AC 2009-1681: BALANCING BREADTH AND DEPTH IN ENGINEERING EDUCATION: UNIFIED ROBOTICS III AND IV

William Michalson, Worcester Polytechnic Institute

Dr. William R. Michalson is a Professor in the ECE Department at the Worcester Polytechnic Institute where he performs research and teaches in the areas of robotics engineering, navigation, communications and computer system design. He supervises the WPI Center for Advanced Integrated Radio Navigation (CAIRN) where he is developing a Public Safety Integration Center focused on the integration of communications, navigation and information technologies for public safety applications. His research focuses on the development, test, and evaluation of systems for both civilian and military applications with a special emphasis on techniques focused on indoor, underground or otherwise GPS-deprived situations. Prior to joining the faculty at WPI, Dr. Michalson spent approximately 12 years at the Raytheon Company where he was involved with the development of embedded computers for guidance, communications and data processing systems for space borne and terrestrial applications.

Gregory Fischer, Worcester Polytechnic Institute

Taskin Padir, Worcester Polytechnic Institute

Gary Pollice, Worcester Polytechnic Institute

Balancing Breadth and Depth in Engineering Education: Unified Robotics III and IV

Abstract

The Robotics Engineering program at the Worcester Polytechnic Institute integrates electrical engineering, mechanical engineering and computer science concepts into a series of unified courses in robotics at the undergraduate level. A need to pack a large amount of technically and philosophically diverse multi-disciplinary material has created a number of challenges. Traditional engineering courses tend to cover a large amount of foundation material along with numerous examples of how this foundation applies to relatively ideal problems. Unfortunately, there simply is not enough time to build sufficient foundation in three different engineering and science disciplines. Further, attempts to do so would virtually ensure that we would not engage students quickly in their chosen area of robotics engineering. This paper describes the approach taken to balance conflicting goals and show how future generations of robotics engineers might be educated.

Introduction

The Robotics Engineering (RBE) program at the Worcester Polytechnic Institute (WPI) is an attempt to integrate electrical engineering, mechanical engineering and computer science concepts into a series of unified courses in robotics at the undergraduate level. Two Sophomore-level courses, RBE 2001 and RBE 2002, introduce students to many of the basic concepts of robotics at an introductory level. In these courses, students gain hands-on experience in the analysis of robot system components and in the implementations of simple robots to perform various tasks. Two Junior-level courses, RBE 3001 and RBE 3002, build on this foundation to ensure that students not only understand the analysis of selected components in a robotic system, but also gain an appreciation of the "system-level" design issues associated with the development of modular robotic systems.

Packing such a large quantity of technical material, covering such a technical and philosophical breadth of topical material, is an extraordinary challenge. In contrast, engineering professors are typically quite comfortable teaching a large amount of "foundation" material along with numerous examples of how this foundation applies to "ideal" problems. This approach echoes the "pour it in" model referred to by Smith were he states:

> *Consider the most common model of the classroom-based teaching and learning process used in engineering education in the past fifty years (and maybe currently?). This model ... is a presentational model where, as one pundit quipped, "the information passes from the notes of the professor to the notes of the students without passing through the mind of either one."*[1]

Using this model of education, students are simply expected to "know" how to use this "poured in" material to synthesize complex systems. Assuming that students will "just know" how and when to apply foundational concepts to solving engineering problems is an inefficient process at best. Relevant to this inefficiency is the observation by Turns that:

> *"Physics is clearly relevant to engineering, so it is likely that research on how students understand physics concepts and solve physics problems is relevant to engineering education. At the same time, because such research is often done with students working toward different degrees (e.g., medicine) and/or with problems that are not specifically within an engineering context, the transfer of these research results to engineering students and engineering problems is not well understood."*[2]

This observation by Turns lies at the crux of the problem. Students tend to learn facts in their, in this case Physics, classes but, like the researchers referred to in the above quote, they don't necessarily learn how those facts are relevant to their chosen discipline. This phenomenon was actually observed during an internal bi-annual review of the capstone design projects in the Electrical and Computer Engineering Department at WPI where it was observed that students were having a difficult time synthesizing designs.[3] To correct this problem, a radically different course in ECE Design was developed to teach the fundamentals of designing electrical systems to students at the end of their sophomore year. With this course we were able to reach students immediately after foundational material was "poured in," allowing us to show them what the material was "good for."

