NONHOLONOMIC MOTION PLANNING FOR UNDERWATER VEHICLES

by

Jean-Paul Tennant

Memorandum No. UCB/ERL M93/89

15 December 1993

NONHOLONOMIC MOTION PLANNING FOR UNDERWATER VEHICLES

by

Jean-Paul Tennant

Memorandum No. UCB/ERL M93/89

15 December 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

NONHOLONOMIC MOTION PLANNING
FOR UNDERWATER VEHICLES

by

Jean-Paul Tennant

Memorandum No. UCB/ERL M93/89

15 December 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Fuel efficient designs for submersibles typically result in vehicles that exhibit nonholonomic behavior. The purpose of this project is to develop an algorithm that generates feasible trajectories for such vehicles.

A general model for underwater vehicles is first derived using kinematic equations of motion. The resulting system is nonholonomic with drift. Vehicle performance is characterized by placing limits on the acceptable inputs.

An algorithm is then presented that takes as input a sequence of "waypoints" through which the vehicle should pass. In addition to the location of each point, the user may specify time of arrival, heading, pitch angle, and/or velocity. The resulting path satisfies the nonholonomic and performance constraints for the vehicle and is near optimal in the sense of minimizing the pathlength.

Contents

Chapter 1

Introduction

1.1 Background

Underwater vehicles, whether manned or unmanned, promise to become an invaluable tool for the understanding and effective utilization of the ocean environment. Applications include research in oceanography and marine biology, location and recovery of submerged objects, bathymetric surveying, and subsea agriculture. In addition to their military and scientific uses, manned submersibles are becoming increasingly popular as a source of recreation at resort areas. Unmanned submersibles are expected to play a growing role in industrial applications such as inspection, cleaning, and maintenance of underwater structures including oil platforms and ship hulls.

Due to the limited amount of fuel that can be carried on board, underwater vehicles must be fuel efficient in order to maximize their range. Consequently, most are equipped with a propulsion system and a hydrodynamic shape designed for efficient motion in one primary direction. Steering is accomplished through the use of control surfaces which can be positioned to cause the vehicle to turn and/or pitch. Such designs result in nonholonomic behavior. In other words, barring obstacles, the vehicle can eventually reach any position and orientation in its environment, but it may have to take a circuitous path to do so.

The movement of such a vehicle can be implemented as follows:

1. A sequence of "waypoints" through which the vehicle should pass is specified by a *high level planner*. This planner can be human or automated. In addition to the location of each point, it may be desirable to specify values for time of arrival, heading, pitch angle, and/or velocity.

2. A *motion planner* takes as input the waypoints with associated parameters and generates a trajectory. The trajectory must pass through these points as specified and at the same time, satisfy the nonholonomic and performance constraints for the vehicle. In addition, because there are generally many paths that meet such a set of constraints, the motion planner should find paths that are optimal in some sense.

3. A *motion controller* then adjusts the thrust and control surface angles as necessary to cause the vehicle track the given trajectory.

The development of automated systems for accomplishing the first of the above three tasks is highly application specific and has been addressed by various experts in the field of artificial intelligence. Much work has also been presented regarding the third problem — methods employed include sliding mode control, adaptive control, neural networks, and fuzzy logic. This project addresses the second task, that of developing a motion planner.

1.2 Overview

Underwater vehicles are typically designed to minimize the amount of sideslip experienced during a maneuver. In addition, although some have active buoyancy control systems, they are typically designed to operate at or near the point of neutral buoyancy. Consequently, for the purposes of planning a path, it is reasonable to assume that the instantaneous motion of the vehicle is always in the direction in which it is pointed.

The dynamic equations describing the motion of underwater vehicles are very complex. By making the above assumption, a model can be derived based on the kinematic properties of the vehicle rather than the dynamics. Such a model is used in the development of the motion planner presented here. The resulting system is nonholonomic.

Furthermore, in order to ensure that the model vehicle has continuous velocity, the kinematic equations are modified to include velocity as a state. The system then exhibits drift as well as nonholonomy.

Vehicle performance is specified via constants that describe the turning rate, pitching rate and acceleration capabilities of the vehicle. Due to the shape and positioning of their control surfaces, most underwater vehicles can not be reliably driven backwards. Consequently, the algorithm is formulated for vehicles that move in the forward direction only.

