Creating Research Opportunities with Robotics across the Undergraduate STEM Curricula

Dr. Janusz Zalewski, Florida Gulf Coast University

Janusz Zalewski, Ph.D., is a professor of computer science and software engineering at Florida Gulf Coast University. Prior to an academic appointment, he worked for various nuclear research institutions, including the Data Acquisition Group of Superconducting Super Collider and Computer Safety and Reliability Center at Lawrence Livermore National Laboratory. He also worked on projects and consulted for a number of private companies, including Lockheed Martin, Harris, and Boeing. Zalewski served as a chairman of the International Federation for Information Processing Working Group 5.4 on Industrial Software Quality, and of an International Federation of Automatic Control Technical Committee on Safety of Computer Control Systems.His major research interests include safety related, real-time embedded and cyberphysical computer systems, and computing education.

Dr. Fernando Garcia Gonzalez, Florida Golf Coast University

Dr. Fernando Gonzalez joined FGCU as an Assistant Professor in the Computer Engineering Program in the fall of 2013. Previously he was an Assistant Professor within the Engineering, Math, and Physics Department at Texas A&M International University in Laredo, Texas. Prior to that he was a Technical Staff Member (researcher) for the U.S. Department of Energy at Los Alamos National Laboratory in Los Alamos, New Mexico. Dr. Gonzalez was also a faculty member in the Electrical and Computer Engineering Department of the University of Central Florida. Dr. Gonzalez graduated from the University of Illinois in 1997 with a Ph.D. in Electrical Engineering. He received his Master's degree in Electrical Engineering and his Bachelor's degree in Computer Science from Florida International University in 1992 and 1989. Dr. Gonzalez research interest includes the intelligent control of large scale autonomous systems, autonomous vehicles, discrete-event modeling and simulation and human signature verification.

Creating Research Opportunities with Robotics Across the Undergraduate STEM Curricula

Abstract

There is a recognized nationwide need to educate next generations of students in Science, Technology, Engineering and Math (STEM) disciplines. In response to this challenge, the authors' institution, College of Engineering, has developed over recent years a sophisticated undergraduate software engineering and robotics laboratory for use in embedded systems and related project courses. As a result, a number of teaching modules have been put in place, with emphasis on complex systems and web-based access. These modules can be used to prepare undergrads for developing robotics applications usable across the undergraduate STEM curricula to encourage students of various disciplines to do related research. The projects have a potential impact on advancing teaching in STEM disciplines by connecting the lab and research to multiple STEM related courses. The paper presents a methodology for using such approach across the STEM curricula and discusses respective applications related to robotics.

Introduction

There is a recognized nationwide need in the United States to educate next generations of students in Science, Technology, Engineering and Mathematics (STEM) disciplines. It is widely believed that STEM focused education contributes to the innovativeness in product development and as such has a significant impact on strengthening the economy and making it more competitive globally. Respective findings have been published on multiple occasions, by various bodies, most notably in a report by the National Science Board, which states, among other things.[1]

> *In the 21st century, scientific and technological innovations have become increasingly important as we face the benefits and challenges of both globalization and a knowledge-based economy. To succeed in this new information-based and highly technological society, all students need to develop their capabilities in STEM to levels much beyond what was considered acceptable in the past. A particular need exists for an increased emphasis on technology and engineering at all levels in our Nation's education system.*

On the other hand, the report of Computing Community Consortium[2] identifies robotics as a "key economic enabler", considering that robotics technology holds the potential to transform the future of the country, and stating among other things that:

> *Unfortunately, the United States lags behind other countries in recognizing the importance of robotics technology. While the European Union, Japan, Korea, and the rest of the world have made significant R&D investments in robotics technology, the U.S. investment, outside unmanned systems for defense purposes,*

remains practically non-existent. Unless this situation can be addressed in the near future, the United States runs the risk of abdicating our ability to globally compete in these emerging markets and putting the nation at risk of having to rely on the rest of the world to provide a critical technology that our population will become increasingly dependent upon. Robotech clearly represents one of the few technologies capable in the near term of building new companies and creating new jobs and in the long run of addressing an issue of critical national importance.

In addressing these challenges, the authors institution's software engineering department has developed over recent years a sophisticated undergraduate software engineering lab for use in embedded and cyberphysical systems and related project courses.[3] As a result, a number of teaching modules have been put in place, with emphasis on developing complex systems, and providing web-based access to the lab stations.

One of the most important foci of the lab is robotics, in addition to other specific aspects of modern computing, such as: wireless sensor networks, game development for mobile devices, embedded microcontrollers, distributed control systems, etc. The projects concentrating on robotic devices can use them as a vehicle to convey STEM related knowledge to undergraduate students across the curriculum. Given the attractiveness of autonomous or manually controlled robots, it is assumed that the student population across various STEM related courses, in respective disciplines, will respond positively to acquiring knowledge associated with applying the robots in their respective subjects of study.

In this view, the objective of this project is to develop robotics applications that can be used across the undergraduate STEM curricula to encourage students of various STEM majors to enhance knowledge in their disciplines, study topics related to robotics and ultimately initiate related research.

The rest of the paper is organized as follows. First, we discuss briefly previous work on robotics for STEM and robotics in software engineering education, then present a general methodology for integrating STEM concepts across curricula. Next, the robotics equipment on hand is described, followed by the description of applications. The paper ends with conclusion and some lessons learned from the project thus far.