Over the course of 7 weeks, students in the ECE Design course perform market analysis, develop system requirements, design, implement and demonstrate electronic system which satisfies their derived requirements. Classroom lectures focus on the process and methods of engineering design, while the laboratory associated with the course focuses on design reviews and individual designs.[4] This class is extremely time-consuming for the students (typically 15-20 hours per week), but consistently receives very positive student reviews. Subsequent outcomes assessments both internally and through ABET have confirmed improvements in the abilities of students to perform design synthesis.

The area of robotics provides a new opportunity to capture the interests of students in grades K-12 and to introduce them to engineering and science. Growth in the multi-disciplinary field of robotics, and a perceived need for engineers trained in Mechanical Engineering, Electrical Engineering and Computer Science led the Worcester Polytechnic Institute to create a new undergraduate degree program in Robotics Engineering in 2007.

Robotics Engineering provides an excellent opportunity to engage students early in their engineering education and to leverage their existing enthusiasm into a meaningful engineering education. Currently, students are exposed as early as K-12 to a growing number of robot competitions such as BotBall (http://www.botball.org), US FIRST (http://www.usfirst.org), Robo-CupJunior (http://rcj.sci.brooklyn.cuny.edu)[5], BattleBots (http://www.battlebots.com) and many others. Strong ties between these competitions, student enthusiasm, research, and education have been observed.[6]

This means that students entering their degree program are: a) very enthusiastic about learning "more about robots" and b) generally only have "hobbyist" knowledge of the science that underlies the robots they have seen and/or constructed. The goal is to create a program in which this “hobbyist” knowledge is transformed into “engineering” knowledge while maintaining the level of interest and enthusiasm that brought them to engineering in the first place. As Donovan observed:

> *“Students come to the classroom with preconceptions about how the world works. If their initial understanding is not engaged, they may fail to grasp new concepts and information presented in the classroom, or they may learn them for purposes of a test but revert to their preconceptions outside the classroom.”*[7]

While foundational courses do, indeed, have their place, foundational concepts often don’t engage students because of the abstract manner in which they ultimately relate to the students interests. Students are often quite comfortable being given something to do, picking a potential solution, and then trying tirelessly to fit their solution to the problem. Often, the result is frustration on the part of faculty who wish students would extend basic concepts to account for system phenomena, and frustration on the part of students who desperately try to hack their way to a solution because “the theory doesn’t work.” The authors believe that there is a better way.

In the area of Robotics Engineering, it is relatively easy to engage students on turf they are already familiar with. Starting with an introductory first-year course in robotics students use hobbyist-style laboratory equipment to perform a series of experiments that get them in the lab quickly, doing projects similar to those they are already intuitively familiar with. The difference lies in a more structured environment and a series of experiments meant to highlight physical and mechanical properties of the systems they build.

In the second year, students have a broader educational background consisting of traditional coursework in mathematics, physics, electrical engineering, mechanical engineering and computer science. In addition, these second year students are again exposed to robotic systems, now from a more sophisticated engineering perspective. At the second-year level, students are now taught about the mechanical and electrical systems of robots. In an associated laboratory, they apply these concepts to the construction of a variety of robots for specific tasks. In the classroom, lecture material is selected not to provide broad foundational concepts, but rather to provide sufficient depth of treatment for the students to readily understand why the material is important to their interests – in other words, lecture material is carefully selected and presented so that it is clear to the students what it’s “good for.” This is then reinforced in the laboratory where students now build robots to validate classroom concepts. At this level, many of the computer hardware and software details are still hidden from the students so they can concentrate on understanding higher-level principles and developing intuition.

