Design of a Cooperative Autonomous Mobile Robot System at the Undergraduate Level

Bradley E. Bishop
Weapons and Systems Engineering, United States Naval Academy

Abstract

This paper describes an undergraduate-level design project in a course on autonomous mobile robot systems. The project is intended to allow a great deal of latitude in implementation and to promote teamwork and integrated design methodologies in a framework that is both instructional and interesting. The technical challenges of the project include limited bandwidth communications, cooperative multi-agent algorithms, data storage and transmission and physical system design/control. Additionally, the project is structured in such a way as to provide the students experience in organizing large teams of cooperative designers and working with small task-dedicated design teams. This design project was a subcomponent of a course in Autonomous Robot Design in the Systems Engineering Department at the United States Naval Academy.

1. Introduction

Mobile robotics is a multidisciplinary field with a broad range of application domains and focus areas[1]. As a test domain, the construction and development of mobile robot applications is both motivational and highly instructional, allowing students to gain experience in algorithm design, computer interfacing, sensor selection and development, physical structure design, and control. A broad range of mobile robotics projects have been successfully implemented in undergraduate education over the last several years, from MIT's famous 6.270 course[9] to esoteric competitions such as the BEAM robot olympics[10].

Traditionally, mobile robotics courses at the undergraduate level have focused on issues in the construction and programming of these devices for tasks that rely on the use of either pre-defined map-based techniques or reactive architectures. Recently, development of a navigation map through exploration has become a primary goal of many mobile robots[2], and cooperation between individual robots is an increasingly accepted method for generating complex system behaviors and capabilities[1,12,13].

In the Autonomous Robot Design course in the Systems Engineering Department at USNA, we have developed a cooperative mobile robot design project that emphasizes a variety of issues that are germane to the design and implementation of real robot systems. The project requires the development of a "swarm" of autonomous robots that must cooperatively generate a map of a room that may later be used for navigation through either artificial potential field or approximate cell decomposition methods[7].

Teams of students must address issues in design for dead reckoning accuracy, data storage and transmission, search behaviors, data synthesis/representation and control This project emphasizes issues beyond the traditional single-robot approach to mobile systems, requiring design of an entire array of mobile robots and an integrated communication and behavior strategy. Other potential topics include distributed sensing and functionality, centralized / decentralized control of robot swarms and advanced map-based path planning. Interfacing between the robots and a PC base station allows units with simple and inexpensive processors to be coupled to the computational power of a centralized platform for data synthesis and dissemination.

The remainder of the paper is organized as follows. In Section 2, we offer an outline of the design project and discuss the test domain selected. In Section 3, we discuss the physical design challenge associated with this project. Section 4 covers algorithm development tasks and learning objectives, while Section 5 addresses the importance of feedback control in mobile robotics. In Section 6, a brief consideration of communications issues is provided. Section 7 offers an analysis of the teamwork-developing properties of this project. Section 8 includes our conclusions.

2. Problem Statement

The principle of this project is to develop a multi-robot cooperative map-building system that can communicate with a local network interface (such as a PC) to generate maps of the environment for navigation. The specifics of the project can be modified to fit a particular curriculum and focus, but issues in construction, data handling, cooperation and communication are all fundamental to the project framework.

The trial run of this project limited the test domain to a cell-based representation of the environment, with an *a priori* segmentation of the region to be searched. This type of approach would be useful in mapping structured environments, such as warehouse rooms, whose external dimensions are known without prior knowledge of the exact contents of the room and their configuration. Additionally, the same sort of algorithms apply to the case of searching an open area (such as a mine field) for a free path, since the limits of the search domain can be set from topographical maps.

The limitations for the design in the trial run were that the system was to be made entirely from elements provided with the LEGOs Mindstorms Robotic Invention Kit. This kit is an outgrowth of the 6.270 course at MIT, and offers a great deal of functionality for relatively low cost. While the provided programming environment is not sufficient for data-intensive mapping tasks, LEGO does provide a Software Development Kit that allows the use of Visual Basic and gives access to the low-level commands utilized by the Hitachi 8 processor that is the heart of the LEGO RCX control unit.