Previous Work on Robotics in Teaching STEM and Software Engineering

The topic of using robotics in teaching STEM courses is not particularly new. Pioneering work of Seymour Papert has been credited with introducing the idea of using robotic devices in teaching.[4] For example, Barker and Ansorge refer to Papert's concepts of teaching scientific and mathematical principles through experimentation with robots,[5] when they talk about increasing achievement scores in informal learning. More recently Berland et al. trace the Lego Mindstorms robotic kit back to his original vision of educational robotics.[6]

Several other researchers have reported on using robotics in STEM education,[7-8] but nearly all their narrative is related to using LEGO robots exclusively in K-12 education, perhaps, with a

few exceptions, such as Mataric et al,[9] who refer to the possibility of using robotics in STEM courses at the college level. However, there are no substantial reports on using robotics across the STEM curricula in colleges or universities. There have been only isolated attempts to use robotics in individual STEM disciplines, of which Software Engineering is one example.

In this regard, a couple of publications are worth mentioning. Roy et al. reported on their experiences of teaching software development in a robotics project course.[10] They described a curriculum for a two-semester course sequence in hardware-software development, involving a full development cycle for an autonomous mobile robot. They share their insights on teaching large-scale system development, with emphasis on software-intensive projects.

Christensen developed and taught a course on Software Engineering in Robotics,[11] with a basic objective to introduce students to the fundamental issues of software engineering and how it applies to robotics. The software engineering part of the course covers the design, analysis and synthesis of systems for real-world operation, with presentation of some basic methods of software engineering, their application in robotics and the use of state-of-the-art software systems to implement these methods.

As opposed to the use of software engineering principles in robotics courses, described above, Göbel et al. reported on using robotics in the software engineering program.[12] Their initial idea of using rather simplistic Lego Mindstorms robots had soon to be expanded to a simulation environment, since the student population got more interested in the subject matter and more sophisticated, requiring full-scale development and testing. The major observation from this project is that keeping focus on software, due to simulations, will let the students learn the algorithmic part of robotics, as opposed to dealing with the robot mechanics and electronics. The same observation is independently confirmed by others,[13] as well has been made by the current authors.[14-15]

Overall, from the literature review of the subject it is clear that using robotics in STEM and related courses is very attractive to students and instructors, because it facilitates the transfer of knowledge and significantly assists in acquiring various skills by students. However, all reported works either focus exclusively on LEGO robots in K-12 education, or touch upon a single STEM area at the college level, and stop short of using more sophisticated robotic devices and applying them across the STEM curriculum. This situation gives the motivation for current work, which is presented in the next sections.

General Methodology: Integrated Approach to Teaching STEM

Integrating Activites in Various STEM Disciplines

Traditionally, STEM courses, whether it is science (biology, chemistry, physics, etc.), technology (information technology, educational technology), engineering (chemical, computer, electrical, environmental, mechanical, etc.), or mathematics (any branch), have been taught individually, without much interaction among them. This is typically called a “silo approach.”

This project advocates the integrated approach to STEM, actually very rarely seen in current teaching practices at the undergraduate college level.[16] The approach, schematically illustrated in Figure 1, shows material and knowledge from all four disciplines incorporated in each activity across the spectrum of all STEM activities in a course.[17] Math and technology disciplines are the basis of each activity, marked at the bottom, and science and engineering disciplines draw from the support of math and technology, developing respective concepts at the higher levels. In this view, science disciplines essentially rely on inquiry and discovery, while engineering activities apply scientific concepts to construction of respective artifacts.

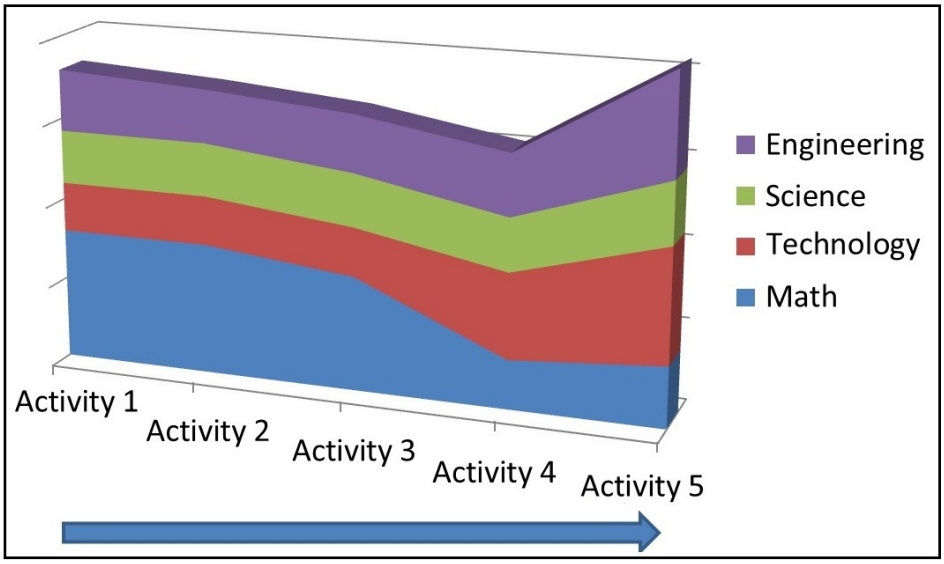

Fig. 1. Integrated approach to STEM[17]

As trivial as this diagram might seem, it implies far reaching consequences, if we clarify it with some practical examples, how specific cross-discipline activities might look like. This is shown in Table 1, for activities related to teaching specific concepts across the STEM curricula.