In the third-year courses we progress into a detailed understanding of the theoretical and practical issues associated with building real-world systems. We diverge significantly from the classic approach by focusing on the theory necessary to do a limited number of useful tasks, and then immediately moving the theory to its mechanical, electrical and computer implementation. Students are then able to see both theory and practice applied to something that they have already been exposed to at a lower level previously. While this reduces the amount of “general” theory the students see, it builds their toolbox of knowledge so that when they see something similar they not only know where to find the appropriate theory, they know how they might apply it in practice. The result is a strengthening of theory, practice and intuition in the students.

It is in these third-year courses that students take an in-depth look at the physical interactions between a robot and its environment. In these courses, students learn not only the mathematical principles behind topics such as dynamics and inverse kinematics, but also learn how these principles translate to real-world electrical specifications, circuit designs and real-time programs. In so doing, the students are exposed to concepts like electrical, mechanical and computational errors, error propagation, noise, sensor function (and limitations) and other concepts that make the development of robotic systems which interact with an uncertain environment challenging.

2000-Level Robotics – Curriculum Overview

The sophomore-level courses, Unified Robotics I and II (RBE 2001 and RBE 2002), emphasize the foundational concepts of robotics such as kinematics, stress and strain, pneumatics, circuits, operational amplifiers, electric motors and motor drive circuits, sensors, signal conditioning and embedded system programming using C language. Professors from each of the three primary disciplinary areas team-teach various components of each course. The goal is to introduce students to the analysis of electrical and mechanical systems as well as the principles of software engineering. In both courses, the emphasis is on robotics applications, project-based learning and on the relationship among the electrical engineering, mechanical engineering and computer science disciplines as they apply to robotics.

The focus in RBE 2001 is the effective conversion of electrical power to mechanical power. The course provides a hands-on introduction to embedded systems programming as well as the analysis of mechanisms and electrical circuits within the context of robotics engineering. Course topics and laboratory experiences include position, velocity, acceleration and force analysis of simple mechanisms, electric motors, H-bridges and motor drive circuits, an introduction to control systems and PID controllers, C programming and software engineering.

RBE 2002 emphasizes the interaction of a robotic system with its environment through sensors and feedback. Concepts of stress and strain in the context of force sensing, material properties and operational principles of sensors used in robotics, basic signal conditioning such as amplification and filtering, as well as programming strategies for integrating multiple sensors are introduced.

In combination, RBE 2001 and RBE 2002 provide a study of the foundations of robotics by integrating the fields of computer science, electrical engineering and mechanical engineering and prepare students for the advanced robotics courses.

Providing such a broad foundation in the 2000-Level robotics courses necessarily requires making compromises in the number of topics covered and the depth coverage in any one topic. It is simply not possible, given practical constraints on class time and student load to introduce students to everything they might require to engineer a robotic system. To balance these conflicting constraints, certain compromises are made in the delivery of the material to the students and in the exercises performed in the laboratory.

The first compromise relates to the material that is selected. Rather than attempt to teach all of the material that might normally be associated with a 2000-level course in any one discipline, the choice was made to pare the material to that which is essential to provide sufficient depth for the students to understand the related laboratory exercises. In this context, the emphasis in the classroom is on the most commonly encountered concepts rather than interesting special cases. In determining curriculum content, every topic is scrutinized to ensure that it is actually used for some significant purpose in the classroom, on homework, in exams and in the laboratory. Topics that can't pass such a "what's it good for" test, are (grudgingly) omitted in favor of the benefits associated with engaging students in the material.

A second compromise relates to the laboratory exercises. In the laboratory students largely work with hobbyist-style hardware and software elements which, while sufficient to reinforce concepts introduced in the classroom, hide many of the lower-level details of the devices they use in the laboratory. This provides a stable environment which allows students to focus on electrical, mechanical or computer science concepts introduced in class without having to resort to "hacking" in other areas.