That the Mindstorms kit provides substantial physical prototyping tools is not in question. There is some debate, currently, over the usefulness of the controller for complex mobile systems[3]. The kit provides serial robust IR communication, motor drivers, shaft encoders and a relatively standard 8 bit microprocessor, but suffers from a poor data storage architecture and a limited command set (even when using the SDK). While it is true that the programming language is limited (especially with regards to data structures), a number of alternative compilers and programming environments have been developed for free distribution[11], allowing the RCX to be programmed in a variety of ways.

The implications of the selection of the RCX as the microcontroller for this project will be discussed in the following sections. It is important to note, however, that very few of the fundamental concepts of this project are in any way affected by the use of the RCX over standard mobile robot microcontrollers such as the HC11 series, Basic Stamps, the Handy Board, etc. As a proof-of-concept, the students used the limited storage on the RCX to generate cell-based maps of the regions of interest. Each search area was broken down into cells of some predetermined size (see Section 3), which had to be visited or have it determined that they could not be reached (i.e., they contained an obstacle or are unreachable due to obstacles).

3. Physical Design

One of the fundamental principles of a map-building project is that accuracy in mapping requires accuracy in dead reckoning (the process of using information about the motion history of the vehicle to determine its coordinates). Recently, many roboticists have begun to look at dead reckoning as the useful tool that it is, rather than the hindrance to robotic tasks that it has been labeled by may proponents of the behavior-based architecture for mobile systems. A minesweeper robot, for instance, might need to make sure that every inch of its search domain has been investigated, and that the exact location of every mine is known. This requires some form of dead reckoning.

When designing a mobile system to search for obstacles, the physical design is essential in two distinct ways. Primarily, the physical design determines the accuracy of the resulting map, based on the gearing of the drivetrain and resolution of the encoders that will be used for dead reckoning. Secondarily, the design determines the complexity of the decomposition, assuming that the search robots will be tasked with path planning using the generated map. If the robot is asymmetrical, or has a nonzero turn radius, the required accuracy and resolution of the map may dramatically increase (as the *configuration space* of the robot increases in dimension)[7]. In the case of an asymmetrical robot whose position *and* orientation vary, the configuration space in which planning must be carried out, and thereby, in which mapping must be completed, is three dimensional. If the students generate a robot that is symmetric with zero turning radius, a path plan can be implemented using only a two-dimensional configuration-space map. Thus, we see that physical design determines both the feasible resolution of the map *and* its required complexity.

Finally, we note that the project requires some form of sensing of obstacles; both impact-based and ranging-type sensors are acceptable for this project, based on the learning objectives to be emphasized. In the trial run, the students used only touch sensors, although ranging sensors such as the Sharp GP2D12 IR ranging sensor or Polaroid sonic range detector could allow for better navigation and more precise mapping at the cost of increased interface complexity, higher data density and more potential crosstalk between various robots.

4. Algorithms

In order for the designed system to autonomously carry out a mapping procedure, a search algorithm must be generated in such a way that all areas are covered or isolated. Furthermore, it is essential that each robot maintain at least a local environmental map in order to be able to effectively search without repeated attempts to pass through or near an already detected object.

Algorithmic flowcharts are a good tool for getting engineering students, especially those with limited coding experience, to think first about the *algorithm* rather than the implementation. Details of platform and compiler can be addressed after a suitable algorithm (flowchart) has been generated. In the trial run of this project, the students were given a flowchart assignment only *after* they had been tasked with the project for a week. In this way, they immediately saw the benefit of the technique over direct coding approaches by realizing where their previous attempts would and (in some cases) would not work.

