Piloting a Balanced Curriculum in Electrical Engineering—Introduction to Robotics

Gregory L. Plett and Michael D. Ciletti
Department of Electrical and Computer Engineering
University of Colorado at Colorado Springs

Abstract

Recent papers have reported that engineering students perceive and assimilate academic content in different ways. A variety of theories have been developed to try to understand this phenomenon better so that instructional methods may be developed to reach all students. One well-known instrument used to assess learning styles is the Myers-Briggs Type Indicator (MBTI) [Myers80], which can be used to classify learners according to a Jungian personality typography. Others have reported on the utility of this approach. Since the engineering profession requires that its practitioners function in all types of circumstances, these results underscore the importance of an educational process that provides a *balance* in teaching methods to reach, reinforce, and challenge students of all personality types and learning modalities.

Comprehension of the Kolb elements of learning combined with the 4MAT system [Harb93] provides an instrument to formulate balanced engineering curricula. In Kolb's framework, students' learning styles are projected onto two dimensions: perception, and processing. Based on these two continuums, Kolb enumerated four different types of learner, an understanding of which forms the basis of the 4MAT system, an instructional cycle aimed first at reaching students of all learning types, and secondly at teaching students how to traverse the learning cycle for themselves, preparing them for life-long learning.

The Electrical and Computer Engineering (ECE) Department at the University of Colorado at Colorado Springs (UCCS) has successfully implemented key features of the Kolb/4MAT learning paradigm in a new freshman-level course *Introduction to Robotics*. This paper will describe relevant details of this new course and relate our results to a Kolb/4MAT learning paradigm. Additionally, we will report on efforts to extend the methodology to a core set of courses in our curriculum under the sponsorship of the National Science Foundation.

I. Former Practice at UCCS and the Need for Change

The ECE Department at UCCS offers undergraduate Bachelor of Science in Electrical Engineering and Bachelor of Science in Computer Engineering (BSEE/BSCpE) degree programs. The majority of courses in both programs take a very traditional lecture-based approach to delivery of by-and-large, very traditional content. The four-year undergraduate degrees each comprise eight semesters, with courses offered sequentially to accommodate prerequisites (*e.g.*, calculus and physics). Before the innovation described in this paper, the first year of the curriculum introduced students to (1) mathematical problem solving using the Matlab system, and (2) the C programming language, as well as to calculus and physics courses. In the sophomore year the curriculum builds on the first year's foundation of calculus and physics, and covers analog circuits (*e.g.*, solutions to linear differential equations by classical and Laplace transform methods), solid-state materials, and digital circuits (combinational logic and finite-state machines). Required courses in the junior year of the curriculum introduce concepts in discrete-time systems (*e.g.*, *z*-transforms), and branch into treatment of electromagnetics, solid-state device theory, electronics, and probability/statistics. The balance of courses required to complete the degree consists of laboratories, electives (technical and socio-humanistic), and a capstone senior design project.

Much of this legacy curriculum was designed before the literature documented a proper understanding of learning theory, so our present structure and delivery comprise, to a large degree, traditional lectures and homework assignments. As will be discussed, this is not a balanced approach.

An additional concern about our curriculum was a particular freshman-level course, *Introduction to Engineering Problem Solving*, which was developed for various historical reasons. While the course title sounds promising, the content comprised lectures in Matlab programming. Our freshman students thus learned: Matlab, C, Maple, and Java in their first year, some of which they did not use again until junior year or later (by which time, they had forgotten most of what they initially learned). In our curriculum, Matlab is important, but it is not needed until the junior year when some of its advanced features may be used.

A more general problem was that our students faced two years of math, science, and other fundamentals before they experienced engineering. We view engineering as the practice of problem solving and design under constraints, neither of which was effectively taught until later in the curriculum. Other majors give students an early "feel" for their chosen area of study. We believe that this lack of "feel" in our curriculum was leading to a misunderstanding of what engineering is all about, resulting in attrition.

We decided to look at this problem as an opportunity. We moved the one-semester-hour freshman Matlab course to the junior year,[1] which left an opening with which to do something constructive. We saw this as an opening to excite students with engineering, give them an early flavor of problem solving and design, get them involved with other students, use technology to learn technology and prepare them to design technology. Furthermore, we saw this as an opportunity to pilot a course with balanced pedagogy of the sort to be described.

This paper continues by first describing what we mean by "balance" in the context of the Kolb/4MAT learning paradigm. Then, it proceeds to describe in detail the new freshman course *Introduction to Robotics*, and how this course fits into the Kolb/4MAT framework. We describe some outcomes and ongoing work to renovate our entire curriculum. Finally, we present some concluding remarks.

[1] The new course is offered as a lab co-requisite with our *Linear Systems Theory* course. It covers the same material as the original freshman-level course in only four weeks, due to the more mature nature of the students, and since the theory complements the practice. This leaves eleven weeks for additional material.