Table 1. Integrating various activities across STEM disciplines

Activities Related to Specific Concepts			
STEM Discipline	**Quadratic Formula**	**Feedback Control**	**Robot Kinematics**
Science	Application in physics (distance calculation): $s(t) = s_0 + v*t + a*t^2/2$	Application in biology, for example temperature regulation in mammals.	Analogies between human and robotics movements.
(Eductional) Technology	How to use the software solution and for what reasons?	Using feedback theory in Instructional Design.	Use of simulators to teach about kinematics of robotic movement.
(Software) Engineering	Programming solution: r1 = (–b+sqrt(Δ)) / (2*a) r1 = (–b–sqrt(Δ)) / (2*a)	Thermostat as a temperature controller based on feedback.	Software development for robot movement based on kinematics.
Math	Solution theory: $a*x^2 + b*x + c = 0$	Linear feedback theory: gain=(amp/1-feed*amp)	Kinematics equations of robot movement.

For example, a concept as basic as a Quadratic Formula can be spread across all STEM disciplines with different flavors in each, making in fact an integrating case. A bit more involved concept of Feedback Control may form a separate module or a learning object, applicable in a number STEM disciplines. The case of Robot Kinematics is more sophisticated, because at the undergraduate level it involves specific research activities.

Gradual Approach to Introducing Integrated STEM Concepts

In teaching STEM concepts, especially in the integrated manner proposed in this project, it is essential to introduce students to the related material gradually, step by step, since not all of them may have the required background or learning ability to grasp all intricacies of the concept at once. The approach advocated here relies on letting the students develop minimal skills and understanding first, then move gradually to the next, more sophisticated, steps of acquiring respective knowledge, ultimately to be able to conduct supervised research.

This is accomplished by offering students multiple levels of sophistication with respect to a certain concept, including three basic enforcement vehicles: demos, exercises and assignments, to ensure that the learning process and the acquisition of knowledge correspond to each student's respective background. Then, at the higher levels, in the science and engineering parts of the knowledge acquisition, students conduct experiments and develop full-scale projects, which lead to formulating research questions and conducting procedures to obtain answers. This results in extensive reports documenting the research outcomes as well as knowledge acquisition and learning processes.

In summary, respective individual steps of the six-step process of teaching STEM concepts across the curriculum, leading to research activities at the end, can be described as follows, with focus on software development:

- demo – this activity is showing to the student the application of a concept, for example, by demonstrating an operation of a device, without a manual intervention of a student; a specific example would be to use a telescope to show a certain object on the sky
- exercise – this activity is allowing a student not only to witness how a certain concept works, but also to run a device demonstrating this concept, perhaps by changing the device parameters; a specific example would be to let a student set programmatically parameters of a telescope to move automatically through a predefined segment of a sky
- assignment – this activity relies on developing an implementation of the concept, for example, in the computer science context, it would be developing a program to run the telescope, focusing on the program development cycle, with writing (editing) the code, compiling, executing, and debugging it
- experiment – this activity relies on enforcing the essential concept of scientific inquiry, which in the context of computer science would be testing the software developed for a telescope, with outlining test plans, conducting actual testing, and showing the test results
- project – this activity is typical to a full-fledged engineering process, that is, developing software in four stages, with formal (a) requirements, (b) design, (c) implementation, and (d) testing; this is the most sophisticated activity, which is normally done in a capstone course in computer science or software engineering programs

- research – this activity plays a crucial role in any of the STEM disciplines, by enabling the faculty work individually with students on a specific problem, guiding them through the process of a scientific inquiry, engaging them in the formulation of a hypothesis, development of a research methodology, and conducting experiments to prove or disprove the hypothesis; it is usually done in an Independent Study or Directed Research course and is meant to prepare the student for graduate studies; in the software engineering environment, such activity is normally associated with software measurements.

For this approach to work effectively in courses, the progression of acquiring knowledge from demos upwards should be built into learning objectives for specific course syllabi. However, it must be noticed that for undergraduate research activities, it is up to the instructor teaching a specific course, to make a selection which particular activities from the hierarchy outlined above will be actually pursued. The essential element of this process is to have students develop specific skills first, at multiple levels of sophistication, including demos, exercises and assignments, to ensure that the learning process and the acquisition of knowledge correspond to each student's respective background. Then, in the "research triad" they conduct experiments and full-scale projects, which lead to guided research with extensive reporting.

All activities are listed here to match the application of this concept in the software engineering curriculum, however, it is meant to be only an illustration and courses in other disciplines can adopt it to their needs, for example, as far as robotics is concerned:

- Sciences – remote control of experiments in biology, chemistry and physics
- Technology – training teachers to do research in robotics education
- Engineering – research in software for remote robot operation
- Math – research in robotics control algorithms.

The next section describes briefly the robotic equipment available in the lab, which is followed by a section on case studies outlining how each discipline from the list above can use this equipment in STEM related courses.

The Robotics Equipment

Currently, there are multiple robotic devices for use in the university's Software Engineering and Robotics Lab. They can be roughly grouped in two categories, depending on the level of their sophistication: (1) devices inherently equipped with vision and/or Internet connectivity, and (2) more primitive devices, with basic robotic functions, requiring additional sensors and added gear for enhanced functionality. The first category includes:

- Anybots QB virtual presence system[18]
- CoroBot CL2 robot on wheels (equipped with Kinect sensors), from CoroWare[19]
- NAO humanoid robot from Aldebaran Robotics[20]
- Three AL5A robotic arms from LynxMotion[21]
- VEX Robotics Clawbot system.[22]

Anybots QB is a remote two wheeled personal avatar, basically a very smart webcam on two wheels that allows one to go anywhere, watch, talk or listen. It is gyroscopically stabilized so that one can control it from anywhere, using only a web browser and the arrow keys on the keyboard. The Anybot has two cameras, one facing the front and one pointing to the ground to help avoid obstacles. It also has a touch screen display, speaker, a microphone and an eye safe laser for pointing at objects.

