Development and Implementation of a MATLAB Simulation Project for a Multidisciplinary Graduate Course in Autonomous Robotics

SANJAY S. JOSHI

Department of Mechanical and Aeronautical Engineering, University of California, Davis, California 95616

Received 13 November 2003; accepted 15 December 2003

ABSTRACT: In this paper, we describe a MATLAB computer simulation course project implemented as part of an introductory graduate course in autonomous robotics at the University of California, Davis. The course consisted of students from Mechanical and Aeronautical Engineering, Psychology and Neuroscience, Electrical Engineering, and Computer Science. Creating a valuable class computer project experience for graduate students with greatly varying backgrounds is a challenging undertaking. Graduate students in different majors may have widely different educational backgrounds, field-specific language and culture, and computer background. It is a main result of this paper that a meaningful autonomous robotics computer project can be developed that allows students to be individually unique and creative, but at the same time allows each student to understand and share ideas, tools, and methods from all the other students. Simulation project conception, teamwork, implementation, results, and lessons learned are described. © 2004 Wiley Periodicals, Inc. Comput Appl Eng Educ 12: 54–64, 2004; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20001

Keywords: computer simulation; architecture; autonomous robotics; graduate course; multidisciplinary

Correspondence to S. S. Joshi (maejoshi@ucdavis.edu).
© 2004 Wiley Periodicals Inc.

INTRODUCTION

In Spring 2003, a new interdisciplinary graduate course in autonomous robotics theory was introduced at the University of California, Davis. Autonomous robotics by its very nature is multidisciplinary, and graduate research in autonomous robotics is very often interdisciplinary. New autonomous robotics techniques and applications are quite often the result of collaborations between scientists in widely separated fields (e.g., Ref. 1). Therefore, a goal of the course was to attract an academically diverse student body. The first offering of the course consisted of 12 students from Mechanical and Aeronautical Engineering, Psychology and Neuroscience, Electrical Engineering, and Computer Science.

The cornerstone of the course was a 10-week autonomous robotics computer simulation design project. This project was meant to give students individual experience in the autonomous robotics methods discussed in the classroom, and to serve as a platform to gain skills in teamwork, team writing, and team presentation. As most robotics educators would agree, the best way to learn and appreciate robotics is to create robotic systems. Ideally, students would work with both hardware and computer simulation. Several examples of autonomous robotics hardware projects have been reported in the literature (e.g., Refs. 2–5). These are mostly at the undergraduate level. Since we were constrained by both time and resources, we instituted a robot computer simulation class project in MATLAB. Autonomous mobile robotics simulation course projects have not been widely reported in the literature to date. Ning et al. [6] describe an educational simulation platform based on the Robocup competition. Simulation projects using computers for traditional robotics design tasks such as arm kinematics and low-level control have been well reported (e.g., Refs. 7–9). However, these projects do not include the study of autonomy.

Creating a valuable class project experience for graduate students with greatly varying backgrounds is a challenging undertaking. However, it is precisely these challenges that make an interdisciplinary project unique and valuable. First, students have widely different educational backgrounds and field-specific language and culture. Ideally, the computer project should be accommodating to various field-specific ideas. Part of the goal of the course was to develop common language to collaborate and explain ideas to others. Second, students have different levels of proficiency in computer programming. It would be expected that graduate students outside engineering would have less background in formal computer programming. However, a trend in all scientific disciplines is the greater and greater use of computer tools, so this difference may not be as large as in the past. Third, students must be able to complete the course project in a 10-week quarter system. Finally, as this is a graduate course, the course project must contribute to the overarching goal of better-preparing the students to contribute to the research field of autonomous robotics. It is a main result of this paper that a meaningful 10-week computer-based autonomous robotics course project can be developed that allows students to be individually unique and creative, but at the same time allows each student to understand and share ideas, tools, and methods from all the other students.

This paper is presented as follows. In "Goals of Computer Simulation Project," specific goals of the computer simulation project are discussed. In "Student Background," the student participants and their backgrounds as related to the computer simulation project are described. In "Team Organization," team organization and related issues are discussed. In "Project Development," development of the computer simulation project is discussed. In "Project Realizations," results from the computer simulation project are presented. In "Recommendations," lessons learned and recommendations for similar courses are described. Finally, in "Conclusions," final remarks are given.