II. Learning Theory

Each student perceives and assimilates academic content differently than the others. A variety of theories have been developed to try to understand this phenomenon better so that instructional methods may be developed to reach all students. One well-known instrument used to assess learning styles is the Myers-Briggs Type Indicator (MBTI) [Myers80]. Students are required to complete a survey that categorizes them as either: introverts or extroverts; sensors or intuitors; thinkers or feelers; and judgers or perceivers. The exact definitions of these terms are not critical here besides noting the following: extroverts are energized in groups, and like working in settings that provide activity and teamwork; introverts are energized by solitude, and prefer internal processing; sensors rely heavily on sensory experience like concrete learning experiences; intuitors rely on intuition, and prefer instruction that emphasizes conceptual understanding; thinkers like logically organized presentations; feelers prefer a personal rapport with their instructors; judgers like well-structured instruction; and perceivers like choice and flexibility in their assignments [Felder02]. The engineering profession requires that its practitioners function in all types of circumstances, so the goal of the educational process should then be to provide a *balance* between all of these modalities to reach, reinforce, and challenge all students.

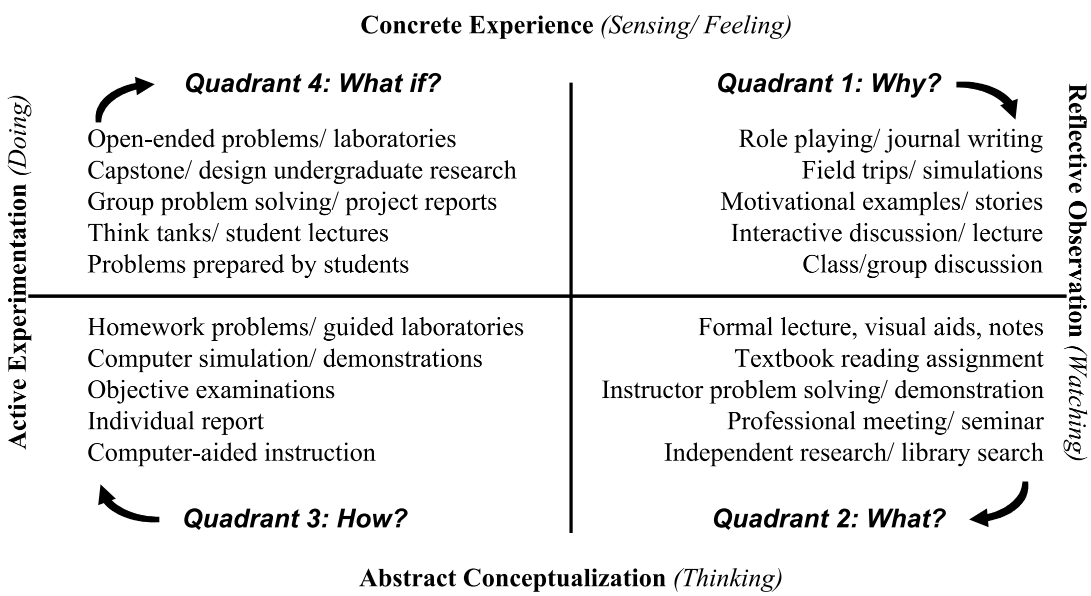

Figure 1: Kolb elements of learning and learning styles with overlaid learning activities and 4MAT learning cycle (arrows); adapted from [Harb93].

The 4MAT system [Harb93] combined with elements of the Kolb learning theory provide an instrument to formulate balanced curricula. A condensed summary is presented in graphical form in Figure 1. In Kolb's framework, students' learning styles are projected onto two dimensions: perception (how a student takes things in), and processing (how a student makes things part of him/herself). Perception may be either concrete or abstract, and processing may be either reflective or active. Based on these two continuums, Kolb enumerated four different types of learner, as identified by the four quadrants in Figure 1. Each quadrant is characterized by a question: quadrant 1 asks the question "Why?"; quadrant 2 asks the question "What?"; quadrant 3 asks "How?"; and quadrant 4 asks "What if?". These four questions form the basis of a learning cycle—the 4MAT system—that an instructional cycle passes through as indicated by the arrows in Figure 1. A curriculum that addresses these four questions can: (1) reach students of all learning types, and (2) teach students how to traverse the learning cycle for themselves, preparing them for life-long learning. Representative teaching/learning activities that stimulate students of each learning style are listed in the appropriate quadrant. In the first three quadrants, the instructor plays the roles of motivator, expert, and coach, respectively. In the fourth quadrant, the instructor serves as a mentor, as the student is fully in charge of learning in this mode.

Our new freshman course, *Introduction to Robotics*, employs instructional aspects from all four quadrants of Figure 1. This course was a tremendous success in its first two offerings in the Fall 2003 and Spring 2004 semesters, and is just as promising this semester. We proceed in this paper by describing the course as implemented.

