Proceedings, 6th World Multiconference on Systemics, Cybernetics and Informatics, July 14-18, 2002, Orlando, Florida, USA

Path following for a poorly controlled mobile robot

Kevin M. NICKELS
Department of Engineering Science, Trinity University
San Antonio TX USA 78212

Abstract

This paper describes the hardware and software architecture used for a car-like mobile robot designed and implemented by an undergraduate design group. This project is built from a radio controlled truck and is designed to be a low-cost modular odometric path follower. The environment model is assumed to be known. The controlling algorithm makes explicit accommodation for the fact that the robot seldom steers in exactly the direction commanded. The hardware architecture is designed to use commonly available off the shelf parts and to be expandable for use by future design groups. The final robot plans and follows a nonholonomic path with errors that can be largely attributed to odometry. Therefore, the poor controllability of the platform is mitigated, and this low-cost modular robot can be used in undergraduate research and design settings in the place of a professional research robot.

Keywords: Car-like Mobile Robot, approximate cell decomposition, layered architecture

1 Introduction

Car-like mobile robots impose an interesting set of problems on researchers interested in path planning and path following work. The use of cheap car-like robots to implement path following imposes another, not completely overlapping, set of constraints on the design. The educational environment in which these designs were developed imposed several other constraints on the designs. We will investigate each of these sets of constraints, some implications of them, and some possibilities for dealing with them in the sections below.

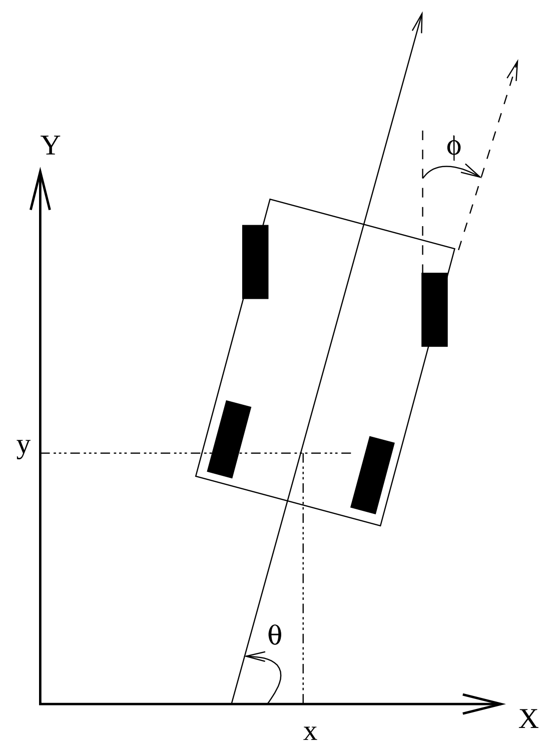

Figure 1: A car-like robot

Motion planning is a very mature area of research [1], although there are certainly open problems to be solved [2]. The major constraint imposed by the consideration of nonholonomic system is that of a limited steering angle (ϕ in Figure 1) and therefore limited turning radius. There is much room for investigation even within the restricted realm of path planning for wheeled mobile robots of the car-like variety [3]. While the planning of a nonholonomic path is necessary for path following on a sloppy robot, it is not sufficient.

Our reference to "sloppy robots" refers primarily to the steering assembly. The steering assembly used in radio controlled trucks typically consists of a steering servo connected to a linkage or linkages that actuates the wheel. The servo is driven by a pulse width modulated (PWM) signal normally supplied by a radio receiver or (in our case) by the MIT Handy-Board. The mapping between pulse width and steering angle is quite noisy. In other words, the robot only moves the the approximate direction commanded.

This and some other inaccuracies are caused by non-optimal or unmodeled elements of the robot. These are often exacerbated in low-cost robots. Robots often cannot instantaneously assume an arbitrary turning curvature [4]. The wheels may slip due to the lack of a rear differential and/or an overpowered drive servo, since the radio-control cars used as bases for these robots are manufactured for high-torque, high-speed use. The wide tires don't have a well-defined contact point with the ground. Further, the robot parameters such as the width of the wheelbase [5] are ill-defined or vary from robot to robot. All of these inaccuracies contribute to poor controllability of the robot. They can be mitigated by more complex kinematic and dynamic modeling. They can also be mitigated (as in our case) by accepting the fact that the models are only approximate and working on increasing the robustness and observability of the higher levels of the control system.