The result of these compromises is that students at the 2000 level have enough theoretical knowledge to "mostly" know how to approach a laboratory problem, and have a set of tools in the laboratory which allow them to rapidly prototype their solution. Many of these solutions fail on their initial attempt, which tends to prompt the students to stay engaged, revisit their errors and iterate on their designs. The result is a reinforcement of classroom theory, the development of better intuition from seeing ideas that don't work, and an increase in their willingness to iterate towards a better design.

The sophomore-level courses, Unified Robotics I and II are discussed in a companion paper at this conference in detail.

3000-Level Robotics

The focus of this discussion is on the junior-level courses, Unified Robotics III and IV (RBE 3001 and RBE 3002), that build upon the intuition that the students began to develop in the 2000-level courses. It is in these courses that the students actually begin to understand and appreciate the details underlying their 2000-level experience. These junior-level courses provide a much deeper theoretical coverage of robotics, including: kinematics and inverse kinematics, control systems, sensors, signals, reasoning with uncertainty, navigation, world modeling and planning. In these courses students no longer have pre-packaged hardware and software components; they now are introduced to interrupt-based programming, software system architecture, object-oriented design and in-circuit debugging.

The focus in RBE 3001 is on developing a deeper understanding of the types of devices they encountered in RBE 2001 and 2002. The course begins with an introduction to the Atmel AVR-series of 8-bit microcontrollers which provide the computational platform for all of the experiments done in the laboratory. These experiments involve topics such as: real-time interrupt-based programming; control of a single axis robot arm; control of a multiple link robotic manipulator; characterizing encoders, accelerometers and magnetometers; characterizing infrared and ultrasonic rangers; and developing a simple, but complete, pick and place robotic system.

Learning outcomes for RBE 3001 are:

- An understanding of how to develop and debug event-driven programs.
- An understanding of the dynamics of a single axis robot arm.
- An understanding of the kinematics and inverse-kinematics of a multi-axis robot arm.
- An understanding of sensors and sensor errors, sample rates and dynamic range.
- An understanding of ranging systems and ranging system errors

The focus in RBE 3002 is on integrating the information in the previous three courses into a complex robotic system. This course begins with an introduction to object-oriented programming and a framework based on a communication protocol between a PC and a robot. By incorporating hardware and software components developed in RBE 3001, the students perform experiments which involve topics such as: hardware/software partitioning; control of a mobile platform; multi-sensor data fusion, motion planning, world modeling and reasoning in the presence of uncertainty.

In teaching the course, each week begins a new major topic and breaks that topic into four components that cover the system hardware requirements, hardware implementation choices, software requirements and software implementation choices. Each of these aspects of a major topic is given (typically) a day of lecture time. This way, for each topic students see how the various mechanical, electrical and computer components of a system interact with each other. In addition, the students see immediately how choices made in one disciplinary area affect the choices in another disciplinary area.

Learning outcomes for RBE 3002 are:

- An understanding of system decomposition and requirements.
- An understanding of hardware and software partitioning.
- An understanding of system errors and error propagation.
- An understanding of world models and modeling.
- An understanding of navigation and navigation system errors.
- An understanding of path planning and planning with uncertainty.

Leveraging 2000-Level Courses

When students complete the 2000-level Unified Robotics courses, they have developed a basic theoretical understanding of robot-related topics, good intuition related to mobile robot platforms, actuators and sensors, and considerable experience designing simple robots to complete relatively well constrained tasks. At this level, however, they don’t necessarily understand how the hardware and software components they are using actually work. Their access to the hardware and software details of the equipment they are using is limited, as is their ability to manage real-time constraints.