The CoroBot is a four wheeled robot created with the goal of minimizing the complexity of robot development. It is equipped with a low-power motherboard and a PC-class CPU. One of the most important characteristic about CoroBot is that it has extensive program storage space and CPU capacity to run additional software. In addition, CoroBot's ample mounting space allows installing additional hardware components such as GPS, laser range finder, environmental sensors and more. In current configuration, it is also equipped with Kinect vision.

NAO is an autonomous, programmable humanoid robot developed by Aldebaran Robotics. These robots are generally designed to help humans rather than replace them, and to work properly they are equipped with a number of features, including: human interaction, the ability to perceive the environment around the robot, the ability to learn and adapt to changes, locomotion generally through the use of legs (although this feature is not present in the lab robot), control of arms, ability to manipulate objects and speech ability.

AL5A robotic arms from Lynxmotion can be purchased for assembly and programmed using an external embedded microcontroller. The main features include base rotation, single plain shoulder, elbow, wrist motion, a functional gripper and optionally wrist rotate. The arm uses 5 servos to allow flexibility. Robot can perform a variety of functions, either autonomously in a fixed position, or cooperatively with a vehicle on which it can be placed.

The VEX robotics design system is a robotic kit designed for introductory robotics courses. It can be programmed for three different behaviors: autonomous, radio controlled and mixed autonomous and radio controlled. Autonomous operation is achieved by pre-programmed routines responding to sensor inputs. Radio controlled behavior allows a human operator to guide the robot and modify its behavior.

Less sophisticated robotic devices also available and include the following (they also form a part of the project and included in some of the applications described in the next section):

- Multiple IntelliBrain bots from RidgeSoft, which are small robotic vehicles programmable in a dialect of Java and expandable through multiple sensors and wireless controllers, such a Bluetooth or RFID.
- Multiple Lego MindStorms NXT2.0 robotic vehicles that can be programmed in a variety of languages including a native language, as well as Java, C and LabVIEW.
- Three older generation SCARA robotic arms, with an interface via a RS232C serial port that allows full scale programming but without providing feedback to the controller.

These robotic devices, both the simpler ones and the ones fully equipped, form the overall platform for implementing the ideas of integrating the STEM curriculum in practical courses.

Applications

In this section, we describe how these robotics devices have been used to develop specific research projects, using the same six-step process as explained in a section on General Methodology, throughout a cross-section of STEM disciplines:

- Sciences – remote control of experiments (Anybots)
- Technology – training teachers to do research (Mindstorms NXT2, IntelliBrain)
- Engineering – research in software for remote robot operation (Lynxmotion, NAO)
- Math – research in robotics algorithms (Corobot).

Science Application – Remote Operations with Anybots

Using Anybots robotic device, in its simplest version, does not require any programming skills, only learning the device operation after logging on to the Internet account (essentially, on the cloud, see Figure 2). Then, the robot can be controlled manually in a number of configurations, depending on the need of a specific project or a course:

- one-to-one, which dedicates the Anybot avatar to one user and can be used for remote operation, such as observation of chemical reaction, animal behavior, etc.
- one-to-many, which allows a user to visit several labs consecutively, by moving and stationing Anybots in each of them
- many-to-one, which allows multiple users at different times to utilize the Anybot, for example, multiple students in a course to watch a specific chemical reaction
- many-to-many, which is the most sophisticated and allows multiple persons to monitor and use multiple Anybots.

From the potential research perspective, the Anybot has a gameable aspect. Since there are only a set of very few movements (only lateral movement), a game can be constructed using only the perspective the Anybot has with the camera. A simple, yet effective game could be designing a maze. Testing maze completion from individuals using the Anybot could cover a wide array of tests and studies, without having to build a maze the size of an individual. Also, since this game is targeted towards a demographic of people who are bedridden, the game could be a more realistic and enjoyable aspect of life that they cannot currently obtain.

For another research application, one key piece that could be added to the Anybot is an RFID transmitter. RFID could allow the robot to gain access to rooms and doors. Rather than leaving doors open, in which noise and air could circulate out, the doors could remain closed and only open when the Anybot drives up to them. RFID also can give some extra information for use of a game, such as simulating the Anybot "picking up" an object in the real world. By imposing a RFID chip on a hammer with a unique signature, and then using the RFID chip on the robot, a user could drive to the hammer and the hammer chip could notify the server computer that the robot is now near the hammer. Thus, the server can adjust the game state accordingly (by notifying the user they now have a hammer equipped).

Fig. 2. Physical diagram of a prospective remote access of Anybots with Android.

Technology Application – Training High School Teachers

This project involved teaching students basic and intermediate programming skills for robotic programming, using Lego Mindstorms NXT2 and RidgeSoft IntelliBrain platforms. The Mindstorms projects used three languages available for programming these devices:

- C programming based on the NXC (Not eXactly C) language
- programming in Java, which allowed GUI implementation using the Swing API
- native G language for graphical programming with basic multithreading.

