Undergraduate AI Research at a Liberal Arts College: A Case Study in Mobile Robotics

Kurt D. Krebsbach and Benjamin H. Willard
Department of Mathematics
Lawrence University
Appleton, WI 54912-0599
{kurt.krebsbach, benjamin.h.willard}@lawrence.edu

Abstract

Recent advances in inexpensive hardware and open-source software have transformed mobile robotics into a viable and exciting environment for undergraduate education and research in artificial intelligence. In this paper, we will describe one curricular model for successful independent student research with limited resources, and will demonstrate the results with a case study written by the undergraduate co-author himself.

Introduction

Educators in computer science have long understood the importance of engaging promising students in undergraduate-level research prior to graduate school. In artificial intelligence, mobile robotics provides a rich environment for such research, and has, in many cases, served as the necessary catalyst in a student's lifelong interest and development in the field. Until fairly recently, however, undergraduate departments have faced prohibitively high prices for these systems. Worse, even when such funds were available, the level of abstraction at which these robotic systems could be programmed was very low, forcing educators to shift their original pedagogical focus from the broader concepts of artificial intelligence to the engineering intricacies of joints, motors, and kinematics.

Recent advances in inexpensive, highly-functional hardware (e.g., the Sony AIBO) and long-term, open-source software projects (e.g., CMU's Tekkotsu project) have successfully bridged the abstraction gap and made mobile robotics feasible for undergraduate research in AI. In this paper we describe one curricular model that has been successful in our small liberal arts environment, and will argue that such institutions—while facing certain obstacles of size and scale—are uniquely positioned to exploit this trend toward low-cost, highly-functioning robotic systems due to strong faculty-student interaction and a commitment by both to individualized learning. Finally, we will illustrate our model through the eyes of the undergraduate currently engaged in such an independent research project.

Computer Science at Lawrence

Lawrence University is a highly-selective liberal arts institution enrolling approximately 1450 students and employing 120 full-time faculty. Due to its size, Lawrence does not have a dedicated computer science department, but does offer a full range of courses with a CMSC designation. The (approximately 2 FTE) faculty teaching computer science courses reside in the department of mathematics, which offers three majors: mathematics, mathematics-computer science, and mathematics-economics, the latter two of which are interdisciplinary. Across these majors, the mathematics department currently graduates about 30 students per year, with 9 specifically completing the math-computer science major on average.

Given these numbers, we currently teach a first course in artificial intelligence as an upper-level elective every other year (Russell & Norvig 1995). We also gain some advantage by teaching the programming languages course that precedes it in Scheme (Friedman, Wand, & Haynes 2001); thus freeing up more time to teach fundamental concepts and other programming paradigms (e.g., logic programming) in the AI course. While this may seem an impoverished environment for undergraduate research in AI, we will show how we have exploited other qualitative aspects of our situation to compensate for these constraints of scale.

A Mathematics-Rich Curriculum

One distinct advantage of being in a math department is the level of mathematical sophistication we can expect from our math-computer science upperclass majors. These students have already taken a minimum of 6 college-level mathematics courses by the end of their sophomore year, including a year of calculus, a course in combinatorics, a course in abstract algebra, and at least one upper-level elective chosen from graph theory, numerical analysis, and mathematical logic. This emphasis on abstract reasoning, rigor and precision in thought, logical argument, and methods of proof prepares students well for pursuing a variety of scientific research endeavors. Whereas some students might be discouraged by mathematical approaches to some area of robotics, these students are typically confident in

---

their practiced analytical skills and are not dissuaded from investigating such approaches, or those that cross disciplinary boundaries.

This idea, on a larger scale, is the fundamental idea behind a liberal arts education: that the best preparation for the future is an education that imparts and nurtures basic and transferable skills of inquiry, analysis, and communication. Education must be able to teach you how to jump—how to respond, adapt, change (Warch 1984). We advocate the notion that the best computer science education is not one that is narrowly confined to only computer science, and that the best researchers in artificial intelligence will come from backgrounds more broadly-based and connection-rich than can be acquired solely from traditional computer science curricula.