GOALS OF COMPUTER SIMULATION PROJECT

The goals for the course project spanned three categories. First, the project was to serve as a vehicle to implement general autonomous robotics theory and techniques discussed in the classroom. The class syllabus included autonomous robot control paradigms (e.g., reactive, deliberative), biological and psychological bases for robotic paradigms, specific autonomous robot control architectures (e.g., subsumption, potential-field based), adaptive robotics including the use of genetic algorithms and neural networks, robot sensors, biological and robotic foraging theory, history of robotics, robotics simulation, and applications of autonomous robots. Second, the project was meant to give students experience in robotics-issues related specifically to simulation including understanding internal versus external representations of environments, understanding uncertainty in sensor models, understanding valid uses for robotic simulation, and understanding how to interpret simulation data in the context of informing hardware design. All these issues were also discussed in the classroom. Finally, the simulation project was meant to give students experience in teamwork skills across disciplines. This included facilitating day-to-day research progress, explaining research progress to others via interim and final presentations, and midterm and final written reports.

STUDENT BACKGROUND

It was important to understand the specific computer background of the students, especially because of the varying educational disciplines of the students. The simulation project was to be completed in 10 weeks starting from no code. Knowing the computer background of the students allowed us to realistically form expectations for the final projects and helped in assigning teams. Information was gathered using a survey in which the students rated their abilities and gave written comments [10]. Figure 1 shows the results when the students were asked to rate their familiarity with computer simulation in general. Only a few students had ever created a computer simulation. However, another part of the survey indicated that every member of the class was extremely familiar with computer programming. This included even the non-engineering majors. For example, the neuroscience/psychology students reported using MATLAB for their own research. Indeed, every student knew at least two computer languages and many students knew five or six (including C, C++, Fortran, Basic, Java, Pascal, Prolog, Visual Basic, Perl, and Assembly).

MATLAB was chosen by the instructor as the simulation environment for the project. Figure 2 shows the results when the students were asked to rate

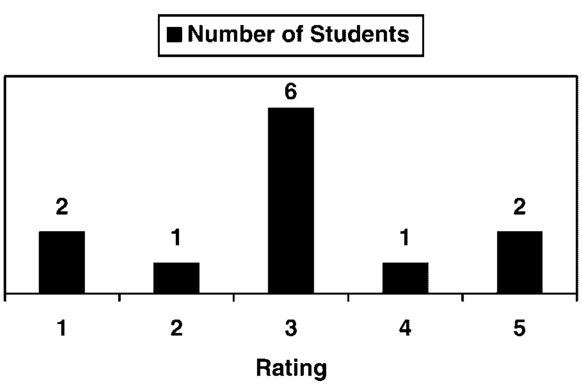


Figure 1 On a scale of [1−5], how familiar are you with creating computer simulations (1 = no familiarity, 5 = high familiarity)?

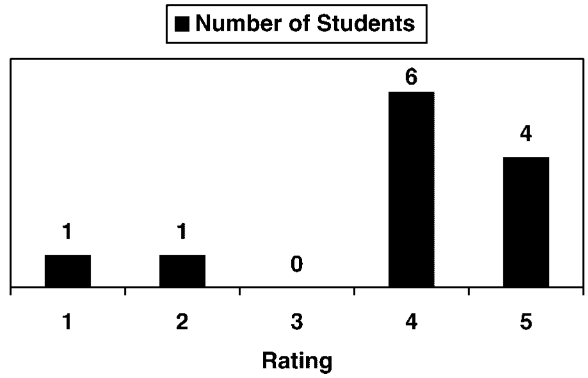


Figure 2 On a scale of [1−5], how familiar are you with the MATLAB programming environment (1 = no familiarity, 5 = high familiarity)?

their familiarity with MATLAB. Ten out of 12 students were quite familiar with MATLAB. This was expected as MATLAB is used in many different disciplines, is well supported at UC Davis, and is incorporated into many engineering classes at UC Davis. The two students who rated themselves inexperienced with MATLAB were both Computer Science students. However, these students also had the most experience in general computer programming. Based on these results, we felt comfortable that the students could achieve a valid educational experience in robotics without being overwhelmed by computer programming issues.

TEAM ORGANIZATION

The inter-departmental composition of each of the project teams was an important decision. Students worked in pairs. There were two choices in forming teams. The teams could be forced to consist of different majors, or the teams could be allowed to include two people with the same major. Clearly, cross-disciplinary exchange and communication were central to the course. The group-makeup decision affects the setting in which cross-disciplinary exchange and communication occurs. Forcing teams to be multidepartmental allows for one-on-one cross-disciplinary exchange outside of class. In addition, methods on overcoming challenges in multidisciplinary research could potentially be practiced at a one-on-one level outside of class.

On the other hand, allowing teams within a single major does not necessarily prevent cross-disciplinary exchange and communication—with proper inclusion of classroom exercises. However, these exchanges would primarily occur in the classroom. In fact, the cross-disciplinary exchanges could be potentially more fruitful since the discussion could be structured and facilitated. Also, teams could be chosen on more practical bases like common free time to have team meetings, and familiarity with the same software tools so that effort could be more easily divided.

