Designing an AI Elective to Encourage Undergraduate Research

Zachary Dodds

Harvey Mudd College Computer Science Department
301 Platt Boulevard
Claremont, CA 91711
dodds@cs.hmc.edu

Abstract

This paper describes the design and execution of a robotics-themed AI elective at a small liberal arts institution. An important goal of the course is to spark and nurture students' interest in pursuing related research while still undergraduates. To this end the curriculum turns the usual structure of a survey course on its head, pushing many fundamental topics later in the term so that students receive an early, thorough exposure to an important, recent algorithm known as Monte Carlo Localization (MCL). By starting with MCL, the de facto standard on cutting-edge mobile platforms, students have a touchstone from which to base novel projects and the time to do so. This work relates both the positive and negative experiences we have had with this approach.

Overview

AI-based survey courses are an important part of many undergraduate computer science departments' curricula. As our course's theme is robotics, it is the novelty of seeing one's data structures, algorithms, and software design physically embodied that attracts students to it. Yet this novelty also ensures that the students arrive entirely new to the material. In addition, as is typical of small, liberal arts colleges, this course is likely to be students' only experience with artificial intelligence and its many subfields. These factors have challenged us in our efforts to provide both encouragement and sufficient background for productive undergraduate research in the field.

Our approach exploits students' lack of experience to turn the syllabus "upside down," presenting some of the most recent and exciting advances in computational robotics at the start of the course, reinforced by lab assignments that ask groups of 2-3 students to implement these ideas. This schedule opens the latter half of the semester for student teams to pursue, if they wish, self-directed final projects. The most promising of these are then considered for summer support (as available), enabling the teams to refine their projects for demonstration and publication within the broader community.

Growing attention to the integration of undergraduate education and research has prompted both instructors and institutions to evaluate their curricula to this end (Gonzales 2001). Often the demands of graduating broadly knowledgeable majors conflicts with a desire to provide opportunities for narrower and deeper small-group investigations. Maxwell and Meeden (2000) describe their Swarthmore robotics course that succeeds in this balancing act: it has led to many entries to AAAI robotics competitions, including several winning ones. Having evolved from exactly that model, our course retains many of its features. It differs primarily in curricular structure, replacing its breadth-based approach with an initial *postholing* technique seen more often in history and the social sciences than in technical fields (Kornblith and Lasser 2001). By detailing these curricular choices and their implementation, as well as the lessons learned over the past five years, we hope to provide grist useful to others who might be designing AI-based electives.

Part 1: MCL as a Keystone Algorithm

Figure 1 presents an overview of the topics covered in the first half of our AI robotics elective. In this part of the term lectures and labs dovetail into a single keystone algorithm, Monte Carlo Localization (MCL) (Fox et al. 1999).

Briefly, MCL tracks a number of hypothesized locations for a mobile robot. As the robot moves, the inescapable inaccuracies of real-world motion increase the uncertainty of these hypotheses. When a robot senses its environment, either with a range sensor such as sonar or using a camera image, this invariably noisy sensor data culls poorly matching hypotheses. The robot uses its remaining, high-likelihood location estimates to reason about what action to take next.

Weeks 1-2: introduction to robotics and our hardware and software platforms; range sensing. **Lab:** open-loop motions and measuring the inevitable inaccuracies of actuation

Weeks 3-4: probabilistic models of uncertainty; basics of image processing. **Lab:** adding a sonar ranger to the robot, calibrating it, and obtaining simple probabilistic models of sensor error
Weeks 5-6: Bayesian reasoning and its role underlying MCL. **Lab:** basic image processing routines (color-based region identification and shape statistics) to extract hallway landmarks
Weeks 7-8: other probabilistic robotics algorithms: SLAM, evidence grids, and coastal navigation **Lab:** implementing MCL in a provided hallway map using their motion/sensing models
Week 9: Spring break

Figure 1 An overview of the initial topics covered in our *Robotics* elective. Details including lecture slides, written assignments, supporting software and lab projects are available online from `www.cs.hmc.edu/~dodds/courses`.