Several lessons were designed, with the teaching objective to master application programming in a gradual manner by proceeding from the simplest to the most complicated topics:

- Lesson 1: Basic Setup and Bluetooth Connectivity
- Lesson 2: Moving the Robots
- Lesson 3: Recording Movement
- Lesson 4: Using the Sound Sensor
- Lesson 5: Using the Touch Sensor
- Lesson 6: Using the Ultrasonic Sensor
- Lesson 7: Using the Color Sensor
- Lesson 8: Final Project

What turned out to be particularly challenging was the design of Lesson 6. By the time this lesson is reached, students should have a reasonable grasp of programming concepts such as if/else statements and looping. The focus shifts to the use of the ultrasonic sensor to allow the robot to "see" its surroundings. In this lesson, the students have the problem of how to program a robot to "hunt" using this new sensor. Autonomously, the robot should continuously look for a target to come within range. As soon as this occurs, it should move directly forward while simultaneously shooting at the target. Manually, users should be able to directly control the robot. Once a user brings the robot in range of a target, it should automatically fire at it (as long as it is still in range). This lesson relies heavily on studying the possible robot's states and conditions.

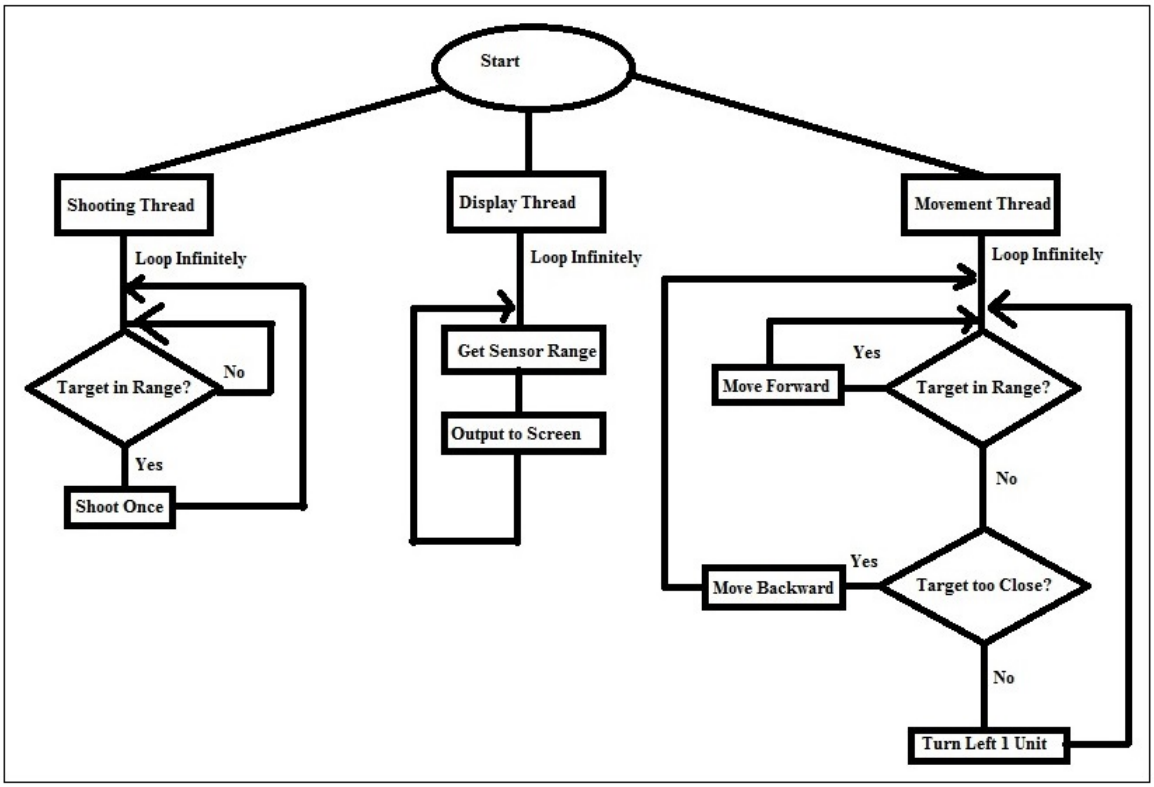

Fig. 3. Illustration of a multithreaded application for a Mindstorms NXT2 robot.

From a design standpoint, this lesson utilizes three threads of programming (Figure 3). The first thread is responsible for controlling the movement of the robot. Autonomously, the robot will circle until an obstacle comes in range, at which point it will start moving towards it while shooting (thread 2). Manually, the robot will not allow the user to move forward if it is too close to a target, and it will shoot any time a target is in range. The movement is displayed by thread 3.

Engineering Applications – Software for Remote Robot Operation

There are two respective projects currently underway, both for remote software development, upload and operation of robotic devices: Lynxmotion and NAO, respecively.

Lynxmotion Robotic Device. The Lynxmotion Robotic Arm Remote Control is focused on development in C# and robot operation under Windows Embedded. For the physical diagram, as shown in Figure 4, the solution is comprised of the following components:

- Camera Server: This server application is responsible to transmit live video feed to the client software. This provides visual feedback to the user as the robotic arm reacts to the user input commands.
- Robot Server: This server application is responsible for transmitting input commands to the robotic arm. This server application constantly listens for incoming remote connections. Once a connection is established, the remote user can send input commands to the Robot Server application which in turn sends them to the robotic arm.
- Robot Remote Client: This client application connects to the server software to capture the live video transmission sent by the Camera Server software and to send motion commands to the Robot Server software.
- Update Server: The main feature of this application is to receive new Robot Server executable files in order to update the currently running Robot Server application.
- Update Client: This client software is used to upload a new Robot Server executable file to the eBox 2300.

