Robotic Navigation of Smooth Contours

by

Justin C. Moore

Submitted to the Department of Mechanical
Engineering in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science

at the

Massachusetts Institute of Technology

February 2007

Signature of Author..
Department of Mechanical Engineering
February 1, 2007

Certified by..
John Leonard
Associate Professor of Mechanical Engineering

Accepted by..
John Lienhard
Chairman, Undergraduate Thesis Committee

Robotic Navigation of Smooth Contours

By

Justin C. Moore

Submitted to the Department of Mechanical Engineering in
Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science

ABSTRACT

The goal of this work is to develop a method for robotic navigation of smooth contours depending on the current and desired locations and orientations. Efficient trajectory generation is an essential capability for many autonomous mobile robots, operating in a variety of situations such as military, medical, and home environments. In this thesis, we propose a method that is based on fitting a spline curve that passes from the initial position and orientation of the robot to a goal position and orientation. The spline is continually recomputed as the robot moves through space. This yields a simple and inefficient method for robot navigation. The method has been implemented and tested in simulation using Matlab and good performance has been demonstrated. Future work should perform experiments with this method on a real robot and should introduce obstacle detection and avoidance.

Thesis Advisor: John Leonard
Title: Professor of Mechanical and Ocean Engineering

Table of Contents

1. INTRODUCTION

This work explores the use of cubic functions for path determination in two-dimensional space for robotic navigation applications and compares the utility of cubic functions to higher order functions in this application. The problem is investigated in a simulated environment generated by Matlab. The environment developed models locomotion of a robot with two coaxial driven wheels of known separation distance with independent rotational speeds, and a third un-driven universal caster equidistant from the two driven wheels at a known separation distance from their common axis. The navigation system works to determine the desired velocities of each driven wheel which is dependant upon robot geometry and a maximum wheel velocity which would be a property of the drivers. It is important to note however, that the discussed methods for path determination can be adjusted to fit any number of vehicle layouts. The particular layout chosen is somewhat arbitrary and the geometry is used simply to accurately express visually the locomotion of the modeled robot in the simulated environment.

This work addresses a small component of the larger mobile robot navigation, motion planning, and control problem. The recent literature in these areas is well captured by Siegwart and Nourbakhsh (2004), Choset et al. (2005), and LaValle (2006). Our work is closest to methods presented in Chapter 3 of Siegwart and Nourbakhsh (2004), which describes a trajectory controller for a mobile robot to travel from an initial pose to a desired position and orientation. Our method is attractive for its simplicity, as it is based on the well-known concept of polynomial spline curves, and is expected to be a useful tool in teaching robot motion control in undergraduate subjects.

2. NAVIGATION METHOD

Path determination requires inputs of current and desired global position and orientation. In order to have a polynomial function path solution for a given set of inputs certain criterion must be met by the points and orientations. Considering a global frame that is described by Cartesian coordinates where the vehicle's current position coordinates are x_0 and y_0 and the vehicle's current orientation is θ_0. Now given a desired position at x_1 and y_1 with orientation θ_1, we can also determine a vector that denotes the relationship of the desired final position to the current position. This vector points in the direction $\theta_r = \arctan[(y_1 - y_0) / (x_1 - x_0)]$. Figure 1 shows the significant measures in the determination of the function based path.