One of the immediate limitations of a cell-based approach to the mapping problem (such as that utilized in the test run, see Section 2) is the necessity of storing the map. Storing obstacle locations, map coordinates and lists of visited cells requires memory and free space which the RCX has in short supply (with only 32 word-length variables for data storage together with limited additional memory). The data structure selected for the trial run of this project had to be able to store not only obstacle cells, but also information about cells visited vice those about which nothing is known. This required data packing, in which one variable stored information about multiple cells, forcing the students to investigate the memory structure of the selected processor and develop algorithms for inserting and extracting data regarding each cell. Identical storage and retrieval problems will arise for any microcontroller-based system, given a large enough test cell and small enough required resolution. As a testbed, the RCX proved more than sufficient to develop fundamental concepts.

The cooperative nature of the project is one of the most intriguing aspects, requiring careful and systematic characterization of subsystem performance, required communication protocols and an analysis of the benefits of high-level supervisory planning vice agent-level distributed control. In the trial run, the mobile agents cooperated when transmitting data, but worked in algorithmically bounded and distinct sub-regions of the environment, reducing the complexity of the agent-level algorithms. This aspect can be enhanced by requiring *optimal* search patterns, under which some additional communication and path planning would be required.

5. Control

One major aspect of algorithmic design that is typically left underdeveloped in mobile robotics projects is that of control. It is well known that differential-drive style robots suffer from a drift phenomenon, in which equivalent voltages applied to each of the two drive motors results in two different shaft angular velocities, causing the robot to arc in one direction[5]. Typical methods for dealing with this difficulty rely on on-line calibration, by which a voltage offset is determined for one of the drive shafts such that the system moves along a roughly straight direction. This type of manipulation works well for behavior-based systems, but experience has shown that, when using dead reckoning, closed-loop feedback control[4] is the method of choice. Feedback control allows the system to compensate for variation in contact friction, shaft loading, battery supply and other factors that can influence the performance of the drivetrain.

There are two fundamental styles of control that can be applied for a differentially-driven system:

Shaft Control: The control system monitors each shaft and makes modifications to the power supplied to that drivetrain in order to cause the system to track a desired time profile (reference trajectory). Each wheel is controlled separately.

Position Control: Using the shaft encoder information, adjustment to the power for each wheel is controlled through a process involving the estimated position and heading of the robot. The whole system is controlled to track a desired trajectory.

Position-based control is the optimal choice for mobile robot trajectory tracking, as transients in the tracking profile of a single wheel do not permanently affect the overall system. Under shaft control, the robot is free to rotate (change its heading) as external influences modify the system behavior and the robot tracks the reference. An example of this situation is given in Figure 1, in which each independent wheel controller is a proportional-derivative (PD) type designed to track the trajectory $X(t) = 2*t$ (where $X(t)$ represents the distance traveled by the shaft). The simplified equations for the system are given by:

$$\dot{w}_1 = K_p(2 - w_1 r) + K_d(2*t - \int_0^t w_1 r dt) + \mathrm{K} w_1$$

$$\dot{w}_2 = K_p(2 - w_2 r) + K_d(2*t - \int_0^t w_2 r dt)$$

$$\dot{\theta} = (w_1 - w_2) / d$$

where w_i is the angular velocity of drive shaft i, θ is the heading angle (measured from the horizontal axis to the centerline of the robot), K a disturbance coefficient, $r = 4$ cm is the radius of the wheels $d = 10$ cm is the distance between the wheels, and $K_p = 16/r$ and $K_d = 8/r$ are control parameters. The values for K_p and K_d are selected to provide good response, in accordance with the principles of control system design[4]. Note that, for simplicity, we have ignored the more complex dynamics of the system.

We assume that the robot is aligned with the desired direction to start with, so that the shaft controllers each try to guarantee that the shaft to which they are attached has moved exactly 2*t meters at time t. To demonstrate the effect of an external disturbance, we add a damping term to the acceleration of the left wheel between t = 2 sec and t = 3.5 sec by setting K to -0.05 over this interval and 0 elsewhere. The result is that the system turns during this interval, but recovers heading after the disturbance is removed. Note, however, that the robot is now no longer on the Y = 0 line.