In the first offering of the course, the students were allowed to pick their own groups—knowing that most would pair up with their own major (due to familiarity of potential partners). Indeed, four out of six groups paired within their major (three Mechanical Engineering Teams and one Computer Science Team). However, two out of six groups crossed majors (Psychology/Mechanical Engineering and Psychology/Electrical Engineering).

Given that many of the groups did not cross department boundaries, a structured forum for discussion of class projects on an ongoing basis was required. In order to address this need, a very structured in-class discussion was required in the form of Simulation Update Classes. Three 2-h sessions were held during the quarter. Each team was required to update the class on their computer simulation activities and gain feedback from the class. Each team's discussion was required to include three components: (1) "How can we use our recent robotics computer simulation experience to help the other teams?" (2) "What recent computer simulation problems/issues came up that we would like the rest of the class to help us with?," and (3) a computer simulation demonstration. In addition, each team was required to absorb the discussion, and produce a one-page memo that was submitted in writing by the next class. The students later reported that this was one of the most helpful components of the class.

PROJECT DEVELOPMENT

It was important that the simulation task be common enough to all teams so that teams could share valuable experiences with each other, but flexible enough to allow maximum creativity. Before the course began, a foraging task was chosen that involved autonomously picking up "good" objects and bringing them to a goal, while disregarding "bad" objects. Foraging is a concept that spans biology, psychology, and robotics (e.g., Refs. 11–13).

Although the general objective of the class project was chosen a priori, the specifics of the task were discussed and agreed upon through class participation. Topics of discussion included:

- How many robots are allowed?
- How many sensors are allowed?
- How many obstacles are in the environment?
- Where is the goal?
- How large is the goal?
- How many objects are in the environment?

After consensus, the final challenge for the simulation was a robot that autonomously searched and retrieved 20 rewards scattered in a given environment using an unlimited number of sensors. The main objective of collecting rewards was further complicated by the presence of both obstacles and undesirable targets that were randomly scattered across the entire field. Furthermore, the robot had no initial knowledge regarding the environment (e.g., goal location, number of objects, etc.).

Several assumptions and rules were agreed upon for the external world. The environment was a 50×50 grid-based field with one 10×1 goal located along the center of one wall. In addition to this goal region, there were 20 reward objects (1×1 each), ten hazard objects (1×1 each), and two impassable, infinitely tall obstacles (4×4 each), all of which were randomly set in the environment with no overlap. Both reward and hazard targets were considered flat—thus they did not hinder or block the actual motion of the robot. This also applied to the goal region as the robot was free to move in, out, and through the 10×1 region. The goal could hold any number of objects. Finally, all effects of dynamics (e.g., friction, momentum) were neglected. The simulation focused on the autonomy control architecture and the modeling of sensors, rather than engineering dynamics.

Each team programmed the same task, but the specific characteristics of the good and bad objects, the number and types of sensors, the computational architecture, and the foraging algorithms were left for each individual team to decide. The challenge immediately suggested certain core features each simulation would need including a control architecture, obstacle avoidance capability, object and goal search capability (intelligent or non-intelligent), and object discrimination capability. However, each team was to tailor the characteristics and methods to their interests. Furthermore, each team was encouraged to motivate their choice of sensors, assumptions, object characteristics, etc. through the use of a background "story" for their project. The teams created a variety of scenarios. One team modeled the task after a factory, in which the robots used bar codes to load some boxes into a truck, but left others behind. Another team modeled the task after an inverse garbage collector, who collected pleasant smelling lemons, but left behind foul-smelling eggs. Yet another team modeled their task after a snake looking for food using thermal cues. In the next section, we review each team project realization in order to show how this generic task led to unique learning experiences for each team.

PROJECT REALIZATIONS

Group 1: Mechanical Engineering–Mechanical Engineering Team

Group 1 created a robot motivated by a snake stalking out food during the night [14]. As such, their robot lived in a thermal world with no visible light. The robot searched a simulated world consisting of 20 hot objects and ten cold objects. The thermal fields from the hot and cold objects interacted to create varying temperature regions in the environment (Fig. 3). Their simulated robot included both a directional infrared camera and omnidirectional ultrasonic rings for sensors. The IR camera was used to sense thermal fields at a distance in order to roughly navigate the robot into an area of "good" objects. The ultrasonic sensors were used to hone in and guide the robot for object collection and retrieval. They created a sophisticated hybrid control architecture in which reactive modules were used for wall-following, obstacle avoidance, and wandering in the absence of map information, and deliberative modules were used for map building and exploration planning. An overview of their architecture is shown in Figure 4.

