Learning Embedded and Real-Time Systems via Low-Cost Mobile Robotics

Paul G. Flikkema
Electrical Engineering Department
Northern Arizona University

Abstract

The importance and impact of embedded and real-time computing systems on today's society far exceed that of traditional stand-alone computers; it is hard to think of new device or system that does not embed some combination of silicon-based intelligence, sensing, and communication. A parallel trend is the growth of high-level, abstract design of these systems and associated languages and CAD tools, driven by increasing processor speed, time-to-market requirements and the complexity of applications. However, many courses in embedded systems focus on low-level issues such as addressing, interrupts and interfacing. In this paper, we describe a new direction: a course with the goal of motivating students to learn the abstract concepts that underly the design of these systems via experiments that require the interaction of robots with the physical world. In this studio-format course, most conventional learning takes place outside the class, while small student teams design, build, and evaluate autonomous mobile robots in the classroom/laboratory. To keep costs down, mobile robots are created using LEGO parts and programmed in the high-level NQC language using the Robolab RCX microcontroller module. As the semester proceeds, students tackle an array of interrelated problems that motivate the study of sensor signal processing, control, scheduling, and resource sharing. In a final project, the students tackle a distributed intelligence project in which an odometry-equipped robot communicates with a PC-based program that tracks the robot's position. To encourage adoption by other electrical engineering and computer engineering programs, a detailed description of the required resources and their cost is included.

Introduction

There is no doubt that developments in microelectronics and computing technology in the last half of this century have changed modern life dramatically. But without equally dramatic improvements in understanding of underlying systems disciplines, among them digital and computer systems, communications and control systems, signal processing, and machine intelligence, we would not have most—if not all—the advances in transportation, consumer products, medical technology, and military capability that we take for granted.

These advances in the systems disciplines, with their foundation on mathematics and statistics, provide a great challenge for engineering educators: We are now in the position of graduating students at the bachelor's level who are largely ignorant of the powerful concepts that have made possible

- high-definition medical signal processing, imaging and remote sensing,
- high-speed digital communications and data storage/retrieval, and
- intelligent, real-time control of everything from automatic dishwashers to automobile suspensions to supersonic flight,

to name just a few applications.

A good example is today's digital cellular phone. Packed into this shirt pocket-sized device is an array of technologies that are barely touched upon in most undergraduate curricula. It is not difficult to think of other devices that use a host of other technologies such as dynamic programming, optimal control, and neural networks.

Though it might be convenient (at least for faculty) to attempt to simply cover more technical material in the curriculum, we believe that updating the learning process is equally important. Several years of experience in teaching communications and signal processing at both undergraduate and graduate levels, as well as discussions with colleagues, have provided the author convincing evidence that engineering educators should make a major effort to show how abstract systems principles can be applied to real-world problems. This leads to a more general and more important concern: students cannot use these concepts effectively unless they can move beyond mere comprehension and develop the higher-order skills of analysis, synthesis and evaluation.

These concerns have driven the development of a new laboratory and a senior-level electrical engineering course: EE 440 Embedded Control. *Three major goals of the course* are to

- *spark students' interest by connecting exciting real-world problems to abstract concepts,*
- *show students how seemingly disparate and abstract systems disciplines such as signal processing, communications and embedded systems design can be joined to attack important problems, and*
- *help students develop the cognitive skills that allow them to use systems principles in the development of new technologies and applications of them.*

More generally, we hope that this course illuminates other deep connections linking many aspects of modern systems design, such as

- embedded systems design for control and communication;