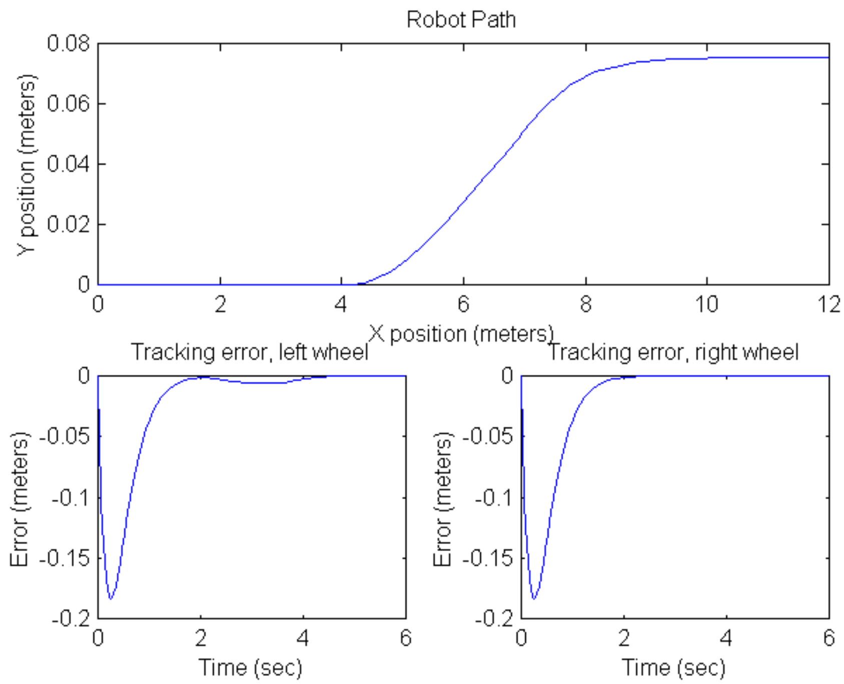

Figure 1. Tracking performance for a differentially-driven robot using shaft control.

The type of difficulty shown above can result from environmental characteristics, such as surface and terrain features, but can also arise from uneven loading on the drive shafts, varying motor characteristics, shaft binding and other factors. By utilizing an estimate of the distance of the center of the robot from the desired straight-line trajectory (and its heading, as well), this type of difficulty can be overcome (see [5] for an architecture, [4] for details of control design). Appropriate control techniques, parameter tuning and architectural selection can be included in the project in order to demonstrate the need for multidisciplinary design (combining mechanical design, software, hardware and control theory). This was not implemented in the test run, as the RCX controller allows only eight distinct power levels to be applied. These types of analysis and design are well suited to implementation using the HC11 and other, more powerful microcontrollers.

6. Communications

The final stage of the project was to develop a communication protocol for relaying information about the environmental state to the base station. The learning objectives that are covered here include fundamental communication designs for robust data transmission from multiple agents to a single base unit. This requires some form of transceiver selection and a handshaking algorithm.

In the implementation of the trial run, the students used the LEGO Mindstorms IR transceiver tower to relay bytes of data from the individual mobile units to the base station. The communication protocol is 8-bit serial across the IR channel and is bi-directional. No two mobile units can communicate with the base station simultaneously, so some form of hand-shaking and turn-taking is required. Additionally, packed data fills a word-length register, and so must be transmitted and manipulated piecewise.

Any wireless communication setup requires that the students characterize the transmission and reception characteristics of the transceivers. This requires an analysis of the data channel given a particular environmental configuration, which may require maneuvering of the mobile units to allow unimpeded data relay. This is a more sophisticated case than was handled in the trial run, as it requires the mobile units to perform on-line motion planning, which we leave for a base station to perform in the implemented example (due to storage restrictions on the mobile units).