In order to develop the algorithm, it is first shown that, because of the way in which vehicle performance is specified, the problem can be decoupled into (1) determining the shape of the path, and (2) determining the velocity that the vehicle should have as it follows that path.

In order to determine the shape of the path, one must first decide what should be meant by an "optimal" path. Several different criteria can be used. Many motion planing schemes attempt to maximize smoothness. The approach taken here is to minimize pathlength. As a result the paths are continuously differentiable but with discontinuities in curvature. The motivation behind this approach is that, although the resulting paths can not be followed exactly by a real system (as is always the case), they can be closely approximated and will be shorter than any smoother paths, and hence better in terms of fuel usage and ability to meet time constraints.

The shape of the path is thus derived using a result of L. E. Dubins [1] for paths of minimal length in two dimensions. Through repeated application of this result, the algorithm generates paths that are near optimal in three dimensions.

Waypoints given without regard to orientation create additional degrees of freedom in selecting a path. Optimization must be carried out over these variables as well. This optimization is accomplished via discretization of the associated space.

Once the shape of the path has been determined, a velocity profile must be chosen to meet the given time and velocity constraints. In general, there are many velocity profiles that satisfy a given set of constraints. The profiles generated by the planner presented here are "trapezoidal", thereby maximizing the duration of the steady velocity portion of the trajectories and at the same time allowing the vehicle to operate at its performance limit when necessary.

Applications for the algorithm include motion planners for autonomous underwater vehicles (AUVs) and autopilots for manned submersibles. The algorithm can also be used for other vehicles that operate in three dimensional environments, namely aircraft.

1.3 Related Work

The work of L.E. Dubins [1] applies directly to generating paths of minimal length for car-like robots that move only in the forward direction. His results play a central roll in the algorithm presented here. Reeds and Shepp [14] develop a method for finding paths of minimal length for cars that can reverse as well. Murray and Sastry present a general method for steering nonholonomic systems using sinusoidal inputs [11].

The presence of obstacles complicates planning for nonholonomic systems. Jacobs and Canny [4] present a planner based on reducing the set of smooth trajectories to a set of canonical trajectories. Jacobs, Laumond and Taix [5] develop a planner for an environment with obstacles that first finds a holonomic path that avoids obstacles and then approximates this path with one satisfying the nonholonomic constraints. Mirtich and Canny [10] present an efficient planner that uses "skeleton" paths of maximum clearance based on a metric derived from the paths described by Reeds and Shepp.

Latombe gives a broad summary of robot motion planning in general in [8]. A compilation of recent research in nonholonomic motion planning is presented in [9], edited by Li and Canny. Nonholonomic planning is also treated by Murray, Li and Sastry in their upcoming textbook [13].

Chapter 2

Underwater Vehicles

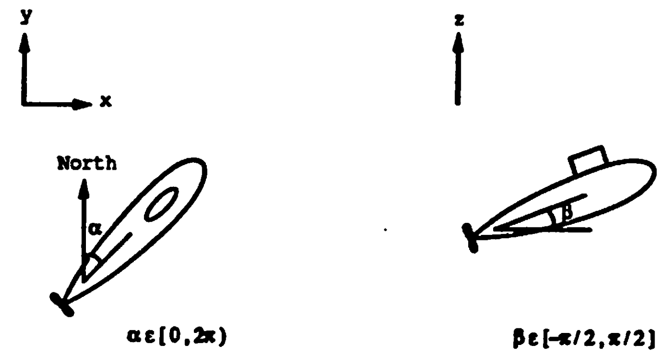
2.1 Configuration Space

The configuration of any rigid body consists of two parts: position and orientation. But for the purpose of developing a kinematic model, one need not characterize the orientation completely. An assumption will be made, namely that the instantaneous motion of the vehicle is always in the direction in which it is pointed. Thus the roll angle of the vehicle has no bearing on the direction of motion, and need not be considered.

The configuration of the vehicle can therefore be represented as:

$$X = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \end{bmatrix} \qquad X \in \mathbf{R}^3 \times S_2$$

where α and β parameterize the unit sphere, S_2. In keeping with nautical tradition, α is chosen to represent the direction in which the vehicle is pointed projected into the horizontal plane, and is referred to as the "heading". It is measured clockwise from true north (the +y-direction). The parameter β represents the the angle that



TOP VIEW SIDE VIEW

Figure 2.1: The Vehicle

the vehicle makes relative to horizontal and is referred to as the "pitch angle" (see figure 2.1).