Building Computer Science Culture

An important part of an undergraduate experience in any discipline is a sense of community and common ownership of something larger. In the past, such a community of computer science undergraduates has traditionally been unusually strong due to the familiar constraint of sharing common computer labs late into the night in the pursuit of working programs. The importance of the hacker culture that emerged is hard to overestimate, and has been critical to the development of some of the best ideas in information technology, including the web, email, newsgroups, open-source software, and freeware.

While much of this automatic college interaction has dissipated with the arrival of laptops, wireless communications, and free downloadable course software, some has simply moved online. For the rest of the interaction, we have the Lawrence University Computer Science Club (LUCS). LUCS is an active student-run organization that has recently embarked on several volunteer software projects including community service projects such as voting software for student government and informational technical workshops for fellow students on topics such as virus protection, spyware, and alternative operating systems like Linux. As a result, LUCS was accorded formal student group funding status, and was allocated several rooms including converted robotics lab space now used for the various projects described here.

To take this idea one step further, several LUCS members organized an application to establish a Computer Science Theme House. These motivated student-citizens were awarded the largest of the campus-owned houses allocated for active student groups, and were the only such group awarded a theme house contract for the following year. The review committee was particularly impressed with the degree of volunteerism demonstrated by the house, and the degree to which the students actually used the house to achieve their curricular and extracurricular disciplinary goals; for example, setting up a wireless network, allocated space for robotics experiments, and holding practice programming contests for their members. Sharing residential space also encourages unplanned interactions among students (such as an evening discussion about genetic algorithms applied to slime volleyball) which, while informal, are nonetheless crucial to fostering active curiosity and experimentation in computer science. These students have shown remarkable leadership, organization, and passion for both curricular and extracurricular activities in their chosen discipline, and provide a model of what is possible with few resources and a surplus of motivation.

Opportunities for Individualized Work

With our low student-to-faculty ratio, Lawrence prides itself on providing one-on-one learning opportunities for its students. Many of these opportunities arise naturally due to small class sizes, close mentoring relationships, and the residential nature of education at Lawrence. Other such opportunities are intentionally built into the curriculum, and these form the basis of the educational model presented here.

Independent Study and Senior Seminar

A primary aim of the mathematics department is to transform our students from *other-directed* learners (e.g., in standard courses) to *self-directed* thinkers. For this reason, we require each of our majors to conduct an independent study course with the faculty member of their choosing sometime during their senior year.

We have also recently developed a written set of requirements and guidelines for preparing a formal proposal and insist that students submit their first draft at least a month before their intended term of one-on-one study. We then guide them through two or three iterations of revisions before the term begins to prevent needless foundering during the actual duration of their project. The importance of this iterative proposal preparation step cannot be overestimated: it provides a valuable lesson in real-world research practices, and makes a significant difference in the final result of their efforts.

Finally, we now require math-computer science majors to participate in a *Computer Science Senior Seminar* during the winter term of their senior year. The purpose of this course is twofold: to provide a more consistent and meaningful conclusion to seniors' required independent studies projects, and to further prepare seniors for their academic or career plans. Outside of class, the instructor will assist each student in preparing a formal presentation of their independent study project results which they will then present to their peers. Students doing their independent study course concurrently with the seminar will also have an early opportunity to do a short presentation of the proposal itself. We believe that this formal annual seminar will enhance an already research-rich environment within the mathematics department, and that such a course will appropriately address the university's current goal of providing each graduating student with a 'senior experience' in their chosen discipline.

Summer Research

Funding for the project presented here was provided in the form of a Lawrence grant from the Provost's *Enhancing Academic Distinctiveness Fund*. This new grant program (2005-06) is designed to support faculty work that contributes to Lawrences distinctiveness and is intended both to support faculty scholarship and to promote Lawrence as a leading institution of liberal learning. Especially emphasized are projects involving one-on-one work with students that either build on existing forms of faculty-student collaboration or explore new types of collaboration.

For the research supported here, the grant paid for one Sony AIBO robot, a laptop for programming and control, and a stipend to allow the student to work full-time on his project throughout the summer. The student's summer work allowed us to complete a significant portion of the experimental phase of the project before embarking on the curricular aspects such as the formal independent study, the senior seminar, and the Honors Project. Finally, we have taken steps to ensure that the knowledge gained and code base developed—along with the robot and laptop themselves—can be successfully transferred to interested students in future years with a minimum of effort.