III. The pilot course: *Introduction to Robotics*

The acts of creating and using technology require proficiency in multiple disciplines. Most generally, technological devices embody complex design choices reflected in their construction, electronics, and programming. Since technology is inescapable in our daily experience, it is valuable for students of all disciplines—and especially those in engineering programs—to have a solid fundamental understanding of these topics. A historic limitation to introducing technological design at an early stage in the student's education has been that significant mathematical and scientific maturity is required before many projects can be contemplated. Recently, however, a number of universities have reported great success using LEGO robotics to teach the basics of engineering to freshman engineering students. The LEGO kits provide a technological medium for hands-on learning of engineering design and problem solving without requiring college-level knowledge of mathematics or the sciences. Supported by a grant from the UCCS Teaching and Learning Center, we have together designed and implemented a new freshman course *Introduction to Robotics*. We co-developed this course, and co-teach it. It has an on-line course reader, an on-line integrated set of laboratory exercises (with pre-lab assignments), and a comprehensive final design project where students must generalize from their lecture and lab experiences to use technology to solve a design problem.

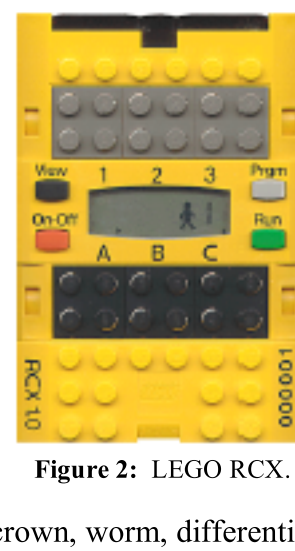

Figure 2: LEGO RCX.

Why Robotics? We chose robotics as a versatile pedagogical tool—letting students use technology to learn about and understand technology. A robot's dynamic nature provides immediate feedback as to whether it is accomplishing its task; robots are understandable to students in all disciplines, of all backgrounds; their design spans the mechanical, electronic, and software fields, and they relate to everyday experience. A significant advantage of using the LEGO robotics approach, in particular, is that we are able to teach fundamental technological concepts in a hands-on way, without requiring a high level of mathematical and scientific maturity. The LEGO kit is marketed to the public as the "LEGO MINDSTORMS Robotic Invention System." It includes a programmable LEGO "brick" (called the "RCX", shown in Figure 2), which can operate motors, lights, and other devices, and can sense its surroundings with touch sensors, light sensors, and rotation sensors. It includes gears (spur, bevel, crown, worm, differential, rack), pulleys, a clutch, axles, wheels, beams and other parts so that one may quickly construct very elaborate robots. Anyone with an interest in technology can take this course and learn something about the fundamentals of technology-based systems and their design. We encourage participation from students in all colleges on campus by making this course open to everyone—there are no prerequisites. Each student can then benefit from all of the advantages of interdisciplinary team participation.

Lecture Period			Hands-on Lab Period		
1.	Getting started	8/23/04	1.	Nobot	8/25/04
2.	The RCX	8/30/04	2.	Tankbot	9/1/04
	[Labor day holiday]	9/6/04	3.	Bumpbot	9/8/04
3.	Introduction to NQC	9/13/04	4.	Bugbot	9/15/04
4.	Intro. to NQC (cont)	9/20/04	5.	Linebot	9/22/04
5.	Robot construction	9/27/04	6.	Scanbot	9/29/04
6.	Robot construction (cont)	10/4/04	7.	Steerbot	10/6/04
7.	Basic control	10/11/04	8.	Diffbot	10/13/04
8.	Basic control (cont)	10/18/04	9.	*Quiz on NQC*. Work on project	10/20/04
9.	Basic electronics	10/25/04	10.	*Quiz on construction*. Project	10/27/04
10.	Basic sensors	11/1/04	11.	*Quiz on control*. Project	11/3/04
11.	Basic sensors (cont)	11/8/04	12.	*Quiz on electronics*. Project	11/10/04
12.	Microprocessor designs	11/15/04	13.	*Quiz on sensors*. Project	11/17/04
13.	Microprocessors (cont)	11/22/04		[Thanksgiving holiday]	11/24/04
14.	Cybernetics	11/29/04	14.	*Quiz on microprocessors*. Project	12/1/04
15.	Robot qualification trials	12/6/04	15.	Final competition (8:00am)	12/10/04

Figure 3: Course syllabus for *Introduction to Robotics*, Fall 2004 semester.