2.2 Equations of Motion

2.2.1 Kinematics

For a typical underwater vehicle operating at or near the point of neutral buoyancy and with a propulsion system that acts in one direction only, it is reasonable to assume that the instantaneous motion of the vehicle is in the direction in which it is pointed. The kinematic equations of motion follow as a direct result of this assumption.

Since motion is allowed in only one direction, there is a plane of disallowed directions. Choosing two independent directions in this plane (see figure 2.2) gives:

$$n_1 \cdot \overline{v} = 0 \qquad and \qquad n_2 \cdot \overline{v} = 0$$

where

$$n_1 = \begin{bmatrix} \cos\alpha \\ -\sin\alpha \\ 0 \end{bmatrix} \qquad n_2 = \begin{bmatrix} \sin\alpha\sin\beta \\ \cos\alpha\sin\beta \\ -\cos\beta \end{bmatrix} \qquad \overline{v} \triangleq \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

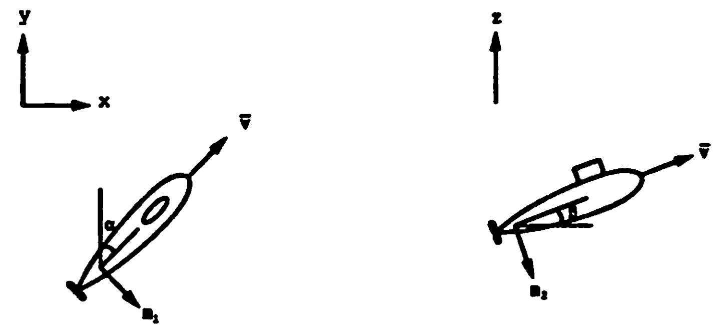


Figure 2.2: Constraints on the Motion of the Vehicle

These constraints can be summarized in matrix form as:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 & 0 \\ \sin\alpha\sin\beta & \cos\alpha\sin\beta & -\cos\beta & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \overline{0}$$

Taking the null space of the constraint matrix gives a set of vector fields describing the allowable motions:

$$g_1(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad g_2(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad g_3(X) = \begin{bmatrix} \sin\alpha\cos\beta \\ \cos\alpha\cos\beta \\ \sin\beta \\ 0 \\ 0 \end{bmatrix}$$

The system can therefore be written as:

$$\dot{X} = u_1 g_1(X) + u_2 g_2(X) + u_3 g_3(X)$$

or equivalently,

$$\begin{aligned}
\dot{x} &= u_3 \sin\alpha \cos\beta \\
\dot{y} &= u_3 \cos\alpha \cos\beta \\
\dot{z} &= u_3 \sin\beta \\
\dot{\alpha} &= u_1 \\
\dot{\beta} &= u_2
\end{aligned} \tag{2.1}$$

where u_1, u_2, and u_3 are the inputs. It can be seen from the equations in (2.1) that the inputs have the following physical meanings:

u_1 – turning rate
u_2 – pitching rate
u_3 – forward velocity

2.2.2 Nonholonomy

The nonholonomic nature of the system can be seen by analyzing the associated vector fields, g_1, g_2, and g_3. The span of these vector fields defines a *distribution.* The distribution assigns a subspace of allowable motions to each point in the vehicle's configuration space. Given a distribution, Δ, the *filtration* of the distribution is defined as the set of distributions generated by iteratively taking Lie brackets of its elements as follows:

$$\begin{aligned}
G_1 &= \Delta \\
G_i &= G_{i-1} + span\{[g,h] : g \in G_1,\ h \in G_{i-1}\}
\end{aligned}$$

For the underwater vehicle, the filtration is given by

$$\begin{aligned}
G_1 &= span\{g_1, g_2, g_3\} \\
G_2 &= span\{g_1, g_2, g_3, g_4, g_5\} \\
G_3 &= span\{g_1, g_2, g_3, g_4, g_5\} \\
&\ \ \vdots
\end{aligned}$$

where g_4 and g_5 are given by

$$g_4 \triangleq [g_1, g_3] = \begin{bmatrix} \cos\alpha\cos\beta \\ -\sin\alpha\cos\beta \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad g_5 \triangleq [g_2, g_3] = \begin{bmatrix} -\sin\alpha\sin\beta \\ -\cos\alpha\sin\beta \\ -\cos\beta \\ 0 \\ 0 \end{bmatrix}$$