While simple, MCL motivates the Bayesian reasoning that plays so central a role within current work in artificial intelligence of all stripes. Though its basic tenets go back centuries, MCL per se is less than a decade old; as a result, it continues to motivate cutting-edge variations and applications (Elinas and Little 2005, Wolf et al. 2005).

The process of implementing MCL frames the labs and lectures of the first half of the semester. Building probabilistic error models of their robots' drive systems provides students a backdrop for learning how to use the hardware and software in the first two weeks. Creating complementary error models for their robots' sensors similarly motivates a short introduction to image processing and the construction of a sonar ranging unit. The integration of these models into the MCL algorithm reinforces these early labs and results in a coherent system that student teams can demonstrate by midterm.

Because MCL is a passive state-estimation algorithm, students can use simple wandering routines or even human-controlled motions to test their implementations. In contrast, many AI-based robotics courses present wandering or wall-following as early lab projects. Because these behaviors depend so intimately on the environment and sensor modalities being used, in our experience this approach focuses students' energies on writing code tailored tightly to our buildings' hallways. Although this provides realistic robot-programming experience and is justified by robotics's history, we found that it did not hint at the rapidly maturing and environment-independent basis for current state-of-the-art systems. Probabilistic robotics (Thrun 2002), on the other hand, puts such site-specific effort into a context that can be applied to any special-reasoning system based on noisy inputs.

Integrating Labs and Lectures

In order that students feel prepared for the lab projects, classroom lectures pace about two weeks ahead of these hands-on implementations. Small written assignments – both in-class and for homework – reinforce the material concurrent with its presentation. One advantage of this schedule is that students see more advanced mapping and navigation algorithms that build on MCL while they are implementing and testing that algorithm. Again following Maxwell and Meeden (2000), the lectures reinforce assigned papers on evidence-grid mapping (Martin and Moravec 1996), SLAM (simultaneous localization and mapping) (Montemerlo et al. 2002), and the entropy-based *coastal navigation* algorithm (Roy and Thrun 1999). These techniques further highlight the breadth and utility of probabilistic spatial reasoning. They also are intended to suggest directions for student-based final projects, which begin immediately after the break.

Part 2: Acknowledging Robotics's Breadth

Because students' lab projects diverge in the second half of the term, the lecture content less directly supports their out-of-class efforts. We try to exploit this disconnect in two ways. First, the curriculum returns to some fundamental robotics topics appearing earlier in more traditionally structured syllabi, e.g., configuration space and the many approaches to path planning within it. Figure 2 outlines these topics. Second, student teams punctuate lecture time with updates and demonstrations of their particular projects' progress. Written and programming assignments also become more important for reinforcing the lectures' topics, since lab work is less likely to do so. For instance, students build a vision program that plays the game of *Set* (`www.setgame.com`). Students' efforts culminate in public demonstrations of their projects on the final day of classes. The final online write-up is posted a week afterwards.

Weeks 10-11: more sophisticated computer vision algorithms, e.g., condensation, identical in spirit to MCL **Lab:** first deliverable in each student team's final project. For the default (fire-extinguishing) project, students build their Lego platform and write programs to test its motor and sensor subsystems.
Weeks 12-13: optical flow and its use in estimating time-to-collision. **Lab:** second final-project deliverable. For the default project, students write and test maze-wandering code.
Weeks 14-15: robots' configuration space and several path planning algorithms. **Lab:** final deliverable in each student team's final project, in which students are testing and tuning full, integrated systems.
Week 16: Dedicated to finishing the lab projects, including a term-end demonstration that can attract outsiders as well.

Figure 2 A synopsis of the second half of the elective.

Ultimately, the risks of overreaching have been the biggest drawback of student-selected capstone projects. We feel that the successful projects more than balance the need for additional instructor effort. Yet the projects that do not succeed leave students with a sour taste for the course, if not the whole field. Such disappointments have been most acute when caused, in part, by limitations of the hardware or software we have provided to the students. As a result, we have recently offered a “default” second-half project: building and programming a “fire-fighting” robot. This has been demonstrated with a wide variety of controllers including the Lego RCX, the Handy Board, and several custom-built platforms. Inspired by the Trinity College robot contest (Verner and Ahlgren 2002), this project has been calibrated to provide a moderate, attainable challenge.

