

A Hole in One: A Project-Based Class on Mechatronics

Gabriel Hugh Elkaim
Computer Engineering Dept.
University of California Santa Cruz
Santa Cruz, CA.
Email: elkaim@soe.ucsc.edu

Abstract—We present a 10 week long project-based course on mechatronics that combines lab exercises with an open-ended team project that takes the students from a very basic understanding through development of a small ground robot. Students' basic skill sets are developed using team weekly lab assignments covering software, analog electronics, rapid prototyping and fabrication techniques, and motor drive using a microcontroller. An open ended team project (specifically *Slug Spackler's Revenge* in 2009) gives the students an opportunity to put these skills to use and demonstrate their ability to synthesize the knowledge. A public competition is held at the end to demonstrate these projects, and is very well attended. Student evaluations indicate that this course, though extremely demanding, provides a peak engineering experience that they carry with them well beyond their educational careers.

Keywords-Mechatronics; Project-based learning; Pedagogy;

I. INTRODUCTION

In engineering and computer science, a great deal of theoretical background is required to analyze problems. Unfortunately, by the time the student has achieved enough of that background, they are near the end of their engineering education, and have yet to develop the experience and intuition that comes from attempting to apply their theoretical background to a set of real (or near-real) problems. The Accreditation Board for Engineering and Technology (ABET) has introduced a requirement for a capstone design sequence into engineering programs. While this has greatly helped students to assemble their knowledge into a coherent whole, often it is a case of trying to do too much in too little time with little prospect of success.

What has been missing is a class focusing on electronics, software, and mechanical system design. This area has evolved over the past few decades, and has gone through several names: electro-mechanical systems, systems engineering, embedded devices, and mechatronics, to name only a few. Indeed, the current operating definition of mechatronics comes from a quote [6]: "synergistic integration of fine mechanical engineering with electronics and intelligent computer control in the design of products and manufacturing processes." Several decent textbooks exist; for instance [1], [3], [4], [2] and [5]. More simply, mechatronics is the art of getting a thousand small details right at the same time (even when each detail is itself quite simple).

The author introduced an *introduction to mechatronics* class into the curricula seven years ago to insert a necessary pedagogical step between the theoretical background material and the largely amorphous capstone design. It is aimed at juniors and seniors in engineering, and is filled every year with 30 students. In this class, the focus is on application, rather than theory, and the students are forced to start building very quickly with limited information, incomplete knowledge, and a general lack of practical skills. In addition to 4 hours of weekly lectures, four initial labs (one per week) bring the students up to speed in the areas of software, analog electronics, rapid prototyping, and motor driving. Following a midterm exam, the students are assembled into teams (not of their own choosing) and given an open-ended project to work on for the remainder of the quarter.

The quarter culminates with a public demonstration of their projects that often attracts a very large crowd of fellow engineering students, friends, family, and local media. The project is chosen as a type of game in which the student's 'droids have to compete against each other in two minute rounds. The class is a very intense experience, with students easily putting in more than 20 hours per week outside of lectures. The projects are chosen to be whimsical and fun, and though no portion of the final grade is based on public competition results, student teams ramp up their efforts to compete with each other.

This paper will detail the class taught in 2009, with the final project, *Slug Spackler's Revenge* having a golf theme (hence the title). The paper is organized as follows: background expectations, lab details, project details, student feedback, and conclusions.

II. COURSE ORGANIZATION

The introduction to mechatronics class at UC Santa Cruz (CMPE118/L), has been developed to synthesize knowledge in four key areas: software, electronics, mechanical design, and system integration. In order to develop the skills, intuition, and experience necessary to gain proficiency in these four areas, the students need to be introduced to the lab environment early (and often).

Paradoxically, students need to be taught early on that failure is a virtue. Many students begin the class with incomplete knowledge in many of the key areas. When



Figure 1. Lab 0 Cockroach

faced with a lab, they look for detailed steps to avoid mistakes. Of course, anyone can follow a cookbook, and with enough instructions they might even complete the assignment. However, they will have learned nothing. By leaving them with vague instructions (and hardening the hardware to the inevitable mistakes) the students eventually overcome their trepidation and begin to experiment with the hardware at hand. By failing, they discover what works, what does not, and learn to supplement their knowledge with that of their lab-mates, colleagues, books, and diligent web surfing. Indeed, experience has shown that the more difficult it is to figure out how to do the lab, the better that knowledge is retained.

In order to maximize the learning experience—and to maintain a semblance of schedule integrity towards the end of the 10 week quarter—the class is substantially front loaded to force the students into the lab very soon (often the afternoon of the first day of class). The lab is set up to be open 24/7, with keycards to prevent unauthorized access. Conventional labs at the undergraduate level consist of scheduled lab sessions where the students are led through the lab experiments. With this class, the scheduled sessions merely indicate the presence of the instructor or TA's to answer questions and offer some guidance. A full complement of help is an instructor, one graduate student TA, and two undergraduate tutors who have taken the class before. Students are encouraged to start labs early and to collaborate with their classmates.