Further analysis shows that:

$$\begin{aligned} &rank\ G_1 = 3 \\ &rank\ G_i = 5 \quad \text{for } i \geq 2, \quad \beta \neq \pm\pi/2 \end{aligned}$$

Because the filtration reaches full rank after one iteration of Lie bracketing, the system is said to have *degree of nonholonomy* equal to one. The growth in rank of the filtration is described using a *relative growth vector* which, in this case, is [3 2]. *Chow's Theorem* states that a system for which the filtration achieves full rank is controllable. Hence, mathematics confirms the intuition that the vehicle can be driven to any point in its configuration space. (See [9] or [13] for further information on nonholonomic systems.)

2.2.3 Improving the Formulation

There is a practical problem with the kinematic formulation given by the equations in (2.1) in that the velocity of the vehicle appears as an input. This leads one to believe that it can be chosen arbitrarily as necessary to control the system. Realistically, this is not the case since the velocity must always be continuous for a real system.

We can modify the system to enforce this fact by making the velocity a state and choosing u_3 as its derivative:

$$\begin{aligned} \dot{x} &= v\sin\alpha\cos\beta \\ \dot{y} &= v\cos\alpha\cos\beta \end{aligned}$$

$$\begin{aligned} \dot{z} &= v \sin\beta \\ \dot{\alpha} &= u_1 \\ \dot{\beta} &= u_2 \\ \dot{v} &= u_3 \end{aligned} \tag{2.2}$$

Now the system has the form

$$\dot{X} = f(X) + u_1 g_1(X) + u_2 g_2(X) + u_3 g_3(X).$$

Note that because zero input no longer implies zero motion, the system is now said to have "drift", making it more difficult to control. Nevertheless, this formulation has the advantage that, in addition to ensuring that $v(t)$ is continuous, it allows v_0 and v_f to be specified with the initial and final conditions respectively. The input u_3 now corresponds to acceleration as opposed to velocity.

2.3 Characterizing Vehicle Performance

The performance limitations of the vehicle can be characterized by restricting the set of possible values for the inputs as follows:

$$\begin{aligned} |\, u_1 \,| &\leq v/R_t \\ |\, u_2 \,| &\leq v/R_p \\ |\, u_3 \,| &\leq a_{max} \end{aligned} \tag{2.3}$$

where $v \triangleq ||\overline{v}||$ and R_t, R_p, and a_{max} are constants.

Note that R_t corresponds to the minimum turning circle radius of the vehicle, and similarly, R_p corresponds to the minimum "pitching circle" radius. In choosing these constraints as such, an implicit assumption has been made that the minimum turning and pitching circle radii are independent of velocity. For real vehicles, this is not the case. However, it is a reasonable approximation. Furthermore, since the goal is to generate feasible paths, for any particular vehicle, R_t and R_p can always be chosen as the maximum values they take on as a function of velocity.

Chapter 3

Planning Motion

3.1 Decoupling the Problem into Two Subproblems

The task of developing a motion planner for a vehicle described by (2.2) and (2.3) can be greatly simplified by noting the following: the geometric shape of the path can be derived first, without regard to vehicle velocity. This property stems from the kinematic nature of the vehicle and the way in which its performance limits are specified — the vehicle's ability to turn and pitch as a function of distance traveled is not affected by its velocity.

This can be seen by reparameterizing the system in terms of pathlength. Since $ds = v\,dt$, (2.2) yields

$$\begin{aligned}
dx &= v \sin\alpha \cos\beta\, dt &&= \sin\alpha \cos\beta\, ds \\
dy &= v \cos\alpha \cos\beta\, dt &&= \cos\alpha \cos\beta\, ds \\
dz &= v \sin\beta\, dt &&= \sin\beta\, ds \\
d\alpha &= u_1\, dt &&= u_1/v\, ds \\
d\beta &= u_2\, dt &&= u_2/v\, ds \\
dv &= u_3\, dt &&= u_3/v\, ds
\end{aligned}$$