For our college/department, the approach we have taken to this course is unique. We have structured the material to be delivered in a partitioned way: one half is lecture-based, and one half is lab/project based. (See Figure 3 for a syllabus). The class meets in lecture on Mondays, where we teach the knowledge base. There are eight major units: (1) Introduction to the capabilities of the LEGO programmable brick; (2) Introduction to programming in the NQC ("not quite C") language; (3) Robot construction; (4) Basic control systems; (5) Basic electronics; (6) Robot sensors; (7) Microprocessor designs; and (8) Cybernetics. The main lecture units (2–7) comprise the following content:

> NQC: How to write programs using structured, procedural programming methods; correctly use program control structures to execute loops and test conditions; use multitasking to simplify program design; use global and local variables and arrays in programs; sense environmental variables with interrupt-driven processing. Note that many of our students have never programmed before.
>
> Construction: How to build robotic structures that are robust to typical abuse; design gear systems with the correct understanding of the relationship between speed, torque and power; use a wide variety of mechanical parts: spur, worm, bevel, crown, rack and differential gears, clutches and pulleys. (Furthermore, understanding the design tradeoffs between using gears versus pulley systems.)
>
> Control: Design of low-level control systems for robots that incorporate sensor feedback, using on/off, proportional, proportional-derivative, and proportional-integral-derivative

control methods; design of high-level control systems with the proper understanding of when to use sequential versus prioritized behavioral control.

Electronics: Topics in circuits and electronics to understand how robots function internally, including the basic concepts of: charge, voltage, current and resistance; voltage divider circuits, diodes and transistors.

Sensors: Exploring how sensors are designed, including binary versus analog sensing, analog-to-digital conversion, light, position, temperature and current sensing with a wide variety of technologies (*e.g.*, sensors that detect position include: tilt, bend, magnetic, reflectance, ultrasound, shaft-encoding and GPS sensors);

Microprocessors: Introducing an understanding of the operation of microprocessors and microcontrollers, including the basic structure of a microprocessor circuit (instruction fetch, decode, execute, ALU, registers), the differences between machine, assembly and high-level languages, and the process by which one is converted to another, and the functional units of a typical microcontroller chip: RAM, digital input/output, analog input/output, timers, pulse-width-modulators, serial interface, and interrupts.

The curriculum is very aggressive. By the end of the semester the students will have designed, implemented, tested and validated robots incorporating sequential, behavioral, and proportional feedback control; they will have written their own programs using a large subset of the widely-used "C" language, incorporating variables, arrays, multitasking and resource management; they will know at the electronic-component level how robot sensors work and what an H-bridge circuit does; and will understand the process by which a user program gets translated into an assembly-language program, then to machine language, and how the machine-language program executes on a microprocessor. Six quizzes (on topics 2 through 7) evaluate their comprehension.

One important focus of the lectures is to point out that there is more to technology than software. We take a systems view, showing that technology is everything from the batteries that power the robot, to the resistors and capacitors and integrated circuits that cause the microprocessor to operate correctly, to the actuators and sensors that allow the robot to interact with its environment, to the very mechanisms that give the robot form and movement. To support the lectures and lab, we use web-based lecture and lab readers that are updated throughout the semester with new and corrected information.

The class meets in the lab on Wednesdays, for the hands-on component of the class. The first eight laboratory assignments are designed to convey significant content and to develop building, programming, debugging and documentation skills in a progressive way. The first assignment requires that the

	Three-member team			Two-member team		
Lab	Builder	Coder	Scribe	Builder	Coder	Scribe
Nobot	M1	M2	M3	M1	M2	M1
Tankbot	M2	M3	M1	M2	M1	M2
Bumpbot	M3	M1	M2	M1	M2	M1
Bugbot	M1	M2	M3	M2	M1	M2
Linebot	M2	M3	M1	M1	M2	M3
Scanbot	M3	M1	M2	M2	M1	M2
Steerbot	M1	M2	M3	M1	M2	M1
Diffbot	M2	M3	M1	M2	M1	M2

Figure 4: Responsibility rotation schedule for the eight preliminary labs.

students write a simple program to turn a motor on, and to reverse its direction each time a button is pushed. From there, the designs become increasingly involved, and incorporate complex hardware design (*e.g.,* rack and pinion steering, tank-tread, single- and dual-differential drive) and complex software (*e.g.,* sequential, procedural and multi-tasking programming). The labs are accomplished by three-member student teams, where the three prime responsibilities of (1) building the robot, (2) programming the robot, and (3) documenting the lab with a formal lab report are distributed by rotating them among the members each week (see Figure 4 for a schedule). The lab experience seems to us to have been highly successful. All groups have finished every lab assignment on time and have successfully demonstrated a robot with the correct operation. Students have been able to experiment with using hardware and software together, using a microprocessor coding environment similar to that used in industry, they have developed approaches to debugging problems with computer-based technology, and have discovered that technology requires systematic testing and verification. They have participated in the embedded programming experience, having written software on one computer platform, cross-compiled it, downloaded it to another platform, and then executed it.

The eight labs have been adapted from the book that we have adopted as our text [Baum03]. We present a short description of each lab here. Figure 5 shows drawings of each robot as well.