Figure 2. IR Beacon

A. Lab 0: The Cockroach

The first lab, started on the first day of class, consists of writing the software to control a small two wheeled ground robot, a.k.a. the “cockroach.” The cockroach (Fig. 1) is a plexiglass robot, designed in-house, that has two brushed DC motors, a pair of 1 Amp H-bridges, four bump sensors located on each corner, and a CdS light sensor. The control is provided by a Freescale 9S12 microcontroller, which is programmed via a serial port. The canonical rules of the cockroach are: run from light, hide in darkness, don't get stuck.

After a series of lectures on event driven programming and state machines, the students are asked to diagram a state machine for their cockroach behavior, and then implement it in teams of two. They are provided with software libraries that abstract away the hardware, providing them primitives such as `checkBumpSensors()` and `driveLeftMotor(speed, dir)`. Thus, even with only primitive knowledge of C and some background in programming, the students are able to make the cockroach achieve its tasks.

Before the students are allowed to program the cockroaches, they complete a series of exercises to set up the programming environment, compile code, and transfer it to one of the microcontrollers. After their initial success with their cockroach, they are asked to load their code onto one of the other nine to demonstrate that their program still achieves full functionality. Most often they fail this test, then learn how to make their code robust to variations in the platforms. Finally, the students go back and embellish their state machines, adding such things as a complex dance to the roach behavior. All of this occurs within the first week of class.

B. Lab 1: Analog Filtering

The second lab (Lab 1) is a whirlwind tour of analog electronics and basic signal conditioning. After covering op-amps, comparators, signal conditioning, and sensors in lecture, the students are given a number of circuits to assemble using a proto-board and discrete components which come as part of their lab kit (\$100). It is assumed that most of the students are already familiar with circuit design and semiconductor devices, but their knowledge is rusty. For those students who have not had this background (CS students, for example), there are a series of four videos provided on the class website that form a crash course in electronics. Students from Earth Science and Biology have successfully navigated the class using only these supplementary lectures along with some dedicated time with the instructor and TAs.

The most difficult part of the lab is building an analog filter that can detect the presence of one of the standard class beacons. These beacons (Fig. 2) are used throughout the class, and output a 2KHz square wave on their IR LEDs. These beacons will be used in the class project, and by



Figure 3. Droid prototyped using SolidWorks and the Laser Cutter

developing the filter stage at this point in the class, each team will have at least two working circuits.

In the lab, the students use the output of a photodarlington, coupled with as many op-amp stages as required, to output a clean high or low signal and light an LED, indicating the presence or absence of the beacon. In general, they must filter out all of the undesired signals (60Hz noise, florescent lighting, daylight, etc.) and amplify the signal of interest. Once basic functionality is achieved, students compete with each other over detection distance.

C. Lab 2: Rapid Prototyping

The third lab (three weeks into a 10 week long class) is to develop students skills at rapid prototyping. Due to the lack of mechanical engineering at UC Santa Cruz, it is rare for the students to have any background in structures or basic beam theory. While most have seen basic forces and moments in their physics classes, they have often not studied these concepts since their freshman year or even high school. A few lectures are dedicated to a very gross understanding of loading and bending and simplified ways of understanding how things will move. Droids from previous years are analyzed to show where they are strong, where they are weak, and how they will bend or fail. Walking the students through these examples seems to be an effective way to make them at least peripherally aware of structural issues.

In the lab, they use SolidWorks to design their parts and assemblies. They also learn how to transfer these designs to a laser cutter to cut parts out of acrylic, foamcore, or MDF. The laser cutter enables very sophisticated structures and designs and makes the mechanical assembly of the droids much easier, as the students can try their designs in software before assembling it by hand. Quite a bit of time is then spent in the lab building some base structures out of foamcore and hot-glue. Lastly, they revisit their filter designs from the previous lab, and solder a working filter on a small

perf-board based on a working design. Students report great satisfaction at this lab, from working with their hands and having something tangible at the end of it. Fig. 3 shows a typical droid resulting from the use of SolidWorks and the laser cutter.

D. Lab 3: Motor Driving

The last of the four labs teaches the students how to drive various motors. This lab reintroduces the microcontroller, this time embedded in a custom board that provides a great deal of robust isolation from the micro to the individual pins. While this has not prevented the students from destroying the microcontroller, it has reduced the rate quite a bit. One of the only remaining vulnerabilities on the board is from physically shorting pins at the chip itself, often while trying to probe a signal with the oscilloscope.



Figure 4. Slug Spackler Teaser

The students, having spent lecture time studying motor drivers, inductive kickback, snubbing, solenoids, stepper motors, and R/C servos, are given a small board capable of driving two DC motors or one stepper. The students start with example code, and modify it to find the limits of the PWM drive for the motors, learning to control speed and direction using the attached

motordrive boards based on Allegro A3949 chip.

Next, the students modify code to drive the stepper motors using the same H-bridge, this time implementing full step, half-step, and wave drive. They then use another daughter board that drives the stepper directly, based on the Allegro A3982 chip, this time driving it using only a direction and step signal. Lastly, they modify another software module to control the RC servo based on a 0.5 - 2.5 msec pulse width.