Honors Projects

Honors Projects at Lawrence are coherent programs of independent work conducted by students on subjects or problems of more than ordinary difficulty and depth. Students in the sciences complete a fairly extensive written thesis under the supervision of a faculty member which is then read by an examining committee comprised of members from several departments including the student's own. The project culminates in an oral examination of the student's thesis by this committee, after which the committee submits its report and recommendation to the university's Committee on Honors, which reviews these materials and sends its recommendation to the faculty for a vote. A student may be awarded no honors, or honors at one of three levels. Such Honors Projects represent a vital step in the professional lives of our best students and potential future researchers.

We will now discuss a case study which we intend to submit for the Honors review process in May, 2006.[1]

Case Study: Map-Making with a Four-Legged Mobile Robot

Although map-making with an autonomous mobile robot is not a new idea, most current approaches rely on a set of common tools. Most recent research on map-making also assumes access to laser rangefinders, sonar arrays, precise shaft encoders, or GPS uplinks. These tools and sensors are regarded as standards for map-making, and it would be difficult to construct a map without any of them. Sonar and laser rangefinder sensors allow a robot to gauge distances to nearby objects. Similarly, wheels with precise shaft encoders can gather relatively accurate odometry information. Many map-making studies (Thrun, Burgard, & Fox 2000; Stewart *et al.* 2003), have employed laser rangefinders on a wheeled robot, but map-making with a robot lacking these standard features has received little attention. The Sony AIBO is one such modern robot that lacks standard map-making features. The AIBO is an excellent robot platform for undergraduate research because of its low cost and robust feature set. Still, the AIBO's sensors are less than ideal for the construction of *good* maps. A good map, in this case, is a map that mirrors the physical features of an environment to within an acceptable level of precision. A good map can also be used for navigating and localizing within an environment. This case study examines a student project that attempts to construct a good map of an indoor environment using only the standard features of a Sony ERS-7 AIBO.

The AIBO Platform for Mapping

Lawrence University currently owns one ERS-7 AIBO that has been affectionately named LARRY. The ERS-7 model was the last to be produced before Sony canceled the entire line. The ERS-7 has a long list of features, but the items most relevant to map-making are as follows:

- 4 legs with shaft encoders
- 3 neck joints (tilt, pan, nod)
- A head-mounted video camera
- 3 infrared distance sensors (short-range, long-range, and downward) located on the head
- Sensor updates every 32 milliseconds, with 4 samples per update.

Gathering useful odometry and range data from the ERS-7 sensors is a difficult and problematic task. Instead of wheels, the AIBO has four legs – legs that introduce a host of problems for determining odometry accurately. The AIBO's legs are quite prone to slippage, especially when rotating. The legs complicate the measurement of rotational distance when turning, and they make it difficult to maintain an accurate position estimate for the robot.

To gather range data, the AIBO must rely on two of its three infrared sensors. Each of the infrared sensors performs a specific task. The short-range sensor detects obstacles from 50mm to 500mm and the long-range sensor detects obstacles from 200mm to 1500mm. The frontal downward angle infrared sensor is used to avoid falling down stairs. The short- and long-range sensors are located on the AIBO's head, and they face

---

[1] Due to the somewhat different timing constraints of *undergraduate* research, the majority of the work—and consequently the main results—are not yet fully available in early October. We intend to have considerably more specific results for inclusion in the working notes in January, and even more to report at the workshop in March.