Some design constraints are imposed on this project by the educational setting, which was a senior design class in an engineering science undergraduate program. This interdisciplinary capstone design course is described in more detail in [6]. These constraints include a total system cost limit of $1000, the requirement to use as much commercial off-the-shelf hardware as possible, and an easily extendible (by undergraduates) hardware and software architecture.

Many of the constraints mentioned above can be mitigated by the use of an absolute positioning system such as landmarks [7] or a global positioning system. However such techniques were not used in this design because they unacceptably add to the cost and/or complexity of the project.

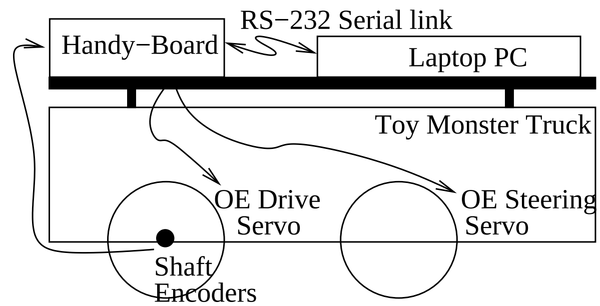

Figure 2: Hardware Overview

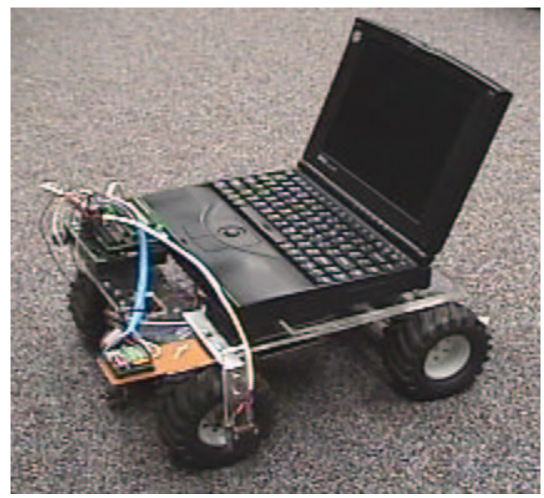

Figure 3: The "sloppy" path-following robot

2 Hardware Architecture

The hardware platform for the project is constructed from a toy monster truck[1] or similar vehicle. A truck was chosen over a car due to payload considerations and over a tank due to the additional problems introduced by track steering. The original equipment (OE) suspension was stiffened and a platform was attached to the frame enabling the addition of a standard laptop computer (PC) and an MIT Handy-Board [8, 9] for control of the OE drive and steering servo motors. In addition, shaft encoders were attached to the rear wheels to gather odometry information. A block diagram and photograph of the robot are shown in Figures 2 and 3 respectively.

The Handy-Board handles the sensors (one

[1]The Traxxas Stampede was used in this implementation.

shaft encoder for each of the two rear wheels) and control of the two OE servos (one for drive and one for steering) on the robot. The PC handles the mapping and motion planning algorithms and issues low level motion commands to the Handy-Board. Motion commands and encoder data are passed between the laptop and the Handy-Board via a standard RS-232 serial link. Each component has a separate battery for power. This modular design allows the Handy-Board to do the low-level input/output (this is the strong point of the Handy-Board) and the laptop to be programmed in standard high-level object oriented languages.

3 Software Architecture

The mapping and path planning is done in the laptop computer using the algorithm shown in Figure 4. It begins with a global path planner (GPP). Any standard motion planning routine can be used here, preferably one that makes explicit accommodations for the kinematic constraints (i.e. nonholonomy) of the robot. We used a variation on approximate cell decomposition (ACD) originally proposed in [10] and explained lucidly in [1]. In this algorithm, the adjacency matrix of ACD is modified to account for the nonholonomic properties of the robot. The output of this algorithm is a series of waypoints in configuration space for the robot to follow. These waypoints are in free space.