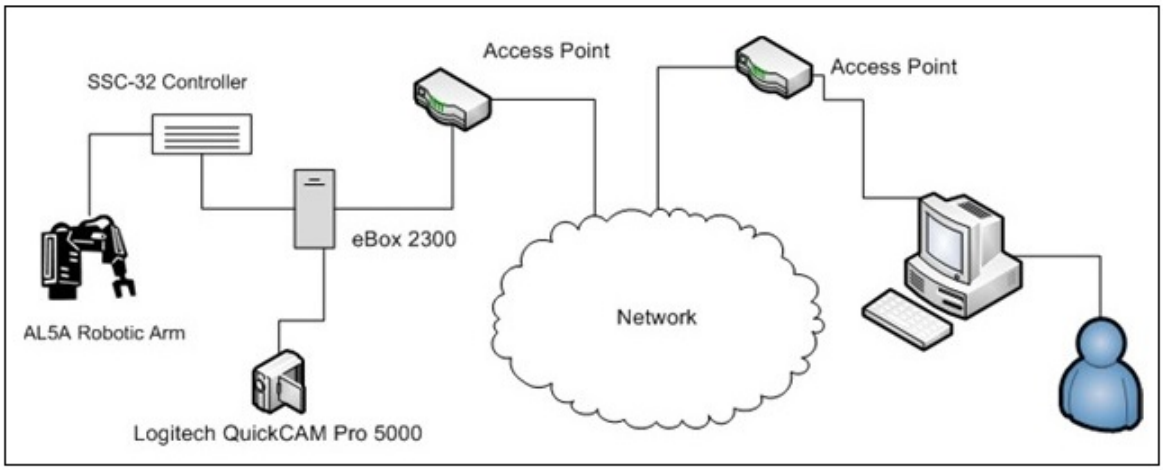

Fig. 4. Physical diagram of a Lynxmotion robotic arm remote control.

From the software engineering perspective, the project resulted in redesigning the Update Server to function as a true multi-threaded server solution which provided a much more robust software update platform for the Robot Server application. Additionally, the new backup and rollback facility added the extra stability required to provide a 100% availability of the Remote Control Robotic Arm infrastructure. The enhancement of the Update Client application, which prevents malicious uploads of a Robot Server application, added to the overall Remote Control Robotic Arm platform an extra layer of security towards the overall stability of the software solution.

NAO Humanoid Robot. The problem addressed in this project is how to turn the NAO robot into a security watchdog. The initial requirements were established as follows:

- The NAO robot shall utilize its camera and other sensors to determine if an object is present in its view.
- The NAO robot shall recognize objects and/or persons as they enter the robot's view.
- The NAO robot shall turn its head to keep objects and/or persons within its view.
- The NAO robot shall stream the video data across the Internet where a remote user can access the feed.
- The NAO robot shall be controlled remotely in order to provide further security options.
- The NAO robot shall be controlled remotely through a WebSocket user interface.