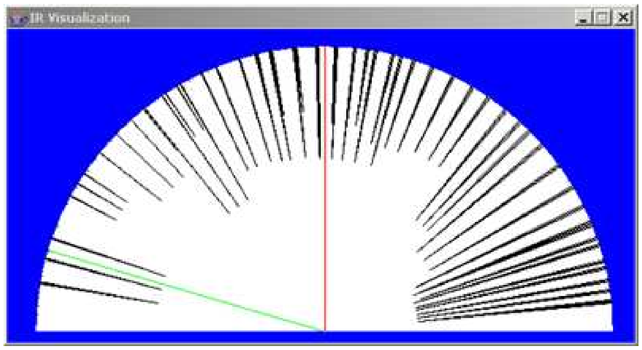

Figure 1: The infrared view module displays data collected from the most recent pan of the robot's head. Lines are drawn from the point an obstacle is detected to the bounds of the sensor range to better visualize the robot's perception of its environment.

directly forward. To acquire a 180-degree view of the environment in front of the robot, it is necessary to pan the head from side to side. This method of gathering range data is much different from the use of fixed sensors such as sonar, as it is impossible to acquire a broad view of the environment at one specific time. By panning the head, it is possible to create a snapshot view of the environment, represented by distances to obstacles, but each point is viewed at a slightly different time. Figure 1 shows a graphical representation of an infrared snapshot, as constructed by our program.

The AIBO platform at Lawrence

LARRY provides students interested in AI and robotics with a platform to build projects and conduct experiments. The AIBO was purchased with a grant to support several students interested in learning more about robotics and extending their knowledge from the *Artificial Intelligence* class at Lawrence. So far, LARRY has been at the center of two large student projects. The first project, entitled *Multi-Robot Navigation and Coordination* (Dan Casner and Ben Willard), made use of two AIBO robots – one student owned, one owned by the university – that attempted to rendezvous in a landmark-rich environment. The second project, *Map-Making with a Four-Legged Mobile Robot*, is the subject of this paper.

Mapping Framework

In this project, we are programming the AIBO to autonomously construct a map of an enclosed hallway area and later use that map to navigate to specific points in the hallway. At first, we will try to accomplish this task without giving the robot explicit knowledge of general hallway structure such as corridors or junctions.

The project is split into two phases. The first phase occurred as summer research, and the second phase takes the form of an Honors Project over the course of the 2006-2007 academic year. The summer was used for literature review, experimentation, and the construction of a development framework for mapping. The framework constructed during the summer will be extended for this Honors Project.

Half of the framework resides on the AIBO itself, and the other half operates on a more powerful laptop. The two halves of the system communicate via a wireless network link. The portion of the framework that runs on the AIBO is built in C++ with the help of the Tekkotsu development framework for AIBO robots developed at Carnegie Mellon University (Tira-Thompson 2004). Tekkotsu spares programmers from redundantly implementing low-level functionality to control robot movement and sensors.

The Tekkotsu-based code running on the AIBO is responsible for controlling the robots limbs, recording the speed of the robot, and collecting IR data by panning the head and reading from the IR sensors. The program running on the robot acts as a slave which essentially receives all higher level commands from the program on the laptop. For instance, the robot itself does not decide where to move. Instead, the robot sends sensor information to the laptop, and the laptop plans the actions the robot should take. Then, the laptop transmits instructions to the robot via the wireless network. The robot executes all instructions it received from the laptop, and the cycle continues.

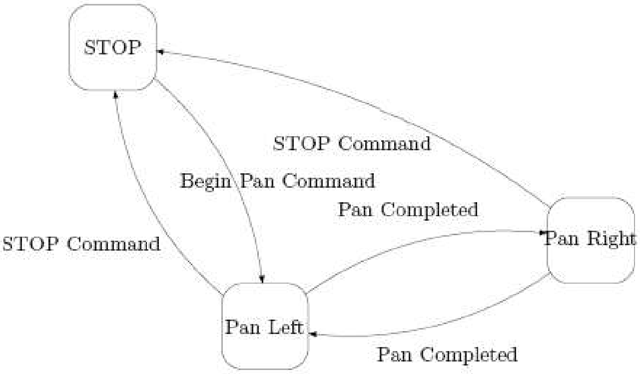

Figure 2: The Tekkotsu state machine that drives the robot's constant head panning is quite simple. A command from the laptop-based controller activates the panning which will continue until a STOP command is sent by the laptop.