The waypoints output by the GPP module can be connected into a nonholonomic path in several ways. One common method [1, 3] is to connect the waypoints with arcs and/or straight lines. This is the method we follow in our implementation.

The executive module in Figure 4 is responsible for updating the demands on the control system. It receives continuous input from the localizer as well as the set of original desired waypoints. Pears [4] develops control schemes to reduce the peak demand on the steering controller, but a simpler algorithm is used in this implementation that assumes an instantaneous change of steering angle.

As mentioned in Section 1, one characteristic of this type of vehicle is the tendency for the vehicle to execute a steering command that is similar to

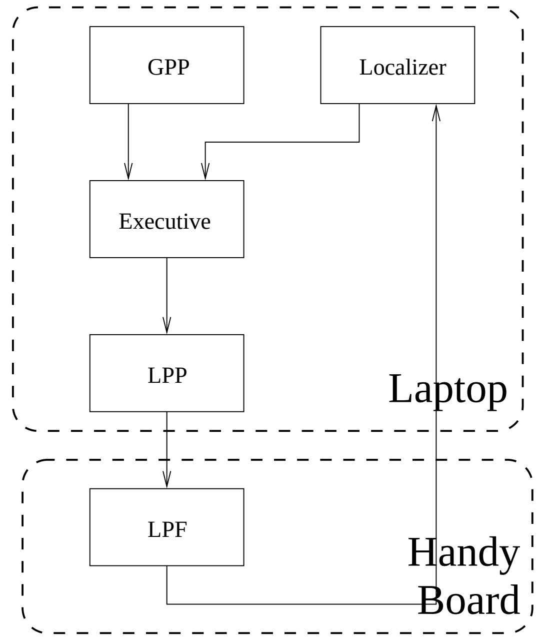

Figure 4: Block Diagram of the path-follower

the one commanded, but not identical. The executive module tracks the progress of the robot through odometry (via the localizer module) and makes adjustments to the demand paths to allow the robot to continue its path-following even though a waypoint has not been achieved.

Once the desired waypoints have been updated, the local path planner (LPP) computes a steering angle and time to goal for the next path segment. The path following is done in discrete segments to reduce demands on the low-speed communications link between the Handy Board and the laptop. This link is used to convey commands as well as to record odometry information. A project for future groups will be to modify this algorithm to implement continuous path following.

The local path follower (LPF) module is located in the Handy Board and is responsible for the low-level outputs. This module actuates the drive servo and monitors the wheel encoders to estimate when the local segment will complete. When the segment is complete, this module sends the odometry information to the localizer module on the laptop and awaits another movement command.

Table 1: The Hardware Budget

Laptop	\$ 1000[2]
Handy-Board	\$ 400
RC Vehicle	\$ 160
Wheel Encoders	\$ 80
Electronic Speed Controller	\$ 60
Total	\$ 1700

[2] The laptop has been amortized across both group's budgets.

4 Budget

One of the constraints imposed on the project by the senior project course requirements is that of budget. The budget for each design group is \$1000. The path-following robot described in this paper is actually the amalgamation of two design projects, an environment modeling robot in the 1999-2000 academic year and a path-following robot utilizing the majority of the existing hardware platform in the 2000-2001 academic year. The basic hardware was built by the former group (their software was not used in the reported project), and the software algorithm was contributed by the latter group. The cost breakdown for materials used in the final robot is shown in Table 1.

5 Path Following Results

The robot exhibits the expected artifacts of odometric path following. That is, there is a constant and increasing error in the absolute location of the robot compared with the computed location of the robot. However Figure 5 shows that the primary source of error, that caused by steering inaccuracies, has been corrected. This error comes from the fact that the robot rarely goes in exactly the direction that is commanded. Some explanations for this behavior are backlash in the steering linkages, the OE servo used in the steering mechanism, and static friction between the oversized and underinflated wheels and the laboratory flooring.