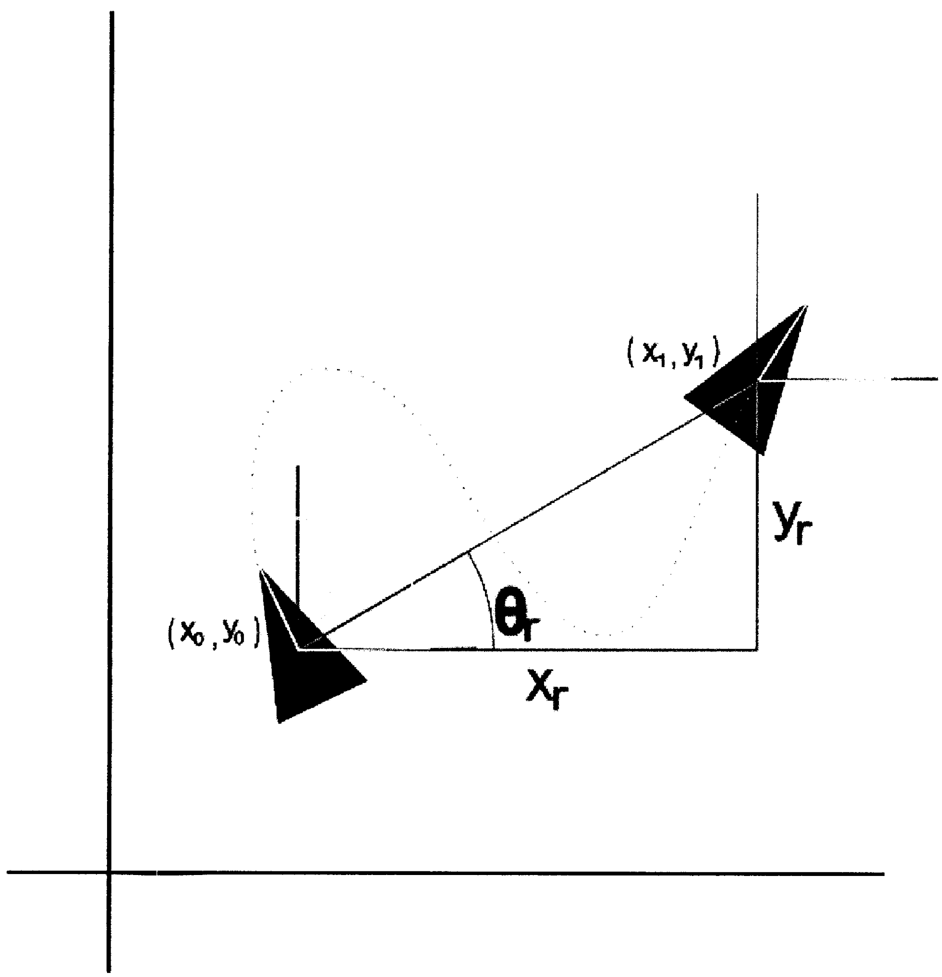

Figure 1. Model of the method in global frame.

Now the reference frame is translated to the location of the vehicle and rotated by the average of the range of the angles θ_0, θ_1, and θ_r. This rotation is done in an effort to smooth the resulting curve, making it less eccentric and reducing the total arc length. Here x_1', y_1', θ_0', θ_1', and θ_r' are the relevant measures in the new reference frame. The simplest polynomial function that can yield a solution is one of third order because minimum number of conditions to meet is four. For a function $y(x) = f(x)$, the conditions are: $y(0) = 0$, $y'(0) = \tan(\theta_0')$, $y(x_1') = y_1'$, and $y'(x_1') = \tan(\theta_1')$. The function can be simply solved for using matrix multiplication given these boundary conditions and a general form for the function y(x). In order to increase final orientation accuracy an additional boundary condition, $y''(x_1') = 0$, was added to yield a fourth order solution.

3. RESULTS

This method was tested for a series of a goal positions. Each goal position is a distance of 10 meters from the origin, with a goal orientation of zero radians and an initial orientation of $\pi/2$ radians. Figure 2 shows the path taken by the vehicle to reach to goal in each case.