Upon entering the junior-level unified robotics courses, students begin to explore topics they were able to take for granted at the sophomore level. In these courses, students are provided a number of components as shown in Figure 1, which include:

- A custom-designed 2-axis robotic arm (the "EduArm") which is composed of modular joints powered by DC motors with incorporated optical joint encoders and potentiometers for feedback,
- The "EduBot" compact, modular mobile robot platform with rear differential drive by dual encoded DC motors, omnidirectional front wheels, battery power, and a modular frame allowing attachment of standard components including the EduArm.
- Embedded controller hardware including an AVR microcontroller, analog and digital inputs and outputs, linear and switching motor amplifiers, power distribution and communications
- Software libraries with varying levels of abstraction for embedded control of the system with varying levels of abstraction
- Programming and debugging connections between the robot and a PC
- Wired and wireless communications with a PC for hierarchal/supervisory control of the microcontroller and data logging

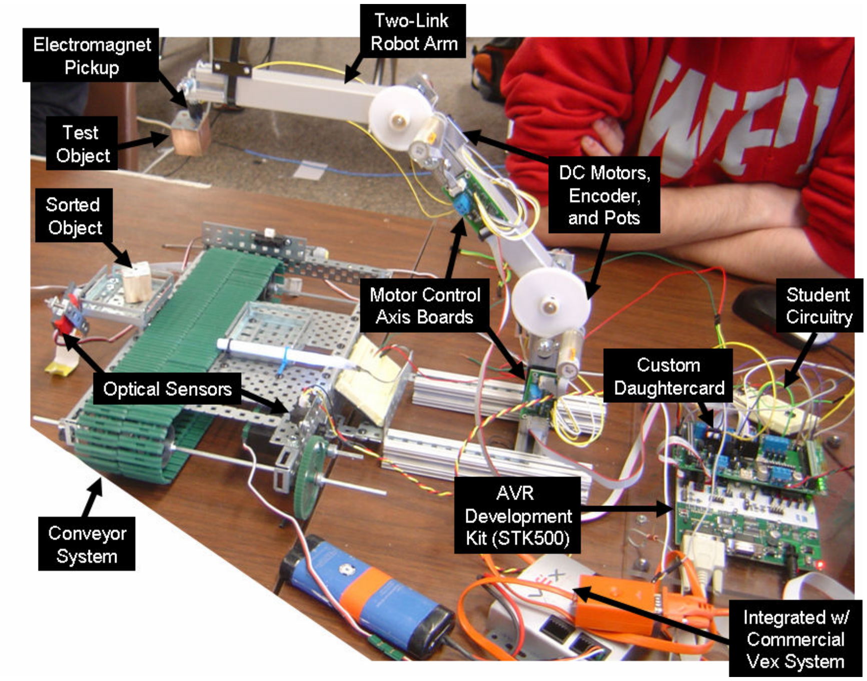

Figure 1. Prototype EduArm system including the 2-Link arm with DC motors, encoders, potentiometers, gear trains, magnetic gripper, sensors, control circuitry, and microcontroller.

The STK-500 development system for Atmel's AVR microcontrollers provides a basic platform which allows serial I/O, provides LED indicators, pushbutton switches and header connectors for accessing the I/O ports of the ATmega644 processor (located underneath the custom-made expansion daughter card). Students are able to purchase the STK-500 boards at a discounted price, which allows them to develop software outside of the laboratory. Development tools consist of AVR Studio 4 and WinAVR which are freely downloadable, but which provide the features necessary to upload and debug programs written in C/C++ or AVR assembly language. This combination provides students with a low-cost way of obtaining a fairly powerful programming environment. In the laboratory students also have access to JTAG MKII interfaces for in-circuit debugging.

The expansion board provides the following features:

- 2 independent linear motor control channels
- 2 independent motor control channels with H-Bridge outputs
- 4 channel, 12-bit digital-to-analog converter (DAC)
- Selectable +5V source
- Highly configurable using on-board jumpers
- Support for two axis control boards
- Support for one ultrasound interface board
- Support for one infrared sensor module
- Support for one compass/accelerometer board.

In addition, custom hardware was developed for axis control, ultrasonic sensing, accelerometers and a magnetic compass.

Given these tools, students are now required to handle all of the low-level details they were able to ignore at the 2000-level. A typical laboratory exercise consists of:

- Reviewing schematic diagrams of the expansion card (and other cards used),
- Reviewing component datasheets as necessary,
- Developing math models based on theoretic principles,
- Developing code to implement the derived math models (for example moving an arm with gravity compensation),
- Developing code to measure and record real-time data as the system operates,
- Transferring data from the system under test to a PC for subsequent analysis, and
- Analyzing results using tools such as Matlab to compare their implementation to theory.