By choosing $u_1' = u_1/v$ and $u_2' = u_2/v$ the system can be written as

$$\begin{aligned} dx/ds &= \sin\alpha\cos\beta \\ dy/ds &= \cos\alpha\cos\beta \\ dz/ds &= \sin\beta \\ d\alpha/ds &= u_1' \\ d\beta/ds &= u_2' \end{aligned} \tag{3.1}$$

Note that because the dependence on v has been eliminated in the first five equations, the sixth can be dropped. The vehicle performance constraints given in (2.3) become

$$\begin{aligned} |u_1'| &\leq 1/R_t \\ |u_2'| &\leq 1/R_p \end{aligned} \tag{3.2}$$

The equations given in (3.1) together with the restrictions on the inputs given in (3.2) embody the nonholonomic and performance constraints on the motion of vehicle. Any path that can be generated by (3.1) using choices of u_1' and u_2' that meet the limits of (3.2) can be tracked by the vehicle. Note that (3.1) and (3.2) are both independent of velocity.

The problem of finding a trajectory can therefore be divided into two parts:

1. Find the shape of the path.
2. Choose $v(t)$ such that the vehicle arrives at each waypoint on time at with the appropriate velocity (when specified).

3.2 Finding the Shape of the Path

3.2.1 The Basic Problem

Suppose that the steering ability of an underwater vehicle is described by (3.1) and (3.2). In order to develop a path connecting a sequence of waypoints with associated constraints, one must first address the following basic problem:

GIVEN:

- initial state $X_0 = [x_0\ y_0\ z_0\ \alpha_0\ \beta_0]^T$
- final state $X_f = [x_f\ y_f\ z_f\ \alpha_f\ \beta_f]^T$

FIND:

a path $\{X(s) : s \in [0, L]\}$ such that:

- $X(0) = X_0$ and $X(L) = X_f$.
- $X(s)$ is feasible.

The term "feasible" is used here in a broad sense — a feasible trajectory is one that the vehicle is capable of reliably following. Thus the generated trajectory must take into account the nonholonomic and performance constraints for the vehicle and must have a positive velocity throughout.

Solutions to the problem as stated are not unique. Hence, the planner should find paths that are optimal in some sense. Propulsion is generally much more expensive than steering in terms of fuel usage. In addition, it is desirable to maximize the range of the vehicle during a given period of time. Thus the planner presented here optimizes on the basis of pathlength.

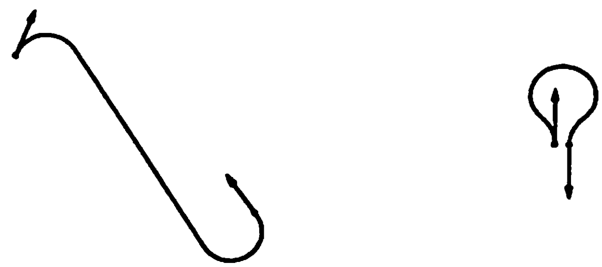
3.2.2 Curves of Minimal Length in Two Dimensions

At this point is useful to consider paths in the plane. Paths of minimal length in the plane are characterized by a theorem proved by L. E. Dubins in 1957 [1]:

Theorem 1 (Dubins) *A planar curve of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents is necessarily a continuously differentiable curve that is one of*

1. *an arc of a circle of radius R, followed by a line segment followed by an arc of a circle of radius R; or*

2. *a sequence of three arcs of circles of radius R; or*



RSL LRL

Figure 3.1: examples of paths of minimal length in two dimensions

3. a subpath of a path of type 1 or 2

where R is the inverse of the limit on average curvature.

The limitation on "average curvature" as opposed to just "curvature" is used to permit paths that have discontinuities in curvature, and hence have points at which curvature is undefined.

Theorem 1 implies that there are six types of paths of minimal length in two dimensions. They can be described as RSR, RSL, LSR, LSL, RLR, and LRL, where R corresponds to turning right at the maximum rate, L corresponds to turning left at the maximum rate, and S corresponds to going straight (see figure 3.1). Given two points and associated tangents, it is a matter of straight forward calculation to determine each of the six associated paths. Then to find the path of minimal length, one must only choose the shortest of the six.