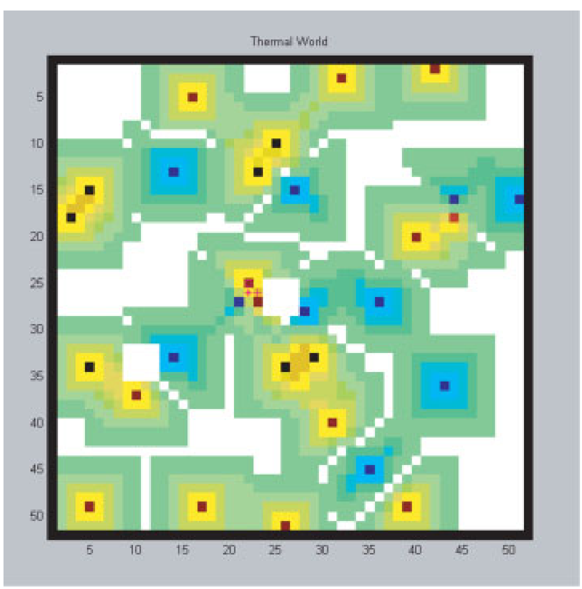


Figure 3 Thermal fields around hot and cold objects in Group 1's simulation [14]. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

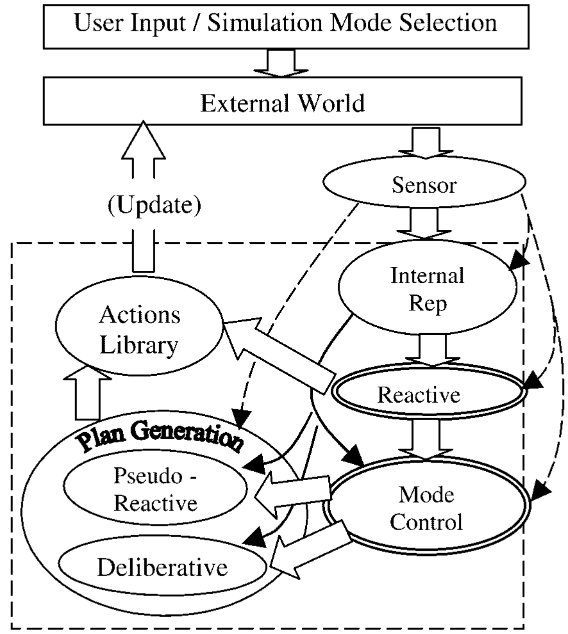


Figure 4 Hybrid control architecture for Group 1 [14].

Group 1's foraging framework and robot allowed them to explore many unique challenges specific to their project. In addition to walls and obstacles, their robot had to deal with overlapping thermal signatures. For example, they found that a hot object with a minimum of three cold objects in the square next to it could be perceived as a cold object since the surrounding cold fields canceled its heat signature. In their simulation, this initially resulted in the robot's inability to find that object. Similarly, a cold object surrounded by hot ones was perceived as a hot object, and sometimes resulted in the robot picking up a cold object and dropping it off at the goal.

Group 2: Mechanical Engineering–Mechanical Engineering Team

Group 2 created a simulated robot that picked up objects that were one color and rejected objects that were another color [15]. In order to accomplish the task, they built a robot that was equipped with a color camera that scanned three cells in front of the robot, magnetic rotary encoders on the wheels for dead-reckoning, digital bump sensors, and RGB electronic color sensors for detecting colors underneath the robot. Unlike other groups, they also assumed that the robot had an extended gripper at the front (similar to the Khepera robot shown in Fig. 5). As such, their

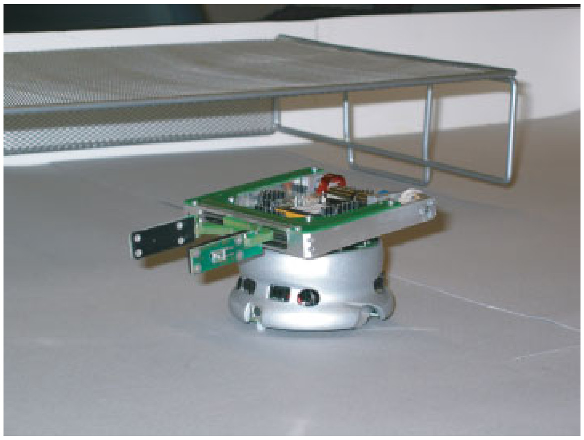


Figure 5 Group 2 modeled their robot with a front gripper, like the one shown on the Khepera robot [16]. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

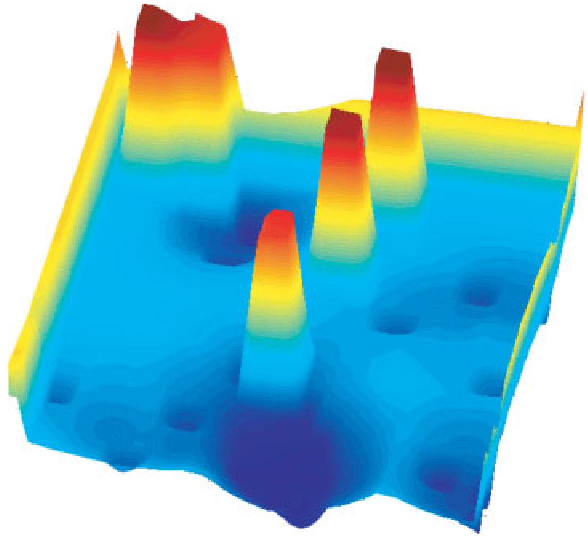


Figure 6 Composite potential field for Group 3 [17]. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

robot occupied two grid spaces in the simulated environment (as opposed to other groups' one grid space robots). This assumption forced them to deal with added navigation issues in guiding the robot and extended gripper around the environment.