> Nobot: Nobot connects the RCX to both a motor and a touch switch. When the program is executed, the motor begins rotating. Every time the touch switch is pressed, the motor toggles its direction. While this is a trivial exercise, it accomplishes several key objectives: (1) The constructed system satisfies all the criteria for being a robot (*i.e.*, it has hardware, software, an actuator, and a sensor), so is the first robot that most students will have ever created (although it doesn't appear much like what the students expect a robot to look like!), and (2) the students gain basic familiarity with the LEGO MINDSTORMS Robotics Invention System and the NQC programming language by (a) writing a simple program; (b) downloading the program to the robot; and (c) using the program to read a switch and control a motor.
>
> Tankbot: Tankbot is a tank-like robot that drives forward for a while, turns on the spot, and repeats. Students learn two ways of steering a robot, and the difficulty of driving straight due to mis-matched motors, friction, or environmental factors. They are also introduced to several programming constructs and good programming style.
>
> Bumpbot: Bumpbot is another tank-like robot that drives around randomly without getting stuck against a wall or in a corner, due to touch-sensor feedback from the front bumper mechanism. The students build two different types of bump sensors and learn the mechanical advantages of each. They also write programs that use sensors to accomplish some real task.
>
> Bugbot: Bugbot navigates around obstacles more effectively than any robot built previously. It uses two touch sensors, each attached to a feeler, to detect the side on which a collision occurred, and responds by driving away from the collision. This is the first robot that uses multiple sensors to control robot behavior. It also introduces students to

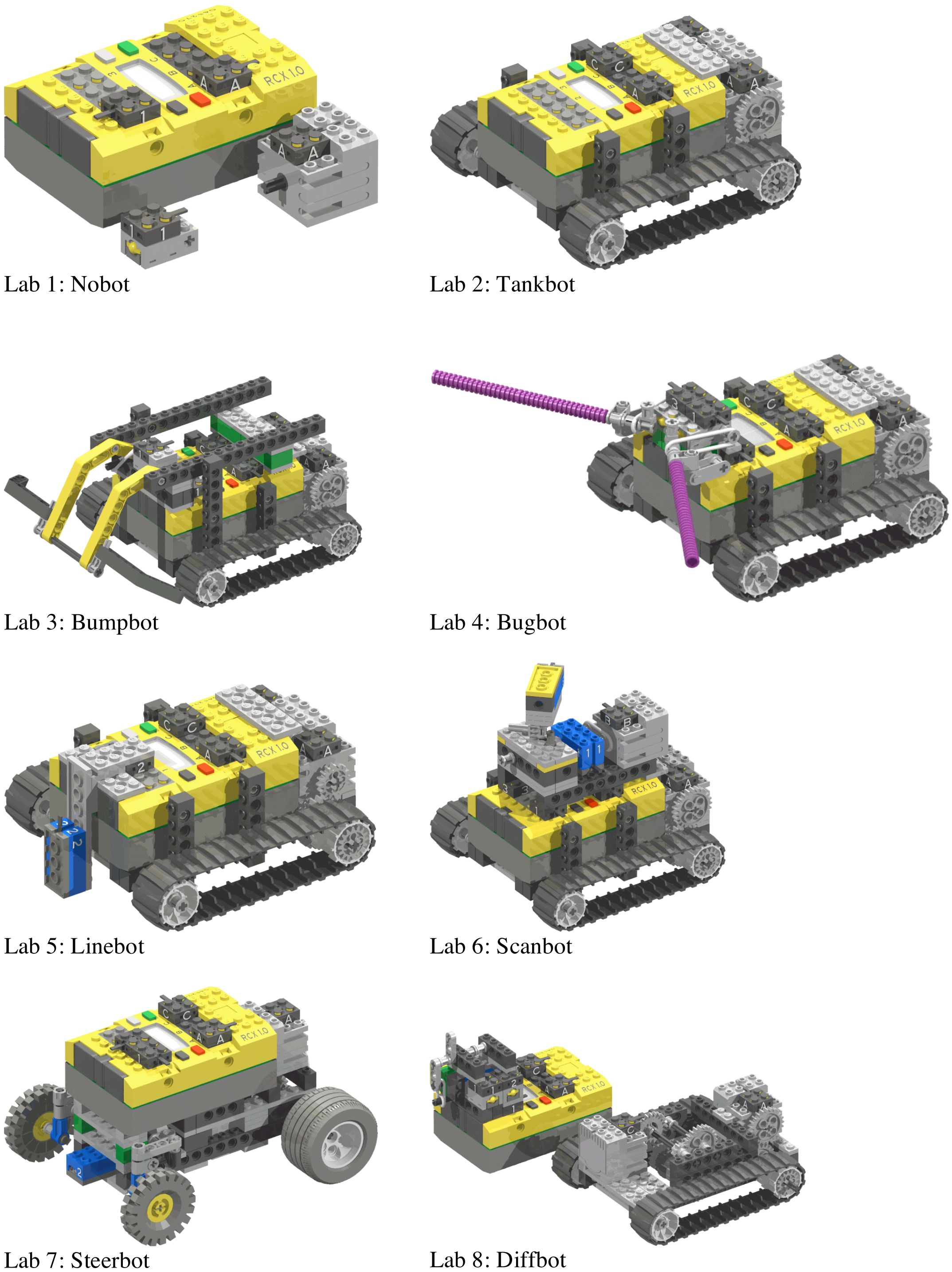

Figure 5: Robots built in the first eight laboratory exercises.

multi-tasking programming, including resource conflicts. Thirdly, students write a program robust to unexpected inputs, which is a critical ability to do well in the final project.