Notice that this theorem applies directly to finding a feasible path for a simple car-like vehicle that never shifts into reverse. The kinematic equations of motion for such a vehicle can be derived in similar fashion to what was done for the submersible. Reparameterizing these equations in terms of pathlength gives

$$\begin{aligned} dx/ds &= \sin\alpha \\ dy/ds &= \cos\alpha \\ d\alpha/ds &= u \end{aligned} \tag{3.3}$$

and the associated limit on turning rate (curvature) becomes

$$|u| \leq 1/R \tag{3.4}$$

3.2.3 Extension to Curves in Three Dimensions

Unfortunately, no one has been able to extend Dubins' result to dimensions greater than two. And even if such a theorem did exist, it would not apply directly to the type of vehicle considered here since the vehicle has steering limits in the form of bounds on turning rate and pitching rate rather than a single bound on curvature. Theorem 1 has direct application to vehicles that operate in a planar environment. As it turns out, it is also useful for path planning in three dimensions.

As an initial goal, we would like to solve the basic problem given in section 3.2.1. As it turns out, this can be accomplished by applying theorem 1 twice. The resulting paths are generally not optimal in terms of minimizing pathlength, however they are near optimal for paths with small pitch angles. The first application of the theorem determines the x and y coordinates of the path. The second application assigns values for z.

Consider the projection of a three dimensional path onto the x-y plane. Let s' represent distance traveled along the projected path. Then s' is related to s by

$$ds' = ds \cos\beta \tag{3.5}$$

The steering equations of (3.1) can be reparameterized in terms of s' by substituting with equation (3.5). The first two equations in (3.1) then become

$$\begin{aligned} dx/ds' &= \sin\alpha \\ dy/ds' &= \cos\alpha \end{aligned} \tag{3.6}$$

and the associated limit on turning rate from (3.2) becomes

$$|d\alpha/ds'| \leq 1/R_t \cos\beta \tag{3.7}$$

Note that the β dependence has been removed in (3.6). These equations now resemble the kinematic equations for a car-like vehicle given by (3.3) and (3.4). In fact, by tightening the restriction given by (3.7) as follows:

$$|d\alpha/ds'| \leq 1/R_t \tag{3.8}$$

the behavior of (3.6) becomes identical to that of the car-like vehicle. Thus we can apply theorem 1 to the points (x_0, y_0) and (x_f, y_f) with tangents α_0 and α_f to generate the x and y components of our three dimensional path. This effectively generates the "shadow" (i.e. projection into the x-y plane) of the three dimensional path that we ultimately wish to derive. These choices for x and y are optimal for the (excessive) restriction of (3.8) but less than optimal for the true restriction on vehicle turning rate given by (3.7). Hence, optimality has been sacrificed in favor of a straight forward method for generating the paths. Note that as the pitch angle approaches zero, (3.7) and (3.8) become the same, and hence the resulting path becomes optimal.

The next step is to somehow expand our (planar) curve into three dimensions. Consider the surface defined by taking the projected path and expanding it in the $\pm z$ directions. This surface is homeomorphic to a subset of the plane, and can be parameterized by s' and z as shown in figure 3.2. Suppose the length of the planar path is given by L'. We must find a path on this new surface from $s' = 0$ and $z = z_0$ with tangent specified by β_0 to $s' = L'$ and $z = z_f$ with tangent specified by β_f. From (3.5), (3.1), and (3.2) respectively,

$$\begin{aligned} ds'/ds &= \cos\beta \\ dz/ds &= \sin\beta \end{aligned} \tag{3.9}$$

$$|d\beta/ds| \leq 1/R_p \tag{3.10}$$

With the exception of an inconsequential coordinate change, the system defined by (3.9) and (3.10) is identical to the car-like vehicle of (3.3) and (3.4). Once again, theorem 1 can be applied to find the shortest path on this surface that meets the constraints. This time, however, there is no further loss of optimality. A set of pairs (s', z) is determined. The path found from the first application of the theorem