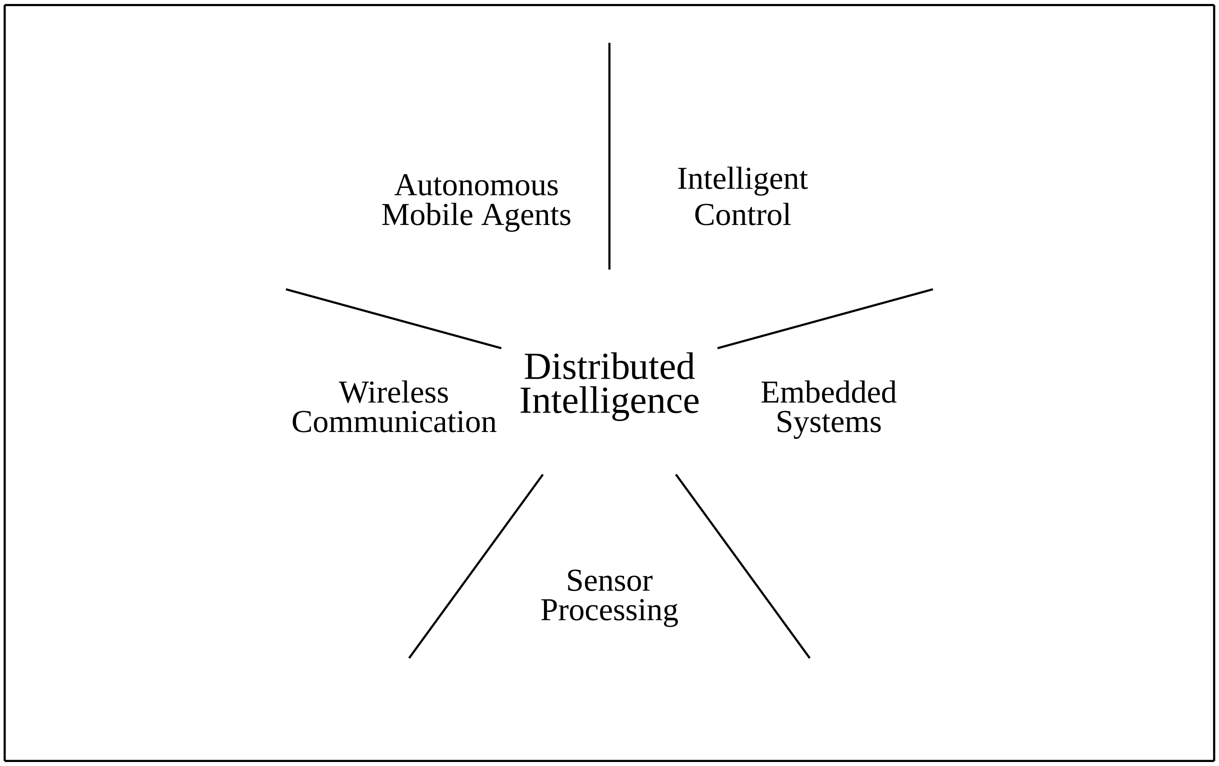

Figure 1: Enabling technologies for distributed intelligence. Skills in these technologies are critical for tomorrow's EE/CSE/CS system engineers.

- how systems concepts must be linked to hardware and software for successful design;
- the importance of high-level approaches in embedded systems and software design;
- the power of layering, i.e., hierarchical organization of systems ranging from computer networks to robot architectures; and

The overall technical theme of this course is the convergence of intelligence—both computational and control—and communication. These areas, along with embedded systems and sensor processing, are critical to the development of the recently-coined technology of *distributed intelligence*, as shown in Figure 1. Future engineers should be well-versed in these areas so that they will be effective contributors to tomorrow's multi-disciplinary engineering projects.

A key aspect of the approach is motivating student teams by guiding them as they design and prototype algorithms in the application domain of mobile robotics. This application serves as an inspirational workbench for students to learn the concepts and tools of modern systems engineering.

Knowledge	Factual, methodological, theoretical
Comprehension	Interpretation, extraction (e.g., of data trends), translation (e.g., from math to text)
Application	Prediction (e.g., based on change of parameter), abstraction
Analysis	Ability to distinguish assumptions, hypotheses, and facts
Synthesis	Communication skills, project planning
Evaluation	Develop criteria and evaluate with respect to them

Table 1: Summary of Bloom's cognitive-domain taxonomy.

Objectives and Pedagogy

Perhaps the best starting point for educational objectives is Bloom's cognitive-domain taxonomy [1], briefly summarized in Table 1. The last three (analysis, synthesis, and evaluation) are recognized as higher-order skills that need to be developed as part of undergraduate education. Because of their importance in engineering, we add to this list **Design** and **Teaming** [2]. (Later we refer to this list as the 'extended Bloom taxonomy'.)

Our method for accomplishing these objectives is the integration of active learning and Kolb's experiential model [3]. The latter has been used successfully in engineering education reform [4] to encourage development of the skills comprising Bloom's taxonomy. It is characterized by learning via case studies, reflective observation, active experimentation, and abstract conceptualization. Active learning (see, e.g., [5]), is characterized by active student participation (instead of passive listening) and higher-order thinking skills. Our approach also can be categorized as problem-based learning [6] (see also [7] and associated web links) since students are given problems that motivate learning.

This approach captures the Kolb model via three mechanisms. First, we use real-world examples (such as the Mars Pathfinder mobile robot) that serve as brief case studies. Second, reflective observation is encouraged in both group and individual settings. Third, team-based laboratory exercises promote active experimentation and abstract conceptualization—and their interplay—by integrating modeling, analysis, and design.

In the lab exercises, students are first asked to individually formulate methods to attack a problem, and refine them with their team into one approach. This allows students to test their ideas in the small group rather than the full class, increasing their confidence. Throughout the course, projects emphasize teaming and higher-order skills associated with design, such as planning an approach to solving the problem and establishing criteria for evaluating solutions.