Linebot: Linebot uses a reflected-light sensor to follow a gray line on the floor and stop where the line becomes black. This robot is an introduction to feedback control algorithms, (on-off control) specifically for line following. Students also learn how to implement self-calibrating sensors in environments with different ambient conditions.

Scanbot: Scanbot repeats a process of sweeping a light sensor back and forth, searching for brightest light, and then turning toward the light and driving forward. Students learn dis/advantages of gear drive versus pulley drive, and see some benefits to procedural programming. The control system is a proportional feedback controller using a rotation sensor to detect how far to turn.

Steerbot: Steerbot is a car-like robot that drives around, and avoids obstacles without hitting them using IR transmitted light and a reflectance light sensor. Students learn about differential drive and rack-and-pinion steering. They experience the tradeoffs of sequential versus concurrent programming, and experiment with server-client programming.

Diffbot: Diffbot is a remote-controlled robot that uses two differentials to cause a tread-based design to drive (exactly) straight (even with mis-matched motors and treads), but still have the ability to turn. Students also use the light sensor to distinguish between multiple colors.

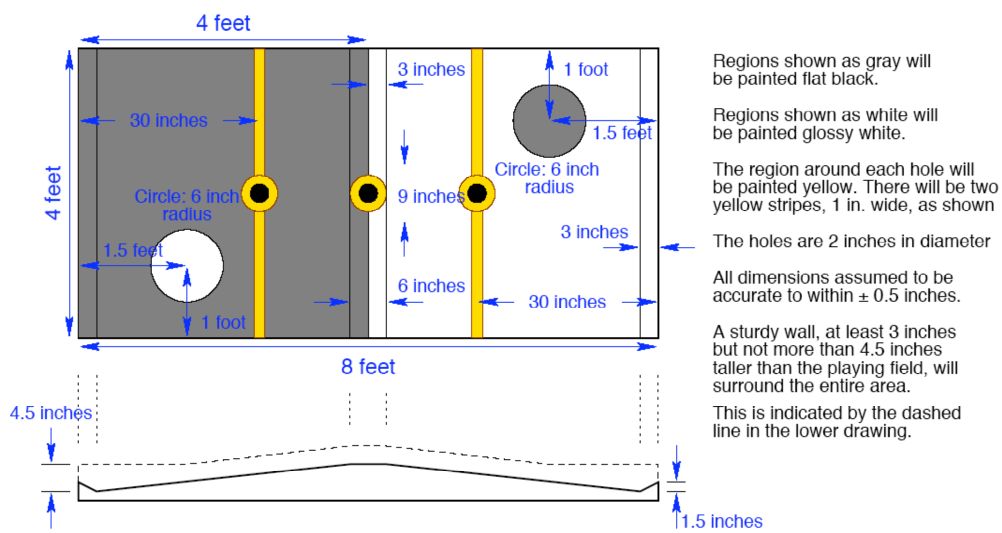

Figure 6. Plan of final contest playing surface.

The course culminates in a competition-based final design project coined "Robo-Challenge". This semester's design challenge is for each team to build and program a robot that will compete against robots from other teams in a game of "robot golf". In a two-minute period, the robots will attempt to collect golf balls positioned on the playing surface (see Figure 6) and place them in holes (2 points for putting it in the hole on the slanted surface on the side the robot started on, three points for the center hole, and –2 points for the hole on the opponent's slanted surface). At the end of two minutes, the robot that has received the most points wins. The specifications for the robot are intentionally very open-ended. All mechanical and software design must be accomplished by the students. Seven class periods are allocated to design, implementation, and testing of robots for this project. Each semester, the students have been highly motivated by Robo-Challenge. They bring to class projects of their own creation, and, they discuss strategies that they have thought about and tested. In short, this course has captured their imagination.

IV. What about Kolb/4MAT?

Our approach to *Introduction to Robotics* addresses all four quadrants of the 4MAT method illustrated by Figure 1. Motivational examples, stories, and interactive discussions (Quadrant 1) serve to stimulate interest in robotics; our formal lectures, reading assignments, and demonstrations (Quadrant 2) provide a base of knowledge to support the laboratory work in Quadrant 3, where a guided series of progressively more difficult robot projects unfolds over eight weeks. Quizzes are administered to encourage study and evaluate progress. The first three quadrants of the 4MAT cycle set the stage for the last, a seven-week self-guided experience in which our students engage in an open-ended design project requiring them to develop a conceptual approach and design a robot to compete against other robots while adhering to constraints that limit the resources that can be used. Thus, this course takes our students through a complete cycle of the 4MAT experience.