The Tekkotsu program is based on two state machines. One state machine is responsible for controlling head movement and the other is responsible for controlling locomotion. Both state machines are embedded within a Tekkotsu *behavior* that receives commands from the network link and passes them on to the appropriate state machine. Boxes represent states, while arcs represent state transitions based on sensor readings or commands sent over the network. Figures 2 and 3 illustrate the two state machines. This setup makes the

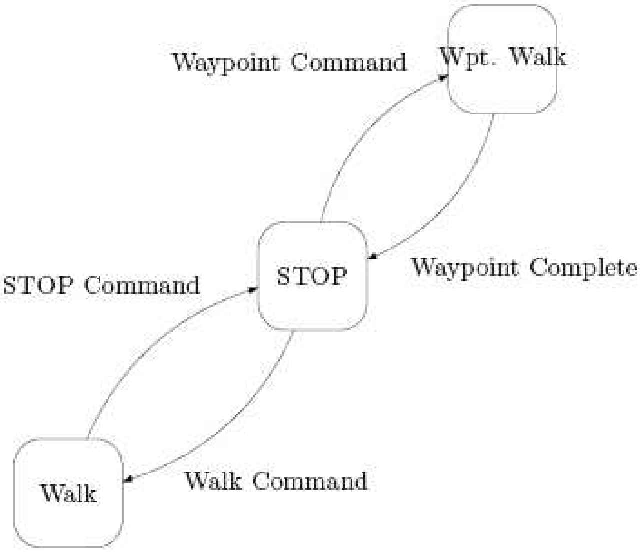

Figure 3: Locomotion is driven by this Tekkotsu state machine. A command from the laptop-based controller either sets the robot on a course specified by a Tekkotsu waypoint or instructs the robot to begin walking in a particular direction until a STOP command is sent.

Tekkotsu portion of the framework greatly expandable and fluid. It is a simple matter to add nodes to the state machines or even to add an entirely new state machine to a behavior, exemplifing how Tekkotsu assists in the construction of useful robot programs at a very high level of abstraction. Instead of worrying about how the legs are physically going to move, the focus is shifted to constructing higher level plans and programs.

We built the laptop-based controller portion of the framework in Java, and its main function is higher-level planning to determine the robot's actions. We designed the core of this program to accommodate the easy addition and removal of features. The unchanging role of the application is to establish a connection with the robot and to maintain sensor data that is gathered from the robot. From that point, it is possible to add modules that can access the sensor data from the robot and send commands back to the robot. Currently, the Java application has three distinct modules: a view of the IR sensor data from one sweep of the robot's head (Figure 1), an occupancy grid-based map (Figure 4), and a control module that uses information from the other two modules in order to steer the robot down the hallway. This framework provides a useful foundation for constructing a usable map in a hallway environment.

Project Status

The current map-making process uses two important types of information from the AIBO, namely the instantaneous velocity of the robot and the infrared sensor information. The robot's velocity is tracked by the Tekkotsu framework and is accessible in the Tekkotsu WorldState data structure. The infrared sensors are also easily accessible from Tekkotsu, but it is necessary to pan the robots head continually to get a full snapshot of the environment. As sensor data reaches the laptop, it is fed to the occupancy grid module. This occupancy grid is based on a model presented in Robin Murphy's text (Murphy 2000) but uses a modified sensor model for the IR sensor. Each grid unit has an associated value which indicates the likelihood of occupancy. For each sensor reading that is detected within a particular grid unit, the occupancy value is incremented. Likewise, all of the units that lie on a straight line between the robot and the detected obstacle are assumed to be empty, and occupancy values are adjusted accordingly. The result of this process is shown in Figure 4.