III. PROJECT: SLUG SPACKLER'S REVENGE

After completing four labs in four weeks, the students are given the project specification and assigned into teams of two or three. Prior experiences with self-selected teams demonstrated that the students tended to choose very lumpy distributions of abilities, based on friendship rather than estimated ability. Using performance on the labs as an indicator, we attempt to "load balance" the teams to have more or less even capability across teams.

Each year, the final project is assigned to combine the skills required as well as some whimsical competition. In keeping with the UC Santa Cruz banana slug mascot, each project has had some form of slug within the description.



Figure 5. Students and Droids at Public Demo

Previous projects have been “The Good, the Bad, and the Slugly” (Duel, 2006), “Joe Banana: the Spy Who Slimed Me,” (Spy, 2007) “Slug Madness” (Basketball, 2008).

In 2009, the project was: “Slug Spackler’s Revenge: Caddyshack Open,” which had a golf theme (see Fig. 4). Specifically, the task was to develop a droid that could place at least two balls (usually ping pong) into three targets on the playing field. The targets were shallow cones with a central tower 12” above the target. Two droids competed on an 8 ft.-square field with a 2 inch wide strip of black tape to mark the boundary. At each 2 minute round, the droids were placed in a random orientation on the field, along with three targets. Each of the three targets carried a 2KHz beacon (the same ones used in Lab 1), which was turned off after two balls were placed in the targets. The full project description can be found in [7].

In order to pass the class, each team had to achieve *minimum specification* demonstrating a functioning droid that could place at least one ball into a minimum of two separate targets, before the public show.

Helping the students along were a series of check-off points that each demonstrated progress toward the goal. In the case of Slug Spackler’s Revenge, these checkoffs were spaced weekly, and included brainstorming designs; time schedules; block diagrams; mechanical schematics; state machine; SolidWorks designs; working sensors and actuators (including schematics); final sensors and actuators (including schematics); autonomous platforms; collision resolution; field boundary detection and resolution; targeting and ball delivery; and minimum spec checkoff.

During the six week duration of the project, the teams put in long hours, working day in and day out, and had many false starts as they discovered what worked and what did not. Several teams went with mobile platforms, while one team (Team Merry-Go-Bot) developed a fixed turret which rolled the balls in. Several different methods were used to put balls into the targets, including dropping, rolling, shooting, and bouncing, each with their own advantages and disadvantages.

In the end, the team droids were as unique as the teams themselves. All demonstrated ingenuity, inexperience, learning, perseverance, and inspiration—and no small amount of perspiration, as well. Even though the public competition

does not count for the class grade, one team completely rebuilt their droid after meeting minimum spec (and passing the class) even though they had not slept in days.

The public competition attracts quite literally hundreds of spectators, along with local media (Fig. 5) and many of the students’ family members travel from all over the state to attend.

IV. STUDENT FEEDBACK

The results of student surveys over the years have been overwhelmingly positive. Though the students report putting well over 18+ hours outside of lecture into the class, they all rated the class overall as either *excellent* (83%) or *very good* (17%). In terms of students’ comments: “BEST. CLASS. EVER. Worked hard, had a lot of fun, made a cool bot.,” “I will recommend this class to every engineering student,” and “Awesome class!”

V. CONCLUSION

In this paper we have presented an Introduction to Mechatronics class, taught at the junior/senior undergraduate level in the computer engineering department at UC Santa Cruz. The course is very intense, with lectures, 4 initial labs then followed by an open ended team project. Student feedback on the course has been overwhelmingly positive, even though they all report putting in excess of 20 hours/week outside of class. Pedagogically, the class succeeds because the students learn by doing. By allowing themselves the freedom to try various approaches, and failing as they explore the limits of their knowledge. Students later in their careers recall the class fondly and rate it as one of their best engineering experiences.

ACKNOWLEDGMENT

The author would like to thank Professor Ed Carryer at Stanford University, whose three quarter graduate sequence was the inspiration for the class described in this paper.

REFERENCES

- [1] D. Bradley, D. Seward, D. Dawson, and S. Burge, *Mechatronics: and the design of intelligent machines and systems*, Stanley Thornes, UK, 2000
- [2] C. W. de Silva, *Mechatronics: A Foundation Course*, CRC Press, FL, 2010
- [3] C. W. de Silva, *Mechatronics: An Integrated Approach*, CRC Press, FL, 2005
- [4] A. Smali and F. Mrad, *Applied Mechatronics*, Oxford University Press, NY, 2008
- [5] J. E. Carryer, M. Ohline, and T. Kenny, *Introduction to Mechatronic Design*, Prentice-Hall, US, 2010
- [6] J. Dinsdale and K. Yamazaki, *Mechatronics and ASICs*, Annals of the CIRP, 1989; (38): 627-634
- [7] *CMPE118 Class Website*: www.soe.ucsc.edu/classes/cmpe118/Winter09/