V. Results

We are very pleased with the instructional outcomes of this course. *Introduction to Robotics* is a Freshman-level course, designed to give an introduction to the field of engineering, and with special focus on the Electrical and Computer Engineering disciplines. This overall goal has been met and exceeded beyond our expectations. Students with no background beyond high-school math and the ability to read English are: writing programs in a high-level language to interact with the environment, designing and building robotic structures to accomplish some task, designing feedback control systems for their robots, learning about electronics and the design and operation of sensors, digging deeper to understand the operation of microprocessors and microcontrollers, including the basic structure of a microprocessor circuit. Furthermore, they are cooperating in inter-disciplinary teams (fewer than half the students currently registered are declared as ECE majors) to create unique designs, where each member takes turns as the “builder”, “coder” and “documenter”, working with technology to understand technology, where the design tools are similar to those used in industry for “real” projects, and the steps to design are identical, and learning how to write a proper laboratory report and final project report, where correct grammar and spelling usage are required.

The breadth of coverage implemented in this course has by necessity required that the depth of coverage be limited. This decision was intentional, however, and we frequently call attention to the fact that students need more in-depth knowledge to pursue any given topic further. Specific math, physics, and engineering courses are highlighted as being the continuation of each topic to motivate the student to further study.

We have now offered this course for three semesters, and to date the student response has been very favorable. Of the eight mandatory lab assignments, all students (in all semesters) successfully completed the prelab assignments (perfectly, although some students required a few iterations on some problems), all groups successfully built, programmed and demonstrated all robots, and all groups completed lab-report write-ups. All groups designed robots that qualified for the final competition, and many students have discovered that there is more to design than just “playing”. We have informal feedback from the registrar’s office that retention has been improved for students who have ‘graduated’ from it in the past versus students who have not taken the class. We attribute this to the highly motivational nature of this course, and the fact that ECE majors get a "feel" for their subject area early on. We will continue to investigate whether retention is increased, and by how much.

In the first offering, we surveyed the class to ascertain the students' perception re. how the class improved their technological understanding and team-participation skills. A majority of the class responded, and the results are summarized in Figure 6. The data shows significant improvement in technical knowledge (*e.g.*, programming, robotic structures, control systems and sensors) and moderate improvement in non-technical components of this course (*e.g.*, cooperation in interdisciplinary teams). These results exceed our expectations.

VI. Partnerships and Resources

Nearly three years ago, the two of us caught a vision for how to create a Freshman course that would be an exciting, motivational introduction to engineering. We formed a partnership and presented our ideas to the Electrical and Computer Engineering Department faculty, who approved our proceeding to develop this course. In the process, we applied for and received a grant from the UCCS Teaching and Learning Center to help develop "Hands-On Engineering Design with LEGO Robotics", which was critical for us to be able to do a good job with an initial offering of this course. The Department paid for the first set of LEGO kits, and we subsequently applied for a UCCS Instructional Fee grant to fund more kits to open this class up to more students from around campus—this proposal was also successful. Through our partnership, we have co-developed all materials for this course (lecture reader, lab reader, lab solutions, quizzes, quiz solutions, Robo-Challenge specifications, inventory procedures), and we have co-taught the course. This has been a very rewarding opportunity for both of us. We continue to promote this course: We have recently given guest lectures to prospective engineering students ("Engineering Student for a Day"), we are preparing advertising flyers for the upcoming final robot competition, and intend to invite students from high-schools around Colorado Springs to also attend. We have also given a talk for the Teaching and Learning Center that describes how to teach a course with hands-on components to increase learning.

VII. The Future of this Course and the ECE Curriculum

Developing this new course in a polished form in a very short period of time has been a challenge for the two of us, but very worthwhile. Certainly, this course will continue and evolve over the foreseeable future. But beyond this, the idea of using the Kolb/4MAT system and especially integrating hands-on components into a course has been tremendously successful, and we intend to restructure further courses in the ECE curriculum to reflect the Kolb/4MAT paradigm. The technological media of robotics has also been well suited to our aims, and we are considering how we might introduce new courses at the Sophomore, Junior, Senior and graduate levels that would follow the Freshman course in greater depth with more and more difficult problems.

Together with other faculty in our department, we have recently been awarded a grant from the National Science Foundation to redesign the systems area of our curriculum, 19 credit hours of courses, (two courses in circuit theory, two courses in electronic circuit design; a linear systems theory course, and their companion non-integrated laboratories) with the Kolb/4MAT system.