The 3000-Level Philosophy

The philosophy underlying the content and design of this resource package described above is to provide a development environment that is structured enough to avoid students wasting time troubleshooting unreliable equipment, and yet is unstructured enough that non-trivial design decisions are made by students. Since the resource package is not a fixed, pre-configured platform, each student's approach to satisfying a laboratory and/or course project outcome is unique. While the components are chosen to simplify assembly and interface concerns at the mechanical, electrical and software levels, it is not a kit with a pre-planned structure and the limitations that such kits pose.

Therefore, the environment provides a friendly undergraduate experience that does not limit freedom of choice, requires students to understand core course concepts and requires students to make critical design decisions in order to be successful. Unprepared students who do not apply engineering design principles across a range of disciplines will be able to build robotic systems that fail while in operation. Students are not protected from engineering science and disciplinary canon by encapsulated, pre-engineered and sterile plug-and-play subsystems.

For example, students will need to understand the current/voltage driving requirements of various inputs/outputs, the torque versus speed characteristics of a given motor, the need to wrap a data stream within a specific frame format and means to link software modules constructed in different source languages – but find no impediments to doing so that would overly consume valuable academic course time.

Conclusions

To date, the implementation of RBE 3001-3002 appears to be having the desired effects of a spiral curriculum. In some sense, everything the students do in the laboratory is directly related to something they have done in the previous RBE 2001-2002 courses in that everything is directly related to something the students have developed an intuition about. The primary difference is that students begin to see the difference between a hobbyist and an engineer.

By focusing the classroom work on theoretical concepts that are directly relevant to the laboratory work, students seem better able to see the connections between theory and practice. Since the hardware environment is stable (unlike prototyping boards), students in the laboratory are better able to focus on translating theory to implementation. This translation necessarily requires the students to apply knowledge gained at the RBE 2001-2002 level about software and circuits, therefore reinforcing and expanding those basic concepts.

In summary, although 2008/09 marks the first offering of the entire four-course sequence, it appears that this approach is leading to better retention of material in previous courses as well as a highly motivated student body which is excited to really see "how things work." A post-mortem on lessons learned, student feedback and a general assessment of "what went right" and "what went wrong" is planned for the summer of 2009.

Bibliography

1. Smith KA, Sheppard SD, Johnson DW, Johnson RT, Pedagogies of Engagement: Classroom-Based Practices, Journal of Engineering Education, pp 1-15, Jan 2005.
2. Turns J, Atman CJ, Adams RS, Barker T, Research on Engineering Student Knowing: Trends and Opportunities, Journal of Engineering Education, pp 27-40, Jan 2005.
3. Michalson WR, Labonte RL, Capstone Design in the ECE Curriculum: Assessing the Quality of Undergraduate Projects at WPI, American Society of Engineering Educators 1996 Annual Conference (CD-ROM), session 1232 Washington, D.C., Jun, 1996.
4. Polizzotto L, Michalson WR, The Technical, Process, and Business Considerations for Engineering Design, Frontiers in Education 2001, Reno, NV, pp. F1G-19-F1G-24, Oct 10-13, 2001.

5. Sklar E, Eguchi A, Johnson J, RoboCupJunior: Learning with Educational Robots, AI Magazine, vol. 24, no. 2, pp. 43-46, 2003.

6. Almeida L, Azevedo J, Cardeira C, Costa P, Fonseca1 P, Lima P, Ribeiro F, Santos V, Mobile Robot Competitions: Fostering Advances In Research, Development And Education In Robotics, CONTROLO'2000 4th Portuguese Conference On Automatic Control, pp. 1-6, 2000.

7. Donovan MS, Bransford JD, Pellegrino JW, How People Learn: Bridging Research and Practice, National Academy Press, 1999.