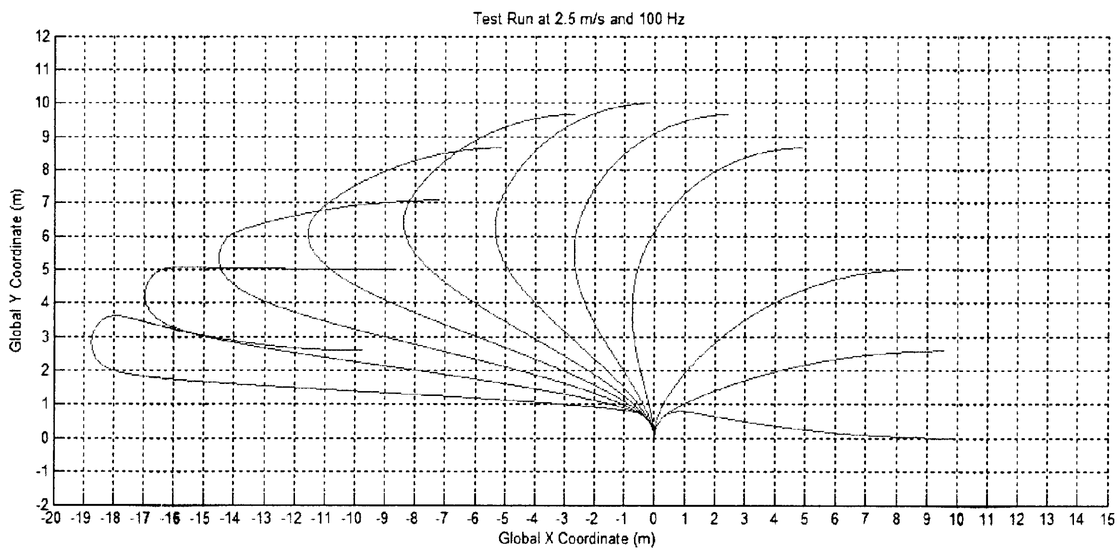

Figure 2. Resulting paths from test runs of various goal positions 10 meters away from the initial position. All goal orientations are zero radians and initial orientation is $\pi/2$ radians.

The tests were performed assuming a vehicle velocity of 2.5 m/s and motor controllers running at 100 Hz. Table 1 depicts the actual time to goal, goal orientation accuracy, and the total computation time (computation performed on 2.0 GHz Intel Centrino Duo Processor).

Table 2. Accuracy, process time, and real time of paths resulting from test runs. Vehicle velocity was 2.5 m/s and goals were 10.0 m from origin.

Relative Goal Angle (radians)	0	pi/12	2*pi/12	3*pi/12	4*pi/12	5*pi/12
Real Time (s)	4.32	4.25	4.38	4.62	5.04	5.64
Orientation Accuracy (radians)	-0.0019	0.0034	0.0073	0.0104	0.0124	0.0161
Process Time (s)	0.0940	0.0780	0.0630	0.0790	0.0780	0.0940

Relative Goal Angle (radians)	6*pi/12	7*pi/12	8*pi/12	9*pi/12	10*pi/12	11*pi/12
Real Time (s)	6.64	7.48	8.66	9.91	11.09	12.11
Orientation Accuracy (radians)	0.0127	0.0109	0.0074	0.0037	0.0002	-0.0029
Process Time (s)	0.0940	0.1090	0.1410	0.1410	0.1560	0.1880

4. CONCLUSION

This work demonstrates that the proposed method of navigation can be executed with a reasonably low computation time and with very small error to the goal orientation (less than 1°). This work does not however present solutions for all positions and orientations. In order for a polynomial function solution to be determined, there must be a reference frame in which angles θ_0, θ_1, and θ_r are all within the bounds of negative $\pi/2$ and $\pi/2$. The global frame may be rotated and translated for the purpose of this test. To simplify the problem, the global frame is translated so that the current vehicle position becomes the origin. Finally, an angle α, which describes how far the global frame is to be rotated before a polynomial function solution is determined, must be found that satisfies the necessary criteria.

Solving for a suitable α is the crux of the problem. A proposed solution was to first determining the ranges of α suitable for each of θ_0, θ_1, and θ_r denoted by α_0, α_1, and α_r. For any given angel θ, the range of angular rotations of the global frame that would make θ' in the new frame in the first and fourth quadrants is from negative $(\pi/2 + \theta)$ to $(\pi/2 + \theta)$. Considering the specifically valid ranges of α, the range of α suitable for all three is bounded on the bottom by the maximum of the lower bounds of all α, and bounded on the top by the minimum of the upper bounds of all α. Finally for the solution of a polynomial function, α is chosen to be the average of these two bounds. Even this proposed method, however, does not solve the problem completely. Future works in this area might explore patches to this issue and propose alternative solutions to cases in which a simple polynomial function can not be determined from only the boundary conditions noted above.

References

H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, "Principles of Robot Motion: Theory, Algorithms, and Implementation", MIT Press, 2005.

S. LaValle. "Planning Algorithms", Cambridge University Press, 2006

R. Siegwart and I. R. Nourbakhsh, "Introduction to Autonomous Mobile Robots", MIT Press, 2004.