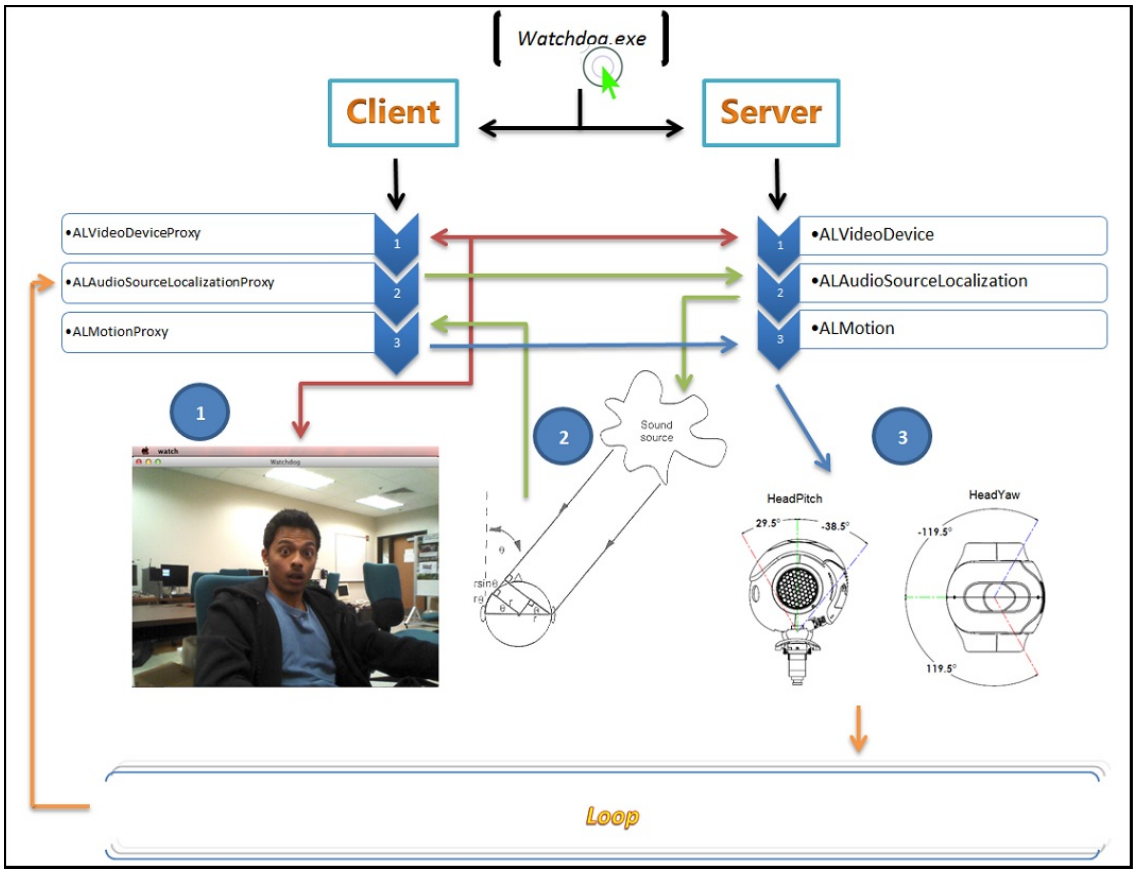

Fig. 5. Physical diagram illustration of the NAO Watchdog.

The development resulted in using the NAO robot to act as a security camera (Figure 5). It takes live images from the NAO's camera modules and returns them to a client computer. It will then use those images to create a live video feed. The robot also finds the source of sound and turns its head to keep a live feed on the source. This allows the robot to act dynamically and track objects and report footage of the object.

Math Application – Graphics for Kinect with CoroBot

Robotic vision is the study of deciding which combination of known methods to apply to an image along with their corresponding parameters to produce the information the robot needs to accomplish its task. Teaching the robotic vision at the undergraduate level is normally a challenge because generally it requires to cover sufficient computer vision to allow the student to perform some basic vision functions needed for robotics yet the material only represents a small fraction of the total course material.[15]

The usual approach involves either covering the material in theory only with no hands-on programming practice or alternatively having a hands-on assignment where the student only implements a small subset of the algorithms. Generally there is insufficient time to allow the student to implement and test a fully functioning, albeit simple, robotic vision system. While this may be sufficient for an in introductory robotics course in some engineering programs, it's not appropriate for software engineering. Requiring the software engineering student to implement a complete vision system is not only expected but also possible with the help of an appropriate tool, so the student can learn respective algorithms a d understand them visually.

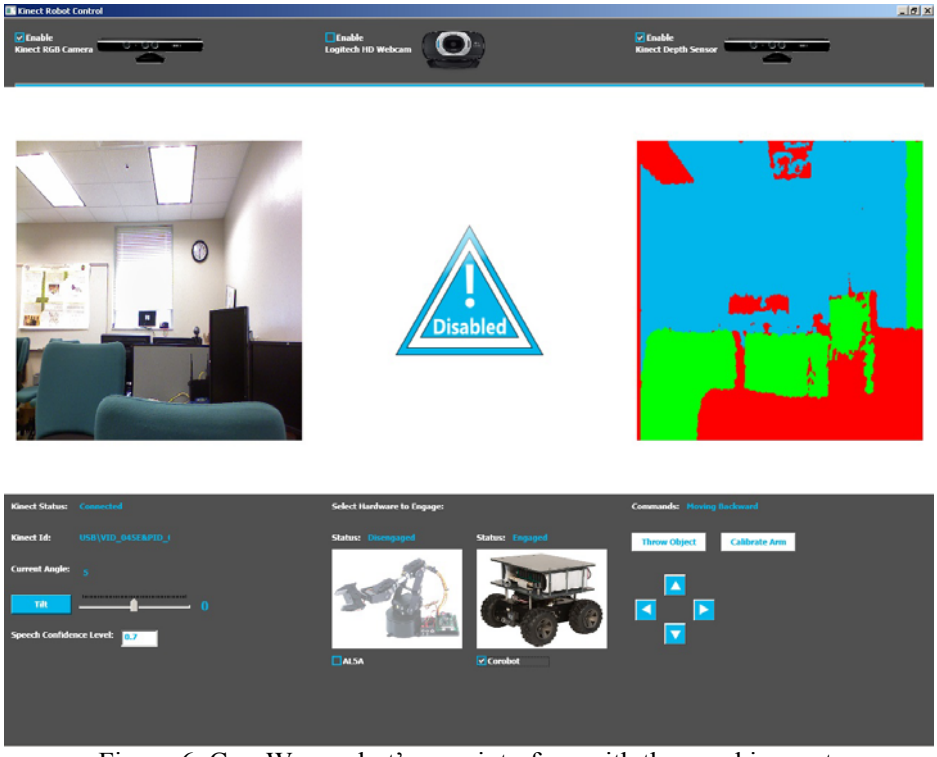

Figure 6. CoroWare robot's user interface with the graphics part.

In a practical application, robotic vision lessons are useful in mathematical calculations for the Kinect vision of the CoroBot. The Kinect sensor's three components: the camera, the depth sensor and the microphone can be used in respective algorithms to move the robot. In this project, only the camera and depth sensors are utilized, with results of image analysis as shown on the right hand side of Figure 6.

Conclusion

With the overwhelming need for focusing education on STEM disciplines, on one hand, and the necessity to educate the population in robotics technologies, on the other hand, this work attempts to combine both aspects by introducing research opportunities for undergraduate students in robotics across the STEM curriculum. To do so, a general methodology has been developed, which allows for gradual acquisition of knowledge, via demos, exercises and assignments, first, through experiments and projects to supervised research.

Students would acquire various levels of STEM knowledge and the ability to document its acquisition and learning processes. It has been shown that the projects have a potential impact on advancing teaching in STEM disciplines by connecting the lab and integration of multiple STEM related courses based on robotics, as follows:

- Sciences – remote robotic control of experiments.
- Technology – training teachers to do research in robotics education.
- Engineering – research in software for remote robot operation.
- Mathematics – research in robotics algorithms.