Group 2 also used a hybrid control architecture. The robot sensed the environment using its bump sensors, encoders, digital camera, and color sensors. The robot first located the goal and created a home reference frame for its internal representation of the environment. A sweeping pattern enabled the robot to scan the field until the mission was completed.

Group 3: Psychology– Mechanical Engineering Team

Group 3 studied potential field methods in robotics and implemented their algorithms using a purely field-based approach [17]. Their robot created internal potential fields inside its memory based on sensor inputs from obstacles, good and bad objects, and the goal. Sensors used to gather information included a linear camera and a laser range finder. They created a series of discrete modes that the robot employed to finish its task. For example, modes included ''Gather-Resource,'' ''Avoid-Obstacle,'' and ''Goto-Goal.'' For each mode, a unique gradient-field was created that was either repulsive or attractive. Modes were executed in parallel, where the fields were summed to create composite fields. An example of a composite field for ''Gather-Resource'' and ''Avoid-Obstacles'' is shown in Figure 6.

Group 3 employed a Central Executive that used a deliberative process to decide which mode the robot should execute at any given stage of the task. In their study, local minima were a challenge. They explored ways to avoid local minima using averaging techniques. In some cases, they found that eliminating one minima created another. Their user interface is shown in Figure 7.

Group 4: Psychology– Electrical Engineering Team

Group 4 built an inverse garage collector robot that collected sweet smelling objects and ignored foul smelling objects [18]. As such, they embodied their robot with a biologically based smell sensor to distinguish between sweet and foul smelling objects. As these ''Sweet'' and ''Foul'' descriptions were not precise, they were able to study and experiment with fuzzy-logic type techniques. Since the smell sensor could only be used at very short range, their robot also used a laser range finder and sonar sensors. Unlike the other groups, they specifically designed their robot control architecture to be purely deliberative. At each step, the robot collected new information about the environment, updated a world model, and formed a new plan of action.

The group also tested four competing search strategies to compare results [18]. These included: (1) sweep and sense: the robot swept across space,

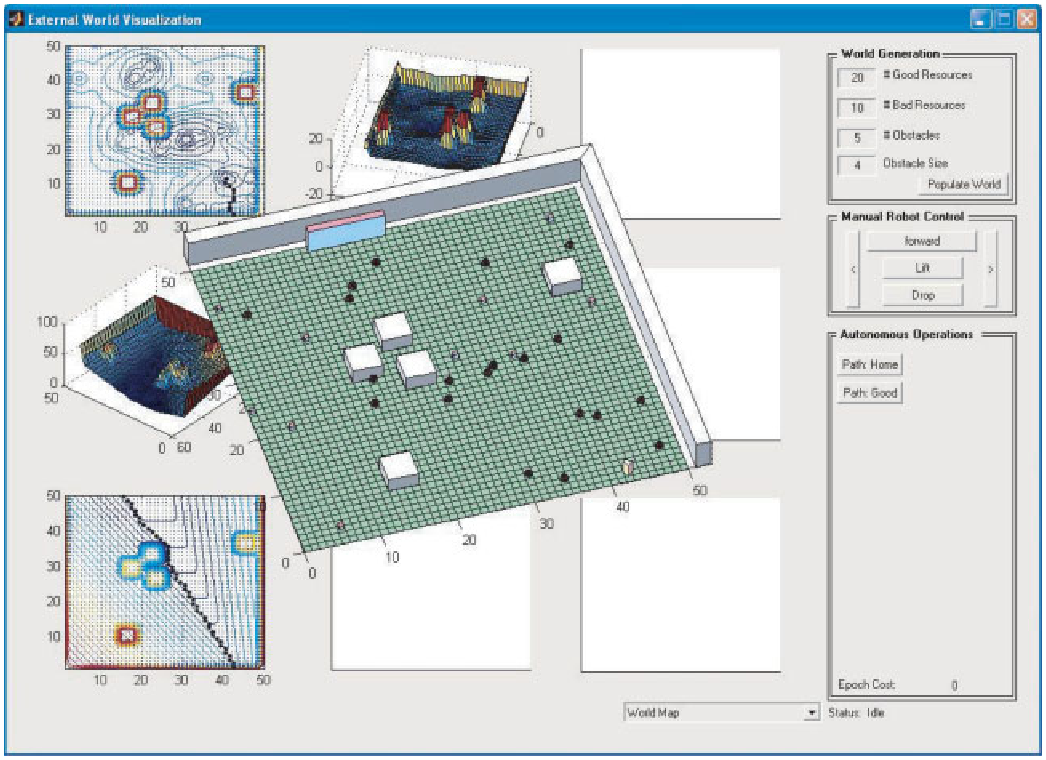


Figure 7 User interface for Group 3's potential field control architecture [17]. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

