building example), optimization
Unit Circle	Visual proof that $x2+y^2 = 1$ holds true for all (x, y) on the unit circle; Confirm cosine/sine values of the angle at the center.	Draw a polygon with multiple sides and show how it morphs to a circle as the segment length reduces and the number of sides increases. Convergence of perimeter to circumference of 2π
Pythagorean Theorem	Draw square areas around the base, height, and hypotenuse of a right-angled triangle.	Scaling as above; change the angle from right angle. Visualization of an irrational

	Count the squares to prove the theorem.	number
Rates	Conceptual understanding of speed and the relationship between distance and time.	Constraints, trade-offs, and side effects
Angles and Quadrants	Obstacle course the students need to go through and avoid obstacles.	Constraints, and Algorithms; Give students a treasure map where they are given an algorithm and they need to figure out which treasure theirs point to
Geometric Shapes	Model it – teacher gives students task to model an object out of simple shapes with certain constraints : the base angles must be congruent, etc.	Constraints
Coordinates	Given certain points, prove/ disprove shape/properties of the shape the points make.	Students can use the coordinates to compute perimeters of polygons
Functions	Transform a function that helps the robot reach a goal/avoid an obstacle	

To get into a little more detail, the 'Trigonometric Ratios' topic revolves around the idea of utilizing the trigonometric ratios to find the distance/height/length of an object given either the angle and the length of one side or given the length of two sides. The student will be able to double check his/her answer using the robot, by having it draw the right angled triangle using the given values (scaling down the side lengths by a common multiple if it's too large to be drawn on paper). Then the student can measure the drawn triangle with a ruler/protractor, scale up the results (if needed), and compare the results for accuracy. This will be supplemented soon with a GUI (graphical user interface), as discussed under the 'Results' section above.

Another variation of this would be to turn this question into an optimization problem by adding certain constraints and having the students figure out how to optimize the system. An example question which is inspired from a question in another pre-calculus book [21] would be the following: There is a house with a 2 meters tall fence 3 meters from the house and you want to have a ladder that, from the ground, stretches over the fence to the house. What would be the shortest ladder that would reach? The student would be encouraged to use the robot to help model the situation. At any time, the student can stop and algebraically solve the question.

The ‘Unit Circle’ topic acts as more of a visual proof that utilizes the Pythagorean Theorem to help link the idea of the lengths of the right triangle's legs being equal to the cosine/sine value of the angle. The current robot can be used to draw a polygon with short segments that will mimic a circle; but we expect to have an algorithm for drawing a more realistic circle by this summer. The circle so constructed will allow students to double check the coordinate values of well-known angles on the unit circle by having the robot draw a line with a given angle away from the origin. The application that would utilize this idea is still being debated, but the question should be aimed at making the idea of the Unit Circle real to the student.

The ‘Pythagorean Theorem’ can be visualized by drawing squares around the base, height, and hypotenuse of a right-angled triangle. A good reference from the University of Nottingham and the University Of California, Berkeley[22] provides guidance on how to count the number of squares on a graph sheet. The same concept can be translated for use with robotics. The Pythagorean Theorem is considered to be the most challenging concept to comprehend. In this representation of the Pythagorean Theorem, students examine a right angled triangle, for example, with base of length 3 and height of length 4, yielding a hypotenuse of length 5, with the corresponding number of squares at 9, 16, and 25. This is the classical example which yields a rational number for the hypotenuse. We can then extend with another example, with base of length 2 and height of length 3. The hypotenuse is the side of a square with an area of 13. In this case, the area of the square is an integer, even though the hypotenuse itself is an irrational number, the square root of 13. This gives a model that robots can simulate and that students can understand.

The ‘Rates’ lesson covers the idea of rates, specifically that of speed. The question involves using two robots: one robot making a small square path with sides that are 5 units long and the other making a larger square path around it with sides that are 10 units long. Both are going at the same speed. The students will need to figure out, apart from changing the size of either of the square paths, how to get both robots to finish tracing their squares at the same time. After the students present their answer, there could be a discussion about each group's solution, the trade-offs/side effects, and which solution falls in line with the constraints the best with minimal negative trade-offs/side effects. This question encourages the student to assess the situation, collaborate with team members, and think of ways they can modify the system while still conforming to the requirements. This lesson can be done with teams competing against one another or as a class effort to take everyone's ideas and use the better ones.