Course Development and Content

There is little question that embedded software is important in a broad variety of products and systems. For example, embedded microcontrollers in a typical new automobile now number in the dozens. This has been recognized in recent curriculum development efforts, such as the VESL project at Michigan State University [8], which focuses in part on embedded systems software development for everyday applications. Our approach also gives students students a sound foundation in embedded systems programming, but within the context of control in robotics. We believe that the ability to program physical robots to accomplish tasks will close the gap between concepts and physical reality, and be more rewarding to students than projects that can never be integrated into the target system. A second multi-disciplinary thread of the course is the notion of real-time systems, again applied to robotics. Finally, the course is taught so that students will start contemplating the notions of machine learning and intelligence.

All student teams build their robots using standard LEGO parts, including the LEGO ROBOLab programmable RCX that is based on a Hitachi microcontroller. The modularity of LEGO construction and the rich array of available LEGO resources [9, 10, 11] allow students to quickly construct capable wheeled robot platforms. All software design is performed at a high level using the NQC (Not Quite C) language. Therefore, unlike in many courses on embedded systems that focus on low-level issues such as interrupts and I/O handling, we are able to concentrate on interaction with the physical world, including sensory and motion control issues. While doing so, students get the needed exposure to high-level embedded and real-time systems design [12].

Review of the proposed course outline (Figure 2) shows it to be quite ambitious. However, we have taken two steps to improve student productivity. First, the course includes an existing microprocessors course with lab as a prerequisite, so students are well-prepared in the basics of microcontroller-based systems. Second, all programming is be performed at a high level, using either the NQC language or MATLAB Stateflow (a CAD tool for finite-state machine design). This allows students to focus on algorithm design rather than details of an assembly language implementation.

The studio-format course (all activities take place in the lab) incorporates four robot design projects:

trackbot - Each team will build and program a tracked robot to follow a path specified by the instructor.

wanderbot - Each team will build and program a wheeled robot that wanders randomly in a fixed area; the robot must sense the edge of the area using an infrared sensor.

searchbot - Each team will upgrade the wanderbot to find and measure objects in a fixed area.

Week	Learning Focus	Laboratory Activity
1	Embedded systems design	trackbot project
2	Design example	”
3	Robot architectures [13, 14]	trackbot project – demos
4	Processor architecture	wanderbot project
5	Input and output	”
6	Sensors and effectors	wanderbot project – demos
7	Software design	searchbot project
8	Multitasking	searchbot project – demos
9	Navigation and odometry [15]	
10	Real-time operating systems	awarebot project
11-12	Scheduling	”
13	Interprocess Communication	”
14		aware project – demos

Table 2: Course Outline.

awarebot - Each team's searchbot will be augmented with position-monitoring capability and its position and track will be recorded and displayed in real-time by a PC-based program.

A large number of well-designed project-oriented laboratories wither after 2-4 years as students learn via the grapevine a few standard approaches that work reasonably well. To avoid this problem, we will alternate these projects with other challenging problems such as (1) robot races and (2) a robot connected to a trailer that can maneuver on a course that requires backing up.

Laboratory and Equipment

We have established a laboratory consisting of six design stations so that up to 18 students can be accommodated in one course section. Each station is equipped with:

1. One Pentium III PC workstation: these host the software design and development software, and are connected to the Northern Arizona University College of Engineering and Technology LAN for access to web-based information.

2. One LEGO ROBOLab Team Challenge Set: These kits allow student teams to construct and program wheeled mobile robots.

In addition, various software packages are required. This includes the NQC cross compiler and high-level MATLAB design tools, including Stateflow. The NQC software comes bundled with

#	Description	at	
6	Computers	$1500	$9,000
7	ROBOLab Team Challenge Sets	$225	$1,575
14	LEGO rotation sensors	$15	$210
	Total		**$10,785**

Table 3: List of equipment and budget for a six-workstation laboratory (spares are included).

one of the required books for the course [10]. This is a clear benefit to students, since they can develop familiarity with the development environment and develop/refine their algorithms outside of the laboratory.

The budget for the laboratory is presented in Table 3. The cost is well under $2,000 if computers are already in-place.

We plan to adopt new technologies as they become available in modern engineering design. Our approach is based on both internal and external investment. The NAU College of Engineering and Technology has a very aggressive policy on computing equipment based on rotating replacement of PC workstations approximately every three years. In addition, we will seek corporate and government support to keep the laboratory and equipment current. One example is to evolve the course towards a hardware/software co-design model to take advantage of new processing architectures, such as those based on hybrid FPGA/programmable DSP models.

Course Evaluation

We employ an assessment program with one simple goal: to continuously improve the course in order to maximize improvement the student's learning experience. Specifically, we hope to help students advance to higher-order skills in the Bloom taxonomy while simultaneously mastering discipline-specific knowledge and tools.