skipping every other row to maximize its sensing area, (2) exhaustive sweep: similar to strategy (1), except that the robot did not skip rows and searched every row, only utilizing its smell sensor when an object was directly in front of it, (3) random without memory: each step the robot took, it chose a random direction in which to explore, sensed that area to determine whether it could move to that grid space (i.e., whether that grid space was free of obstacles and walls), and moved in that direction if possible, and (4) random with memory: similar to strategy (3), except the robot stored locations of all grid spaces it visited.

Group 5: Mechanical Engineering–Mechanical Engineering Team

Group 5 explored an almost purely reactive approach to the problem [19]. They outfitted their robot with a light intensity sensor and a series of bump sensors. Their robot wandered around in random directions until it found a good (well-lighted) object. It then returned it to the goal using dead-reckoning. If their robot encountered an obstacle or wall, it moved randomly until its bump sensor was disengaged. They created a series of simulation tests using a random placement of objects. As shown in Figure 8, time-to-complete-task and steps-to-complete-task varied greatly between runs.

Total Time-steps	Moves	Turns	Time Elapsed (s)
56996	45479	11517	9220
9631	7702	1929	1791
37184	29521	7663	6880
15286	12089	3197	2490
31300	24951	6349	5810
18210	14547	3663	3399

Figure 8 Results from simulation experiments employing Group 5's reactive architecture [19].

Group 6: Computer Science– Computer Science Team

Group 6 also used an almost purely reactive approach to the problem, but in their case, they searched for heated objects of a certain color [20]. They created a subsumption-type architecture where several behaviors ran in parallel. Obstacle avoidance was done with the aid of a laser range finder that allowed the robot to detect walls and obstacles at a greater distance than Group 5's bump sensors (of course at greater cost). Interesting, they found that using the long range of the laser caused problems in the ''Return-to-Goal'' behavior as walls behind the goal were sensed and avoided. Thereby, the ''Return-to-Goal'' behavior conflicted with the ''Avoid-Obstacle'' behavior. They avoided this conflict by reducing the range of their laser sensor. Object-gathering was done by using a combination of reactive heat-seeking and color-matching. ''Return-to-Goal'' was done using dead reckoning. A view of the user interface they created in MATLAB is shown in Figure 9.

A summary of the main characteristics for each project is shown in Figure 10.

RECOMMENDATIONS

In order to assess the success of the student project experience and to receive student feedback, we obtained quantitative information and we held one-on-one meetings with each student. There are several conclusions and recommendations that resulted.

Feasibility and Time Commitment

One quarter (or one semester) is not too little time to complete a meaningful autonomous robotics simulation project. The scope of the project depends on the computer background of the students and the