We are proposing a radical redesign of these core courses from the top down to introduce proper (*i.e.,* Kolb/ 4MAT-based) balance, with the long-range goal of this redesign spreading to the whole curriculum. That is, we propose to redesign our teaching methodology to include aspects of all four quadrants of the Kolb learning system in regular cycles as prescribed by the 4MAT system. By redesigning the entire systems-area sequence of courses at once, we will have flexibility with respect to when and how material is presented, and we will avoid accidental duplication of coverage. These courses are, by in large, taught only by the investigators on this project, so there will be no bureaucratic difficulties in implementing the innovations.

ACKNOWLEDGMENT

This work was supported in part by grants from the University of Colorado at Colorado Springs' Teaching and Learning Center, and from the UCCS student technology fee. Ongoing work is being funded by the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the University of Colorado at Colorado Springs, or of the Teaching and Learning Center, or of the National Science Foundation.

Bibliography

[Baum03] D. Baum, *Definitive Guide to LEGO Mindstorms*, Apress, 2003.

[Harb93] J. N. Harb, S. O. Durrant, and R. E. Terry, "Use of the Kolb Learning Cycle and the 4MAT System in Engineering Education," *Journal of Engineering Education*, Vol. 82, No. 2, April 1993, pp. 70–77.

[Myers80] I. B. Myers and P. B. Myers, *Gifts Differing*, Consulting Psychologists Press, Palo Alto, CA, 1980.

MICHAEL D. CILETTI
Michael Ciletti received the BS, MS, and PhD degrees in Electrical Engineering from the University of Notre Dame. He has taught at the University of Colorado at Colorado Springs since 1974, and is presently Professor in the Department of Electrical and Computer Engineering. He has written five textbooks in the areas of circuit analysis and VLSI circuit design and synthesis with hardware description languages (Verilog and VHDL). His overarching goal in teaching is to be clear, translating complex concepts into understandable building blocks for students, with heavy reliance on graphical presentations and illustrations. His mantra for students: "Simplify, clarify, and verify." He can be reached by email at ciletti@eas.uccs.edu.

GREGORY L. PLETT
Gregory Plett was born in Ottawa Canada in 1968. He received the B.Eng. degree (with high distinction) in Computer Systems Engineering from Carleton University in 1990. He received the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1992 and 1998. Since 1998, he has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Colorado at Colorado Springs. He can be reached by email at glp@eas.uccs.edu.

Appendix A. Survey & results

Department of Electrical and Computer Engineering
Mid-Semester Evaluation of ECE 1001, Introduction to Robotics

Answer Scale: None = 0, 1 = Low, 2 = Moderate, 3 = High

1. Indicate your level of experience before and after taking this course in **writing computer programs** for real-time execution requiring interaction with the host machine's environment.

Before	After
______ None	______ None
______ Low	______ Low
______ Moderate	______ Moderate
______ High	______ High

Average Before	**Average After**
1.1 = Low$^+$	1.9 = Moderate$^-$

Improvement = +73%

2. Indicate your level of experience **building robotic structures** before and after taking this course.

Before	After
______ None	______ None
______ Low	______ Low
______ Moderate	______ Moderate
______ High	______ High

Average Before	**Average After**
0.8 = Low$^-$	2.1 = Moderate$^+$

Improvement = 163%

3. Indicate your level of experience in **designing low-level feedback control systems** before and after taking this course.

Before	After
______ None	______ None
______ Low	______ Low
______ Moderate	______ Moderate
______ High	______ High

Average Before	**Average After**
0.5 = None$^+$	1.5 = Low$^+$

Improvement = 200%

4. Indicate your level of **knowledge about electronics** before and after taking this course.

Before	After
______ None	______ None
______ Low	______ Low
______ Moderate	______ Moderate
______ High	______ High

Average Before	**Average After**
1.3 = Low$^+$	1.8 = Moderate$^-$

Improvement = 38%

5. Indicate your level of **knowledge about sensors** and your experience using them before and after taking this course.

Before	After
______ None	______ None
______ Low	______ Low
______ Moderate	______ Moderate
______ High	______ High

Average Before	**Average After**
0.9 = Low$^-$	1.8 = Moderate$^-$

Improvement = 100%

6. Indicate your level of experience in **cooperating in inter-disciplinary teams** before and after taking this course.

Before	After	Average Before	Average After
______ None	______ None	1.8 = Moderate^{-}	2.2 = Moderate^{+}
______ Low	______ Low		
______ Moderate	______ Moderate	**Improvement = 22%**	
______ High	______ High		

7. Indicate your level of experience in **working hands-on with technology** to learn about technology before and after taking this course.

Before	After	Average Before	Average After
______ None	______ None	1.7 = Moderate^{-}	2.3 = Moderate^{+}
______ Low	______ Low		
______ Moderate	______ Moderate	**Improvement = 35%**	
______ High	______ High		

8. Indicate your level of experience in **writing proper laboratory reports** before and after taking this course.

Before	After	Average Before	Average After
______ None	______ None	1.7 = Moderate^{-}	2.4 = Moderate^{+}
______ Low	______ Low		
______ Moderate	______ Moderate	**Improvement = 41%**	
______ High	______ High		