Working Beyond the Semester

Despite the availability of this default project, we continue to seek out ways of encouraging students to explore outside the safety of well-worn paths. To this end we use external venues - and their deadlines - to motivate students beyond the necessarily short-range schedule of a four-month semester. We encourage student teams to point their work toward a particular poster, paper, or competition submission. The following experiences highlight the variety of venues that have provided concrete objectives toward which our students have aimed their efforts:

Educational Venues

In the spring semester of 2003, a pair of students wanted to investigate the then-new Evolution ER1 platform. Although difficulties with the software led to far less progress than they had hoped, the start they made and a continuation the following fall led to the development of a new graphical user interface and python-based support code for that robot. Although not robotics research per se, this effort was presented at the following spring’s education-themed AAAI Symposium. It also provided the following year’s students with a better starting point for their projects.

Beyond Botball

In the spring semester of 2004, a team of five students chose to build an entry to the Collegiate Botball competition (Miller and Winton 2004) as their final course project. Although we did not have summer support, three of the students could enter their platform (Figure 3) and attend AAAI, a collocated conference, because it was nearby. This competition runs each summer, now under the name *Beyond Botball*, under the aegis of the National Conference for Educational Robotics.

Poster Presentations

The experience at AAAI prompted one of the students to work during the fall of 2004 on an independent study to extend Monte Carlo Localization in order to take advantage of visually-mapped environments (Figure 3, top). The student submitted his work as a poster to both a regional site, SCCUR, and a national venue, SIGCSE, for undergraduate student research. In addition to benefiting the student presenter, this work was folded into the lectures of the 2005 offering of Robotics. Such feedback helped reinforce the idea that students’ course projects could be a jumping-off point, rather than a terminal experience in AI and, more generally, computer science research. Different events, such as CCSC’s and ACM-SE’s workshops, make such opportunities available throughout the country.

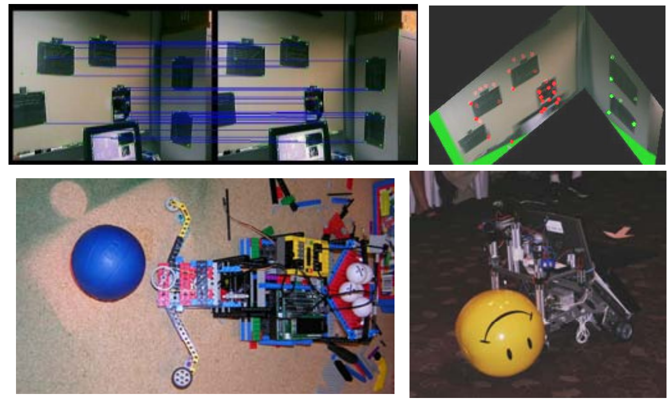

Figure 3 (**Top**) The feature tracking and resulting visual map anchoring the student work submitted to SIGCSE’s 2004 undergraduate student research competition (**Bottom Left**) A birds’-eye view of the students’ entry to the 2004 *Collegiate Botball* competition (**Bottom Right**) An Evolution ER1 platform retrieving the beach ball in the 2005 AAAI robot scavenger hunt.

AAAI Robot Competition

In 2005, one of the student teams worked toward entering the AAAI’s 14-year-old annual robot competition. These four students started with the course’s ordinary MCL labs through the first half of the course. In the second half, however, they integrated behaviors such as object identification and arrow-following (both extensions of the *Set*-playing assignment) into their map-based localization routines. In this case, summer support was available. Three of the students used May and June to add sensors and a second computer to their ER1 to create their competition entry, shown in Figure 3. In July they participated in the AAAI robot scavenger hunt and robot workshop, an experience all four consider a highlight of their undergraduate years.

Traditional Research Venues

The work presented at SIGCSE has led to the proposal and study of an algorithm coined *Monte Carlo Correction*, a vision-based MCL variant. Through the fall of 2005, a student has been investigating the effects of a pose-correction step in which MCL’s particles are updated based on the alignment of image features and their hypothesized world locations. The result has been prepared into a submission to a pattern-recognition research conference, with its status still pending. Similarly, the AAAI scavenger hunt work motivated those students to write up both the algorithmic (Davidson et al 2006) and engineering (Davidson et al. 2005) details of their platform.