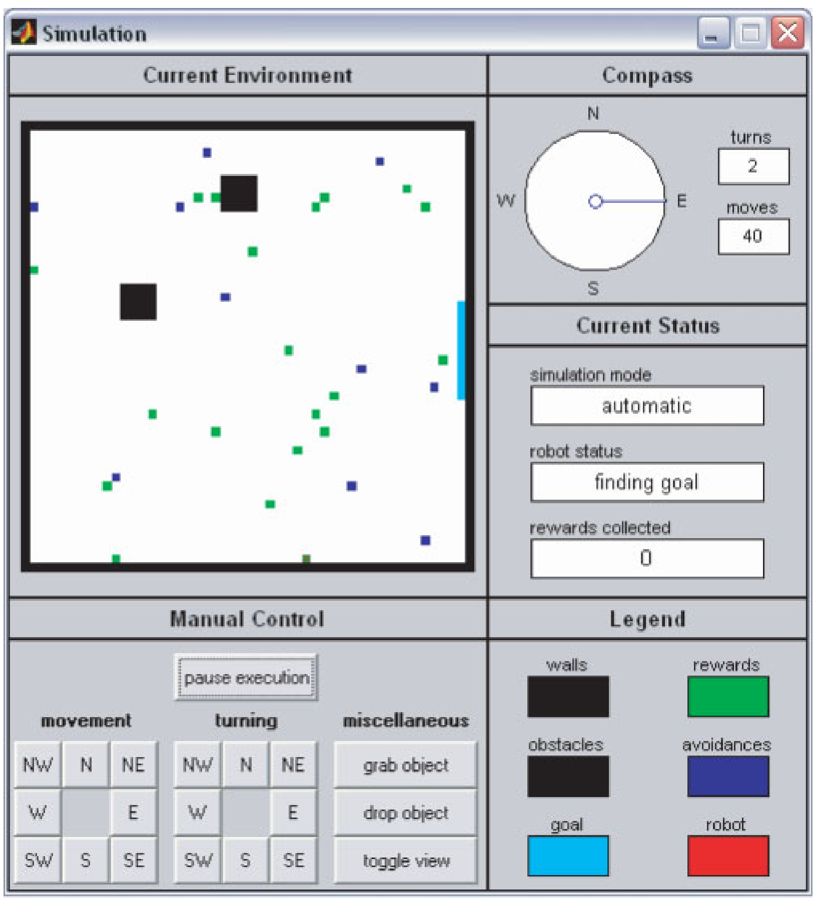


Figure 9 User interface for Group 6's robot [20]. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Team Majors	Mechanical Engineering-Mechanical Engineering	Mechanical Engineering-Mechanical Engineering	Psychology/Neuro-Science-Mechanical Engineering	Psychology/Neuro-Science-Electrical Engineering	Mechanical Engineering-Mechanical Engineering	Computer Science-Computer Science
Sensors Used	Infrared Sensor, Ultrasonic Sensors	Color Camera, Wheel Encoders, Bump Contact Sensors	Linear Camera, Laser Range Finder	Laser Range Finder, Sonar Sensors, Chemical Sensors	Photodiode Light Intensity Sensor, Wheel Encoders, Bump Contact Sensors	Laser Range Finder, Infrared Sensor, Reflective Color Sensor
Control Architecture	Hybrid	Hybrid	Hybrid	Deliberative	Reactive	Reactive
Unique Robotics Techniques	Map Making	Map Making	Potential Fields	Fuzzy Logic Ideas	Dead Reckoning	Subsumption Ideas

Figure 10 Summary of projects.

availability of a common software environment. Since students come from many different academic departments, it is not easy to predict their computer background and first-week polling is helpful. Since MATLAB is a tool used in several disciplines, it is one natural choice for a simulation environment.

When asked to rate the course time commitment on a scale from 1 to 5 (1 = time required much less than other graduate courses, 2 = time required somewhat less than other graduate courses, 3 = time required about same as other graduate courses, 4 = time required somewhat more than other graduate courses, 5 = time required much more than other graduate courses), almost all students rated in the (3–5) range. If course time commitment becomes an issue in the future, there are some ways that the course time commitment may be reduced. For example, all aspects of the course project could be defined a priori to give students more time (discussed more below), students could be given starter code that could be modified instead of starting with nothing, or the instructor could define success as working out only part of the overall task. These issues will be the subject of future study in subsequent offerings of the course. Even with the above average time-commitment, students were very positive about the course on the year-end anonymous class evaluation.

Problem Definition

Allowing students to decide the specifics of the project definition helps promote ownership of the project and commitment to complete the goals of the class project. However, this implies that the project definition is not complete from the first day of class. As a result, some time is lost for actual project work. Some students provided feedback that in such a short course, more time for work would have helped. This could have been accomplished with a completely defined project at the beginning.

Team Size

In this course, students worked in pairs. Students reported that this worked well, but splitting up work was not always easy. As a result, much of the work was done in the same room at the same time. The project would probably not have been possible for most students working alone. This would also defeat the teamwork goals of the project. However, the students reported that having three people on a team would not be feasible and could even harm progress.

Team Composition

Working in teams composed of different majors was very rewarding for those teams that did it. Students reported that they exchanged very different ideas that were helpful to their project. However, in an academic environment, this also caused some difficulties with mundane but important issues such as scheduling common time to work on the project. Students within the same major had less difficulty since their schedules were similar. Students also reported that the Simulation Update Classes were very helpful to gain new ideas and exchange cross-departmental ideas.

These sessions were especially important to teams that did not cross disciplinary boundaries.

Project Specifics

For our particular simulation project, some students reported it would have been helpful to constrain the number of sensors. From an educational point of view, we decided early not to create a competition. Students would be allowed to use as many sensors as they wished to research. However, a valid concern was the tradeoff in sensor accuracy versus sensor cost. Some students expressed that giving each team a fictitious budget would help with deciding which sensors to use, and how to use sensors efficiently.

CONCLUSIONS

In this paper, we described a quarter-long (10-week) computer simulation project completed in an introductory graduate autonomous robotics course at UC Davis. All the teams started with no code and were able to program robot visualization, an external environment, simulated sensors, and a robotic architecture within 10 weeks. Each team approached the problem in an individually unique way that allowed them to explore individual sensing and reasoning methods. It was a main result of this paper that a meaningful autonomous robotics computer project can be developed that allows students to be individually creative, but at the same time allows each student to understand and share ideas, tools, and methods from all the other students.