Thus we augment traditional course/professor evaluations for two reasons: first, they are usually used for tenure/promotion and thus rely on averages of quantitative measures, and second, they are usually undertaken during class meeting hours and towards the end of the semester. Thus they are flawed—students usually make snap judgments because of lack of time and end-of-semester workloads. To improve this situation, we are employing *student awareness* and student *E-Teams*.

The first step is awareness on the part of the students. At the outset of the course we explain the extended Bloom taxonomy and its importance in assessing progress, to the class. Our hope is that the simple act of making students aware of the collegiate skills development process will engage them and accelerate that process. The second step is the organization of student "E-Teams" for evaluation and feedback to the instructor during the course. Each team is composed of students who volunteer as liaisons to the professor and provide a conduit for feedback from the students.

In previous use of E-Teams [16, 17], we found that the E-Team members provide a natural, trusted conduit for anonymous feedback from the rest of the students. The instructor meets informally with the E-Team members at mid-semester and at the semester's end. The E-Team members are also asked to prepare a short final report describing their findings.

References

[1] B. Bloom, editor. *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. David McKay Co., 1956.

[2] American Society of Engineering Education. The Green report: Engineering education for a changing world. In *Proceedings of the 1994 ASEE Annual Conference*, June 1994. (See also http://www.asee.org/pubs/html/green.htm).

[3] D. A. Kolb, editor. *Experiential Learning: Experience as the source of learning and development.* Prentice-Hall, 1984.

[4] Synthesis Engineering Education Coalition. The Synthesis strategic plan. http://best.me.berkeley.edu/ aagogino/synthesis/strategic.plan.html.

[5] C. C. Bonwell and J. A. Eison. *Active Learning: Creating excitement in the classroom (ASHE-ERIC Higher Education Report No. 1).* The George Washington Unviersity, School of Education and Human Development, 1991.

[6] D. L. Maskell and P. J. Grabau. Multi-disciplinary cooperative problem-based learning approach to embedded system design. *IEEE Transactions on Education*, 41(2):101–103, May 1998.

[7] J. Rhem. Problem-based learning: an introduction. http://www.ntlf.com/html/pi/9812/pbl_1.htm, December 1998. (National Teaching and Learning Forum On-Line Edition).

[8] B. Cheng, D. Rover, and M. Mutka. A multi-pronged approach to bringing embedded systems into undergraduate education. In *Proc. 1998 ASEE Annual Conference*, June 1998.

[9] F. Martin. The art of LEGO design. *The Robotics Practitioner: The Journal for Robot Builders*, 1(1), 1995. (also avialable at ftp://cherupakha.media.mit.edu/pub/people/fredm/artoflego.pdf).

[10] D. Baum. *Dave Baum's Definitive Guide to LEGO Mindstorms.* Apress, 2000.

[11] LEGO User's Group Network. www.lugnet.com.

[12] W. Wolf. *Computers as Components: Principles of Embedded Computing Sytem Design*. Morgan Kaufmann, 2001.

[13] E. Gat. On three-layer architectures. In D. Kortenkamp et al., editors, *AI and Mobile Robots*. 1998. (see also www-aig.jpl.nasa.gov/public/home/gat/tla.html).

[14] R. A. Brooks. A robust layered control system for a mobile robot. Technical Report 864, MIT AI Lab, 1985.

[15] A. Borenstein et al. Where am I? Sensors and methods for mobile robot positioning. ftp://ftp.eecs.umich.edu/people/johannb/pos96rep.pdf, 1996.

[16] P. G. Flikkema, L. P. Dunleavy, H. C. Gordon, R. E. Henning, and T. M. Weller. Wireless circuit and system design: A new undergraduate laboratory. In *Proc. 1997 Frontiers in Education Conference (FIE '97)*, 1997.

[17] T. M. Weller, P. G. Flikkema, L. P. Dunleavy, H. C. Gordon, and R. E. Henning. Educating tomorrow's RF/microwave engineer: A new undergraduate laboratory uniting circuit and system concepts. In *IEEE MTT-S Int'l Microwave Symposium Digest*, June 1998.

Paul G. Flikkema

Paul G. Flikkema is an Associate Professor of Electrical Engineering at Northern Arizona University. He has published papers in the areas of broadband wireless communication as well as in innovative engineering education techniques. He received the Ph.D. in Electrical Engineering from the University of Maryland, College Park, in 1992. From December 1993 to 1998, he was an Assistant Professor at University of South Florida, where he developed a new course in digital wireless communication and a new integrated lecture/laboratory course in signal and image processing. In addition, he served on a team that developed a new undergraduate laboratory in wireless and microwave communications technology, funded by the National Science Foundation and Hewlett-Packard. In the summer of 1998, he was a Japanese Society for the Promotion Science (JSPS) Research Scholar at Yokohama National University. In the summer of 2000 he was a Visiting Researcher at the Advanced Telecommunication Laboratory, SONY Computer Science Laboratories, Tokyo.