Regardless of the outcome for a particular paper or robot submission, however, each of these efforts contributes to our overall goal: integrating students' coursework as much as possible with the ongoing flow of AI's educational and research communities.

Results and Lessons Learned

As the above descriptions show, the results from our offerings of *Robotics* vary based on many forces not completely within our control: students' interests, external venues for different kinds of projects, and the funding available for summer continuations of the semester's work.

As expected, the course's design has not produced only success stories. For each of the undergraduate research projects described in the last section, there have been more that have foundered, e.g., a group of seniors learning that FastSLAM was too much to completely implement as a final project. The flexibility offered in project hardware and scope carries substantial risks. The Evolution ER1's software stymied a dedicated pair of students in a full-semester effort to get it working in 2003; similar frustrations arose for students working with the Palm Pilot Robot Kit (Resko et al. 2002). In 2005 both the hardware and the software of the XPort Robot Controller, which uses the Nintendo GameBoy Advance as its computational engine, proved too poorly documented to support our final fire-extinguishing task.

More detail on the experiences and substantial efforts these student teams made toward their final projects' goals are documented in an archive of online write-ups available at `www.cs.hmc.edu/~dodds/projects`. In all of these cases, we feel the difficulties arise fundamentally from the ambition of the course design. After all, the student teams were investigating algorithms and hardware with which the instructor was not familiar. In addition, the variety of student-chosen projects undertaken in a class of 8-10 teams (20-24 students) yields limited instructor time available to master the details of each one.

Our philosophy has been to take a longer-term view of these efforts. Frustrations with the ER1 software led a group of students to rewrite its drivers and support code from the ground up (Dodds et al. 2004). That platform was used with success by *all* of the course participants in 2005. We are also looking to the next version of the XPort, known as the XBC (LeGrand et al 2005), as a very promising future platform, and we believe that with adequate support code provided, implementing FastSLAM will move from a final project to a required extension of MCL.

Indeed, students' reflections on the course suggest a realization that such challenges are an often unavoidable part of hands-on work with robots. Reported satisfaction levels have not varied in its five offerings. Suggestions for improvements consistently run the gamut from "if this is a computational course, why are we dealing with hardware at all" to "there should be more time on the hardware – its design, the sensors, etc."

Despite the drawbacks of our course design, we believe that five years of tinkering with its structure has helped to create an offering that can seed and nurture undergraduate research projects. Across that timespan, the following choices have the most valuable in our efforts to encourage undergraduate research through a survey course:

- **ensuring younger students are in the class** Sophomores and juniors are much more likely than seniors to seek summer research opportunities. The subsequent fall allows a natural buffer for writing and submitting results in time for graduate school application deadlines. Our early, all-senior offerings of the course left the next year with no experience on which to build.

- **starting with a keystone problem** As the previous sections describe, the careful treatment of a suitable problem, such as MCL, can help spark interest in a field's current work while motivating motivate fundamentals at the same time.

- **seeking out scalable tools** We are constantly considering how the software and hardware provided to students can best support *both* coursework and research. We have used the search for *pedagogically scalable* tools (Blank et al. 2004) both to improve students' lab resources and as a means for students to contribute to the larger AI and robotics communities.

In our search for hardware and software appropriate to our goals, we have gravitated toward the python-based Pyrobot software (Blank et al. 2005) and the laptop-carrying Evolution ER1. Both are low-cost and simple for beginners, and both scale easily to handle more sophisticated applications. Yet these resources are evolving: Pyrobot's API and accompanying resources continue to mature; the ER1 has sold out and Evolution Robotics does not plan another run. By encouraging students to consider educational-resource development as one way to extend their efforts beyond the course, we invite them to participate in the ongoing search for the "right" tools.