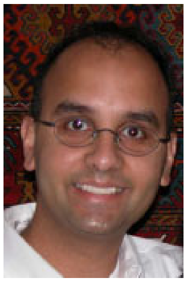
ACKNOWLEDGMENTS

The author thanks the students of Mechanical and Aeronautical Engineering 298-2, Spring 2003 at the University of California, Davis. The author also thanks Associate Editor Dr. Rodney Cole for his valuable comments and discussion.

REFERENCES

[1] B. Webb and T. R. Consi, editors, Biorobotics: Methods and applications, AAAI Press/MIT Press, 2001.
[2] P. Gregson and T. Little, Using contests to teach EE juniors, IEEE Trans Educ 42(3) (1999), 229−232.
[3] K. Newman, J. Hamblen, and T. Hall, An introductory digital design course using a low-cost autonomous robot, IEEE Trans Educ 45(3) (2002), 289−296.
[4] C. Pomalaza-Raez and B. Groff, Retention 101: Where robots go, students follow, J Eng Educ 92 (2003), 85−90.
[5] S. Shooter and M. McNeill, Interdisciplinary collaborative learning in mechatronics at Bucknell university, J Eng Educ 91 (2002), 339−344.
[6] W. Ning, Y. Zhao, Z. Yang, Y. Cai, R. Ma, and J. Wang, Design and implementation of educational platform in Robocup simulation games, Proceedings of the first IEEE international conference on machine learning and cybernetics, Beijing, Vol. 2, 2002, pp 578−581.
[7] K. Sridharan, Teaching computer graphics and robotics using symbolic computation software, Comput Appl Eng Educ (2000), 18−30.
[8] B. White, R. Read, M. Koch, and R. Schilling, A graphics simulator for a robotic arm, IEEE Trans Educ 32(4) (1989), 417−429.
[9] R. Inigo and J. Morton, Simulation of the dynamics of an industrial robot, IEEE Trans Educ, 34(1) (1991), 89−99.
[10] T. Angelo and K. Cross, Classroom assessment techniques, Jossey-Bass Publishers, 1993.
[11] L. Bautista, J. Tinbergen, and A. Kacelnik, To walk or to fly? How birds choose among foraging modes, Proc Natl Acad Sci 98(3) (2001), 1089−1094.
[12] G. Mobus, Foraging search: Prototypical intelligence, American Institute of Physics Conference Proceedings, Vol. 517, 2000, pp 592−605.
[13] E. Østergaard, G. Sukhatme, and M. Mataric, Emergent bucket brigading: A simple mechanism for improving performance in multi-robot constrained-space foraging tasks, 5th International Conference on Autonomous Agents, 2001.
[14] P. Sriyotha and A. Campbell, A thermal based approach to simulating the foraging problem, Final Report, MAE 298-2, UC Davis, June 2003 (unpublished).
[15] F. Ganji and G. Nagai, Development of robotic foraging simulator, Final Report, MAE 298-2, UC Davis, June 2003 (unpublished).
[16] K-Team, Khepera II Reference Manual, 2000.
[17] J. Spurling and C. Moore, Simulated foraging behavior in mobile robots applying hybrid architecture and a global field gradient technique, Final Report, MAE 298-2, UC Davis, June 2003 (unpublished).
[18] J. Tran and M. Cohen, Simulation for an autonomous foraging robot using odor detection sensors, Final Report, MAE 298-2, UC Davis, June 2003 (unpublished).
[19] N. Giannini and L. Hargreaves, A glimmer of light in the darkness: Using photodiodes to detect and collect objects of varying light intensities, Final Report, MAE 298-2, UC Davis, June 2003 (unpublished).
[20] S. Engle and S. Whalen, Autonomous multi-robot foraging in a simulated environment, Final Report, MAE 298-2, UC Davis, June 2003 (unpublished).

BIOGRAPHY



Sanjay S. Joshi is Assistant Professor of Mechanical and Aeronautical Engineering at the University of California, Davis. He obtained a BS from Cornell University in 1990, and a PhD in Control Systems from UCLA in 1996. From 1991 to 1994 and 1996 to 2000, he was a Member of the Technical Staff in the Guidance and Control Analysis Group at the NASA Jet Propulsion Laboratory, Pasadena, CA. From 2000 to 2001, he was Visiting Assistant Professor of Engineering at Harvey Mudd College, Claremont, California. In 2001, he joined the Department of Mechanical and Aeronautical Engineering at UC Davis. He currently directs the Robotics, Autonomous Systems, and Controls Laboratory, where he conducts a number of research projects in the theory and application of autonomous systems.