By modularizing the software into an executive that monitors the progress of the robot and a follower that carries out commands, the executive can modify future commands to mitigate the impact on the overall path-following task without a complete re-plan.

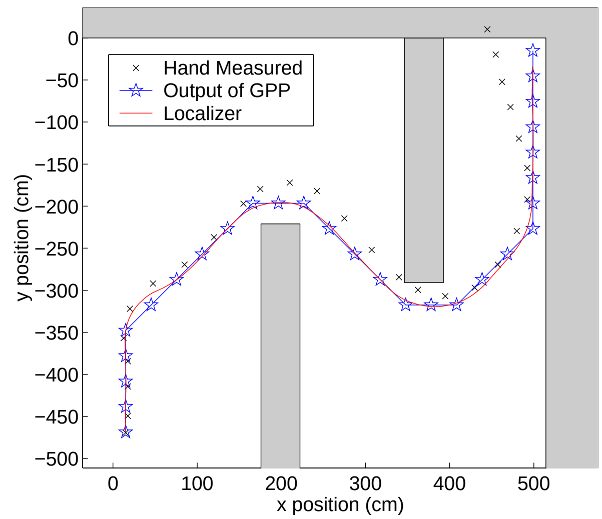

Figure 5: Example Path

6 Conclusions

A robot designed by two undergraduate senior design groups was presented. The hardware platform is constructed from a low cost but poorly controlled radio controlled truck. An MIT Handy-Board and commodity laptop computer provide the device control and planning capabilities, respectively. The controllability problems of the truck are mitigated by a layered software approach, with an executive module monitoring both the actual progress and the requested motion commands of the robot. Future commands are built from this information, reducing the path following error to that expected by using odometry with a much more controllable platform.

This platform shows promise for mobile robotic projects that operate under a prohibitive (for typical mobile robot applications) cost ceiling. It maintains acceptable and repeatable performance through the use of supervisory software. It utilizes readily available off the shelf hardware and is reasonably easy to program and modify. All of these features render the system acceptable for use in such settings as senior design project or undergraduate research, where nonspecialists need to access and modify the hardware and software with minimal setup time and startup cost.

Acknowledgments

The author would like to thank the two design groups who contributed to the design of the robot: Daniel Houy, Andrew Leininger, Jeff Liddle, Michael Sutton, Brian Deaton, Jeff Dickerson, and Jared Newton.

References

[1] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.

[2] M. Salichs and L. Moreno, "Navigation of mobile robots: open questions," *Robotica*, vol. 18, pp. 227–234, 2000.

[3] W. Wu, H. Chen, and P Woo, "Time optimal path planning for a wheeled mobile robot," *Journal of Robotic Systems*, vol. 17, no. 11, pp. 585–591, 2000.

[4] N. Pears, "Mobile robot tracking of pre-planned paths," *Advanced Robotics*, vol. 15, no. 1, pp. 97–107, 2001.

[5] M. Corradini and G. Orlando, "Robust tracking control of mobile robots in the presence of uncertainties in the dynamical model," *Journal of Robotic Systems*, vol. 18, no. 6, pp. 317–323, 2001.

[6] P. Giolma, F. Aminian, and D. Ibaroudene, "An autonomous robot — the ideal design project?," *Computers in Education Journal*, vol. 10, no. 1, pp. 62–67, 2000.

[7] A. Lambert and T. Fraichard, "Landmark-based safe path planning for car-like robots," in *Proc. IEEE International Conference Robotics and Automation*, 2000.

[8] F. Martin, *Robotic Explorations: A Hands-on Introduction to Engineering*, Prentice-Hall, New Jersey, USA, 2001.

[9] F. Martin, "The handy board web site," `http://www.handyboard.com`, Accessed Sept 18, 2001.

[10] J. Barraquand and J. C. Latombe, ," *Revue d'Intelligence Artificielle*, vol. 3, no. 2, pp. 77–103, 1989.