The question revolving around the idea of 'Angles and Quadrants' involves the students using the robot to navigate an obstacle course. The students will be presented with an area containing a couple of obstacles, a starting line, and a finish line. The students would be encouraged to survey the course and come up with their own algorithm of commands (consisting of a sequence of angles and distances) that would safely bring the robot to the end. To make this exercise more fun, the course could have 'bonus' items scattered around that , when hit, would give the students a certain number of points. The students would then have to gather the most number of points with the least amount of the traveled distance. A variation of this would be giving the students a treasure map showing an algorithm that would take the student's robot to a certain item in the room. The way the algorithm in the map is presented can be cryptic: showing angles in radians, reinforcing the concepts of the quadrant, showing the distance as the solution to an equation, etc. Again, this lesson can be done with teams competing against one another or as a class effort to take everyone's ideas and use the better ones.

For the 'Geometric Shapes' topic, students will be given the task to model an object out of simple shapes with certain constraints, using the robot. The teacher could show the class an actual picture of the object that needs to be modeled (say a seesaw) and say that the modeled seesaw design must use a regular polygon with an interior angle of 60° and two parallel lines. The students would then be given the robot to draw the requested shape. This question was made to address math common core standard [23] which states to "Use geometric shapes, their measures, and their properties to describe objects." This is done so the students can get used to seeing geometry in their everyday lives.

The 'Coordinates' idea involves the students using the robot to build a shape through using given coordinate points. An example question could be like this one from a website that follows the Common Core Standard[24]: "Prove that the shape with corners A (1, 4), B (3, 0), C (1, −4) and D (−1, 0) is a rhombus." The students can give the coordinates to the robot to draw the shape and help themselves visualize the object and assess its properties before using the distance formula to prove it.

Finally, for 'Functions', the students are given a robot that has a function which it uses to dictate its path. The only issue is that the function makes the robot run into obstacles or avoid the goal. The student is given the function that the robot is using and is told to transform it in such a way that the robot does what is desired. The students are encouraged to modify the function and try out their modifications to see if it fixes the robot's path. Functions to be transformed can range from simple linear equations to more complex trigonometric functions. For example, if the robot is following a linear function and it keeps running into an object, that is the y-intercept of the function, the student must modify the y-intercept to keep the robot safe from the obstacle.

Another problem, suggested by math teachers, is the FOIL method for multiplying two binomials. We will elaborate this in the conference presentation.

Besides the math lessons discussed above, we have created a video to show students how to build their own robot [25]. The video goes through the process of showing the components they need, the parts they should have with their kit, and step by step details on how to put the components together to have a functional robot. The video also has background music added to it that makes it enjoyable to watch.

Our immediate future plans: This briefly documents our future plans with regard to our efforts to integrate robots in math education. Our preliminary finding indicates that teachers are willing to substitute one to two weeks of math education with robotic math. We will start with these few examples at the level of 8th or 9th grade and expand from there. This will help us in two ways: practical issues that we might face; and an opportunity to conduct a pre and post survey, to find the level of interest and impact among students. If, as expected, we find that it is a robust platform and there is a strong interest, we are then able to present the results to teachers in other grades and recruit them for the development of further lessons. This is an iterative process that is bottom-up - since we know that math teachers, like other teachers, are burdened with a strict regimen of material to cover and a substitution must help students in some significant manner, hopefully in this case with better class interaction to understand some fundamental concepts and with some fun at the same time via social and team interactions. The math professor in this group is offering a course this summer to Math teachers; we expect to recruit a few teachers from the class to work with us in find further ways to incorporate robots in math. No major case-control study can be planned until a lesson plan is developed and a group of willing teachers are identified to implement the same in their class curriculum. This is expected to follow after about one more year of due diligence on our part.

Conclusion

With the completion of this project, the robotic platform will be an affordable choice for high schools to reinforce their students' math. The platform is expected to be able to reliably move and turn so that the demonstrations will be more precise (repeatable) and accurate. The schools will also have more of an incentive to teach programming and engineering concepts to students as well as more advanced concepts of math due to higher retention of material. The students themselves will attain a deeper understanding of the mathematical concepts with the utilization of the platform. They will begin to see how math is applied in programming and engineering more thoroughly when utilizing the robotic platform. They will also develop more of an interest in STEM-related fields with their new-found problem-solving skills.

Although many problems in mathematics classes have exact solutions, simulations in the real world involve real world error. Our case studies suggest that some students readily understand this and can accurately interpret such errors. However, in bringing the robots into the classroom, these errors must be minimized so that teachers can help a wide variety of students. Our approach is to reduce these errors (using methods described). We also will show the exact mathematical answer (corresponding to classroom instruction) and compare to the robotic simulation. Students making the robots, as well as those helping to prepare the robots for the classroom, understand the connections between exact answer and real world simulation. Planned work will make the robots more precise and accurate, and will also show the connection between mathematical answer and the simulation so that a wide range of students understand the math (at earlier stages) as well as learning about engineering (at later stages).