7. Organization and Design Skills

The development of a cooperative mobile robot system is well suited to large team efforts, with small subgroups tasked with component designs. In the trial run, ten students were assigned to each group and tasked with generating at least three cooperating mobile robots and an integrated base station mapping system. In both test groups, the organization broke down into three subgroups. One subgroup designed and tuned the physical system performance, one developed the mobile unit search and storage routines, and the final subgroup designed a PC interface and communication protocol.

The students involved discovered rapidly that each subgroup needed to be fully updated on the design decisions of the other subgroups. It was also determined that the physical design (considered the most straightforward portion of the system) was also the most sensitive to variations. Several students became “roamers” after contributing their subcomponent, moving through other subgroups and keeping each updated on the overall flow of the project. Some students did complain that they felt underutilized, highlighting the danger of allowing a large group to self-organize.

A valuable lesson taught through this project was that, when a student is a member of a design team, they should make an effort to become integrated into the project so that at the completion of their initially determined duties they can assist with secondary tasks. A crucial component to successful utilization of manpower resources during a project of this nature is a careful project flow timeline, showing those subtasks to be completed in parallel vice those that must (or should) be completed in series. In the trial run, the students were not instructed to estimate the time required for a given subcomponent, and so found some underutilized resources available at the end of the project.

8. Conclusions and Observations

This project addresses fundamental issues in mobile robotics traditionally left for the graduate level, and does so in an intuitive and engaging way. The lessons learned apply to mobile robot designs for surveillance, reconnaissance, exploration and map-building for mobile systems. The project is broad enough to allow focus on a variety of sub-components while guaranteeing some level of overall system integration experience. This project is well suited to both engineering and computer science curricula, and can be utilized as a term project or a multi-week laboratory, depending on the complexity of the required solution and available hardware.

Bibliography

1. Arkin, R., *Behavior-Based Robotics*, The MIT Press, Cambridge, MA., 1998.
2. Anousaki, G. C. & Kyriakopoulos, K. J., "Simultaneous Localization and Map Building for Mobile Robot Navigation," *IEEE Robotics and Automation Magazine*, Vol. 6, No. 3, pp. 42 - 53, September 1999.
3. Auyeung, T., "LEGO Mindstorms: A Technical Perspective," *Robot Science & Technology*, Vol. 1, No. 6, pp. 42 - 47, September/October 1999.
4. Dorf, R. C. and Bishop, R. H., *Modern Control Systems*, Eighth Edition, Addison-Wesley, Reading, MA, 1998.
5. Jones, J. L., Seiger, B. A. & Flynn, A. M., *Mobile Robots: Inspiration to Implementation (second edition)*, A K Peters, Natick, MA., 1999.
6. Khatib, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R. & Casal, A., "Coordination and Decentralized Cooperation of Multiple Mobile Manipulators," *Journal of Robotic Systems*, Vol. 13, No. 11, pp. 755 - 764, 1996.
7. Latombe, J.-C., *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.
8. McComb, G., *Robot Builder's Bonanza: 99 Inexpensive Robotics Projects*, TAB Books, New York, 1987.
9. URL: *http://www.mit.edu:8001/courses/6.270/home.html*
10. URL: *http://nis-www.lanl.gov/robot/*
11. URL: *http://www.crynwr.com/lego-robotics/*
12. Wagner, I. A., Lindenbaum, M. & Bruckstein, A. M., "Distributed Covering by Ant-Robots Using Evaporating Traces," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 5, pp. 918 - 933, October 1999.
13. Yamaguchi, H., "A Cooperative Hunting Behavior by Mobile Robot Troops," *International Journal of Robotics Research*, Vol. 18, No. 9, pp. 931 - 940, 1999.

BRADLEY E. BISHOP

Bradley E. Bishop is an Assistant Professor in the Weapons and Systems Engineering Department at the United States Naval Academy. He received the B.S. degree in Electrical Engineering from Michigan State University in 1991, and the M.S. and Ph.D. degrees in EE from the University of Illinois at Urbana-Champaign in 1994 and 1997, respectively. His research interests include robotics, nonlinear and hybrid control, and computer vision.