As we continue to mold this elective's design, we have not come to a final conclusion on how flexible to make students' laboratory projects. On one hand, projects like the Beyond Botball and AAAI contest entries show that flexibility can harness considerable student energy and creativity. Yet it also seems important, especially at the undergraduate level, to mitigate the risks inherent in such flexibility. To be sure, we will continue to tune this course's future offerings to better strike this balance between following a proven curriculum and encouraging experimentation beyond it.

References

Blank, D., Kumar, D., Meeden, L., and Yanco, H. 2005. Pyro: An Integrated Environment for Robotics Education. *Proccedings of the 22nd National Conference on Artificial Intelligence* (AAAI '05): pp. 1718-1719.

Blank, D.S., Yanco, H., Kumar, D., and Meeden L. 2004. Avoiding the Karel-the-Robot Paradox: A framework for making sophisticated robotics accessible. *Proceedings of the AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education* March 22-24, 2004. AAAI Press.

Davidson, A., Tribelhorn, B., Leung, T., and Dodds, Z. 2005. Low-cost range sensing for a laptop robot. *Proceedings of the 2005 Robotics and Automation conference* (RA 2005): pp. 329-333.

Davidson, A., Mason, M., Ricco, S., Tribelhorn, B., and Dodds, Z. 2006. Scavenging with a laptop robot. *Proceedings of the 2005 AAAI Robot Workshop* (to appear) AAAI Press.

Dodds, Z., Santana, S., Erickson, B., Wnuk, K., Fisher, J., and Livianu, M. 2004. Teaching Robot Localization with the Evolution ER1. *Proceedings of the AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education* March 22-24, 2004. AAAI Press.

Elinas, P. and Little, J. J. 2005. sMCL: Monte-Carlo Localization for Mobile Robots with Stereo Vision. Proceedings of Robotics: Science and Systems, June 8-10, 2005 Cambridge, MA.

Fox, D., Burgard, W., Dellaert, F., and Thrun, S. 1999. Monte carlo localization: efficient position estimation for mobile robots. *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (AAAI '99). July, 1999. AAAI Press.

Gonzalez, C. 2001. Undergraduate research, graduate mentoring, and the university's mission. *Science* 31; 293(5535): pp. 1624-6.

Kornblith, G. and Lasser, C. eds. 2001. Teaching the American History Survey at the Opening of the Twenty-First Century: A Round Table Discussion. *Journal of American History*, 87(4), pp. 1409-41.

LeGrand, R., Machulis, K., Miller, D. P., Sargent, R., and Wright, A. 2005. The XBC: a modern low-cost mobile robot controller. *Proceedings of the International Conference on Intelligent Robots and Systems* (IROS 2005). Piscataway, NJ: IEEE press.

Martin, M. C. and Moravec, H. 1996. Robot Evidence Grids. *Carnegie Mellon University Robotics Institute Technical Report #96-06*, 1996.

Maxwell, B. A. and Meeden, L. A. 2000. "Integrating Robotics Research with Undergraduate Education", *IEEE Intelligent Systems*, 15(6), pp. 22-27.

Miller, D. P. and Winton, C. 2004. The Botball kit for teaching engineering computing. *Proceedings of the ASEE National Conference.* Salt Lake City, UT., June 2004.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Proceedings of the AAAI National Conference on Artificial Intelligence*, AAAI Press, 2002.

Reshko, G. Mason, M. and Nourbakhsh, I. 2002. Rapid prototyping of small robots. *Carnegie Mellon University Robotics Institute Technical Report #02-11*, March 2002.

Roy, N. and Thrun, S. 1999. Coastal navigation with mobile robots. *Advances in Neural Processing Systems* 12(12): pp. 1043-9.

Thrun, S. 2002. Probabilistic Robotics. *Communications of the ACM*, 45(3): pp. 52-57.

Verner, I. and Ahlgren, D. 2002. Fire-fighting robot contest: interdisciplinary design curricula in college and high school, *ASEE Journal of Engineering Education*, 91(3): pp. 355-359.

Wolf, J., Burgard, W., and Burkhardt, H. 2005. Robust vision-based localization by combining an image retrieval system with monte carlo localization. *IEEE Transactions on Robotics* 21(2): pp. 208-216.