Acknowledgements:
We are grateful to the faculty members from other disciplines, viz., Math, Education, and K-12 for their incisive comments and feedback. It is anticipated that by this summer, we would have developed a reliable and useful robotic platform along with a set of Math lessons. The magnet program coordinator for our school county has agreed to do a pilot run with her after-school program this summer. It will also provide research material for her PhD in Education.

References:

1. Shankar, R., Ploger, D., Nemeth, A., and Hecht, S.A. (2013). Enhancing Pre-College Mathematics Learning with Real-World Examples. Mathematics Division, Technical Session I, Paper ID 7891, *120th ASEE Annual Conference and Exposition*, Atlanta, GA, June 2013.
2. Ploger, D., Shankar, R., Nemeth, A., and Hecht, S.A. (2012). Exporting Engineering Technology Practice to Enhance Pre-College Mathematics Learning, pp. 236-245, *2012 ASEE Gulf Southwest Annual Conference*, El Paso, TX, April 2012
3. Ploger, D., Shankar, R., Nemeth, A., and Hecht, S.A. (2013). Engineers and Other People: Communicating Engineering Results to a Larger Audience. Session ETD-435, *Conference for Industry and Education Collaboration* (CIEC), Phoenix, AZ, 2013
4. FAU's Robotics site (2015). http://robotics.fau.edu/ -- the site has individual website links to the work of engineering and high school students.
5. FAU High School Student Demo Videos (2011). http://faculty.eng.fau.edu/shankar/research/robotics/
6. Barker, B. S., & Ansorge, J. (2007). Robotics as means to increase achievement scores in an informal learning environment. *Journal of Research on Technology in Education*, *39*(3), 229-243.
7. NSF (2000). The Interplay between Mathematics and Robotics: Summary of a Workshop. *National Science Foundation*: Arlington: VA
8. Holmquist, S. (2014). A multi-case study of student interactions with educational robots and impact on Science, Technology, Engineering, and Math (STEM) learning and attitudes. *PhD Dissertation*, University of South Florida, Tampa, FL.
9. Brand, B., Collver, M., & Kasarda, M. (2008). Motivating students with robotics. *SCIENCE TEACHER-WASHINGTON-*, *75*(4), 44.
10. CMU Robot Algebra (2014).CMU Robotics Academy: Robot Algebra. http://www.education.rec.ri.cmu.edu/content/educators/research/robot_algebra/ accessed on 2/2/2015
11. Le Mellec, J., Math with EV3 (2014). http://smartsystems.eng.fau.edu/math-with-ev3/ .Work of a visiting graduate student at FAU
12. PID Controller. http://en.wikipedia.org/wiki/PID_controller , accessed on 3/14/15
13. Arduino, an open source embedded system platform. http://www.arduino.cc/ , accessed on 2/2/15
14. Sobel, M.A. and Lerner, N. (1991). *Precalculus Mathematics*, 4th edition, Upper Saddle River, New Jersey: Prentice Hall College Div.
15. Raspberry Pi, a low cost microcontroller. http://www.raspberrypi.org/ , accessed on 2/2/15
16. Triangulation, http://en.wikipedia.org/wiki/Triangulation , accessed 3/15/15
17. Criminis, A., Single-View Metrology: Algorithms and Applications, http://research.microsoft.com/pubs/67279/criminisi_dagm2002.pdf , accessed 3/15/15
18. OpenCV, Computer Vision Algorithms, http://docs.opencv.org/modules/core/doc/intro.html , accessed 4/6//15
19. CCSS for Mathematics (2010): Common Core State Standards for Mathematics. Retrieved on March 14, 2015, from http://www.corestandards.org/assets/CCSSI_Math%20Standards.pdf
20. About the Standards. (n.d.). Retrieved April 6, 2015, from http://www.corestandards.org/about-the-standards/

21. Larson, R., & Hostetler, R. (2006). Applications and Models. In *Algebra and Ttrigonometry* (7th ed.). Boston: Houghton Mifflin.
22. Mathematics Assessment Resource Service of the University of Nottingham & UC Berkeley (2012). The Pythagorean Theorem: Square Areas. Retrieved on March 15, 2015, from: http://map.mathshell.org/materials/download.php?fileid=1206
23. High School: Geometry » Modeling with Geometry. (n.d.). Retrieved April 6, 2015, from http://www.corestandards.org/Math/Content/HSG/MG/
24. Shmoop Editorial Team. (2008, November 11). Common Core Standards: CCSS.Math.Content.HSG-GPE.B.4. Retrieved April 6, 2015, from http://media1.shmoop.com/worksheets/geometry/parallel-perpendicular-lines/g-gpe-worksheet_4_ans.pdf
25. Augustin, M., Build your own FAU Robot, https://vimeo.com/120392503