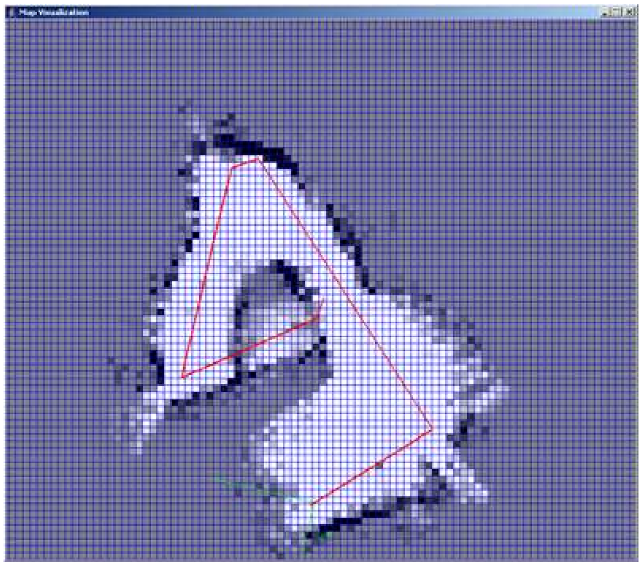

Figure 4: The occupancy grid module builds a map as the robot wanders through the environment. Darker squares indicate a greater likelihood that the area is occupied by an obstacle or a wall.

A notable failed experiment involved fusing data from the AIBO's video camera and infrared sensors in an effort to improve the quality of our occupancy grid. Inspired loosely by (Se, Lowe, & Little 2002), we attempted to use distinct landmarks to nail down particular locations in the environment. The robot was responsible for discovering the landmarks visually, with no initial landmark knowledge. Upon sighting a landmark, the robot would determine the location of the landmark using IR sensors and fix the location on the map. Then, if the robot ever happened to see that particular landmark again, it would compare the previously recorded location of the landmark with the currently observed location. A variety of different methods were tested to smooth out the error on the occupancy grid after the same landmark was observed in different locations, but no attempt proved satisfactory. In many cases, the map looked much worse than it would have without using the landmark data. The main problems with the approach were that the robot had difficulty distinguishing the unique landmarks, and no suitable method existed to fix the grid upon discovery of a discrepancy between old and new landmark positions. Although we do not intend to revisit this specific branch of the project, we may need to revisit the idea of fusing sensor data, particularly if the infrared sensors cannot provide enough data to accurately represent the environment.

Future Work

Between the extremely error prone odometry and the noisy, range-limited IR sensors, forming a good map-making strategy presents many challenges. Through the course of the 2006-2007 academic year, we hope to construct a high quality map-making system that will create reasonably good maps of indoor hallway environments. With a map-making framework in place, our short term goal is to minimize the odometry error produced as the robot walks. Turning introduces more error than walking in a straight line, so we will attempt to refine the way that odometry is computed while turning. Custom calibration of the Tekkotsu walk engine, fusion of data from the odometry and infrared sensors, and perhaps even modification of the leg movements involved in a turn are approaches we are considering for minimizing odometry error. Once the odometry error has been reduced below an acceptable threshold, we will move on to the next goal of refining the quality of the range data.

Conclusion

We have described a model for encouraging students to conduct independent research in artificial intelligence, and have demonstrated how available robotics hardware and software have facilitated this important undertaking. In particular, we have argued that while smaller computer science programs face certain quantitative barriers, other *qualitative factors*—such as a rigorous grounding in abstract reasoning, an enthusiastic and cooperative computer science student culture, and a faculty committed to individualized learning—can overcome these challenges. We invite others in similar circumstances to consider adopting or adapting some of the practices that we have found successful, and urge students and faculty at institutions like ours to support and enjoy faculty-student collaboration.

Acknowledgments

Many thanks to Lawrence Provost David Burrows for supporting this work through a Lawrence *Academic Distinctiveness* grant. Thanks also to Dan Casner for his work in developing the AIBO project at Lawrence, and to Paul Rybski for spending several days advising students in their AIBO-based endeavors.

References

Friedman, D. P.; Wand, M.; and Haynes, C. T. 2001. *Essentials of Programming Languages, second edition.* MIT Press.

Murphy, R. R. 2000. *Introduction to AI Robotics.* MIT Press.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Se, S.; Lowe, D.; and Little, J. 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research* 21(8):735–758.

Stewart, B.; Ko, J.; Fox, D.; and Konolige, K. 2003. The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In *UAI*, 551–558.

Thrun, S.; Burgard, W.; and Fox, D. 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics & Automation.*

Tira-Thompson, E. 2004. Tekkotsu: A rapid development framework for robotics. Master's thesis, Robotics Institute, Carnegie Mellon University.

Warch, R. 1984. *Liberal education, careerism, and the world of work.* Matriculation Convocation Address, Lawrence University Archives.