At this stage of the project, it is too early to share more specific lessons learned. One important thing to be noticed is that with this variety of equipment and the level of its sophistication, it is a tremendous challenge for instructors and technicians to keep it all running smoothly. On the other hand, keeping the equipment in operation enforces the students to acquire important practical skills. Although not an intrinsic part of research itself, it actually becomes an important component of doing research in any of the STEM disciplines.

The immediate future step in this work is to provide "ready-made" learning units (such as modules or learning objects) based on these individual projects and offer them to instructors teaching STEM course beyond the Software Engineering discipline. Steps have been made to attract respective interest not only in engineering, sciences and math, but also in courses unrelated to STEM, such as Digital Media, to produce videos on development and behavior of robotic devices, and even Composition, to write term papers on the societal perception of robotic technologies.

In summary, the entire endeavor turned out to allow students and instructors explore new avenues for pursuing STEM education. The scope and technical level of the projects have to be verified to ensure the quality of the developed approaches. Likewise, various aspects of pedagogy have to be analyzed, to document the validity of the approach and allow for broader dissemination of respective concepts and artefacts.

Acknowledgements

This project has been supported in part by NASA through the UCF's Florida Space Grant Consortium, Award No. NNX10AM01H. Additional funding has been provided by the Small Business Administration, grant SBAHC-10-I-0250, and by the National Science Foundation under Award No. 1129437. Thanks go to FGCU's Computer Science and Software Engineering students, Nicholas Alteen, Abraham Baquero, Mike Bizub, Sam Caguana, James Carroll, Vincent Giannone, Matt Grojean, Nathan Hart, Austin Hughes, Jessel Serrano and Tyler Thomas, for their contributions to the development of respective experiments and projects.

References

1. National Science Board, A National Action Plan for Addressing the Critical Needs of the U.S. Science, Technology, Engineering and Mathematics Education System, Report NSB-07-114, October 2007.
2. Computing Community Consortium, A Roadmap for US Robotics: From Internet to Robotics. Report from the Workshop on Emerging Technologies and Trends, May 21, 2009.
3. Zalewski J., "Lab-by-Wire: Fully Web-based Hands-on Embedded Systems Laboratory", *Proc. EDUCON2013, IEEE Global Engineering Education Conf.*, Berlin, Germany, March 13-15, 2013, pp. 928-933.
4. Papert S., Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York, 1980.
5. Barker, B., Ansorge, J. "Robotics as Means to Increase Achievement Scores in an Informal Learning Environment." *Journal of Research on Technology Education*, Vol. 39, No. 3, p. 229-243, 2007.
6. Berland M. et al., "Programming on the Move: Design Lessons from IPRO." *Proc. CHI 2011, ACM Human Factors in Computing Systems Conference,* Vancouver, Canada, May 7-12, 2011, pp. 2149-2154.
7. Williams K. et al., "Enriching K-12 Science and Mathematics Education Using LEGOs." *Advances in Engineering Education*, Vol. 3, No. 2, Summer 2012.
8. Green A.A., "Lego Robotics in STEM Education," *D&T Practice*, No. 1, pp 22-24, 2013.
9. Mataric M.J., Koenig N., Feil-Seifer D., "Materials for Enabling Hands-on Robotics in STEM Education." *Proc. AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*, Stanford, Calif., March 26-28, 2007.
10. Roy N. et al., "The Experience of Teaching Software Development in a Robotics Project Course." *Proc. 3rd International CDIO Conference*, Cambridge, Mass., June 11-14, 2007.
11. Christensen H.I., "Software Engineering in Robotics," College of Computing, Georgia Tech, Atlanta, Georgia, 2010. URL: http://www.cc.gatech.edu/~hic/8803-SER-10/Welcome.html
12. Göbel S., Jubeh R., Raesch S.L., "A Robotics Environment for Software Engineering Courses." *Proc. 25th AAAI Conf, on Artificial Intelligence*, San Francisco, Calif., August 7–11, 2011, pp. 1874-187.
13. Flot J. et al., "Learning How to Program via Robot Simulation." *Robot Magazine*, Issue 37, pp 68-70, November/December 2012.
14. Gonzalez F., Zalewski J., "A Software-based Robotic Arm Kinematic Simulator for Use in Teaching Introductory Robotics Courses." *Proc. 121st ASEE Annual Conf.*, Indianapolis, Ind., June 15-18, 2014. Paper No. 9835.
15. Gonzalez F., J. Zalewski, "A Software-based Robotic Vision Simulator for Use in Teaching Introductory Robotics Courses" (submitted for publication).
16. Bybee R.W., The Case for STEM Education: Challenges and Opportunities, National Science Teachers Association, 2013.
17. Kenny R., J. Zalewski, "Physical Computing: An Applied Approach to STEM Education", Collier County Public Schools 2013-2014 STEM Conference, Naples, Florida, October 12, 2013. URL: https://apps.collierschools.com/events/Pages/Guest/9/EventSessionList.aspx
18. Anybots Virtual Presence Systems. URL: https://www.anybots.com
19. CoroBot Explorer. URL: http://robotics.coroware.com/corobot-robots
20. Nao for Education. URL: http://www.aldebaran-robotics.com/en/Solutions/For-Education/introduction.html
21. AL5D Robotic Arm. URL: http://www.lynxmotion.com/
22. VEX Robotics Education Design System. URL: http://www.vexrobotics.com/vex