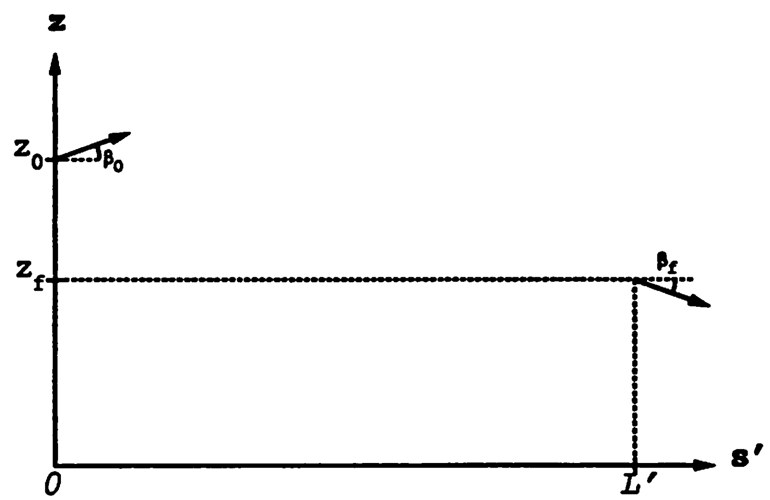


Figure 3.2: parameterization of surface defined by expanding planar curve into the $\pm z$ directions

can be used to map each s' to a pair (x, y). The resulting set of triples (x, y, z) completely determines the path.

It should be noted at this point that in some (rare) cases, shortest paths on the s'-z surface of the RLR and LRL types will enter regions in which $s' < 0$ or $s' > L'$. In these regions, there are no previously generated (x, y) pairs to which to assign a z value. This problem is easily solved by extending the x-y plane path at its endpoints (in a straight line, for example) to obtain the necessary (x, y) pairs.

The algorithm can be summarized as follows:

step 1: Apply theorem 1 to points (x_0, y_0) and (x_f, y_f) with tangents given by α_0 and α_f to generate the x and y components of the path. Let L' be the length of this path.

step 2: Apply theorem 1 to points $(0, z_0)$ and (L', z_f) with tangents given by β_0 and β_f to generate a set of pairs (s', z).

step 3: For each pair (s', z), use the result of step 1 to map s' to a pair (x, y). The resulting triples (x, y, z) fully determine the path.

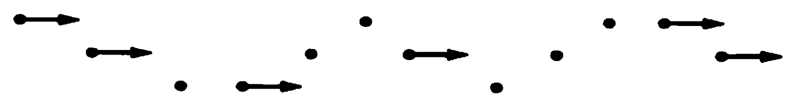


Figure 3.3: path planning with waypoints

3.2.4 Extension to Paths with Waypoints

The algorithm presented thus far generates a feasible three dimensional path between two points with specified tangents. In addition, it may be desirable to specify intermediate "waypoints" through which the vehicle must pass as well (see fig 3.3). In some cases, one may wish to specify a tangent direction associated with a point. In other cases, the direction may be unimportant in which case the motion planner should choose the direction so as to minimize the length of the overall path.

The more general problem can be stated as follows:

GIVEN:

- a sequence of points in $\mathbb{R}^3$: $\{p_i\}, \;\; i = 1..n$.
- a set of headings associated with a subset of the points:
  $\{\alpha_j\}, \;\; j \in J \subset \{1..n\}$
- a set of pitch angles associated with a subset of the points:
  $\{\beta_k\}, \;\; k \in K \subset \{1..n\}$

FIND:

a path $\{X(s) : s \in [0, L]\}$ such that:

- $X(s)$ passes through all the points and with the required headings and pitch angles when given.
- $X(s)$ is feasible.

Consider first paths of minimal length in the plane.

Proposition ***Suppose a planar path with a limit on average curvature passes through a sequence of waypoints and is of minimal length. Then each subpath connecting consecutive waypoints must satisfy the criteria of theorem 1.***

Proof. By contradiction. Suppose path P is of minimal length and has subpath P_1 connecting two consecutive waypoints. Suppose P_1 does not meet the criteria of theorem 1. Then, by theorem 1, there exists a subpath P_1' with the same initial and terminal positions and tangents as P_1 that is of minimal length. Since, by theorem 1, P_1 is not of minimal length, P_1' must be shorter than P_1. Thus there exists path P' formed by substituting P_1' for P_1 in P that passes through the same waypoints with the same tangents as P and is shorter than P. Hence P is not of minimal length. □

Since each subpath meets the criteria of theorem 1, each subpath is a function of its endpoint positions and tangents only. The endpoints are always given. Therefore the only variables in choosing an overall path are the tangents directions for those waypoints whose tangents are not already specified. Thus, for the two dimensional case, the problem is one of optimization in which one must choose the the unspecified tangent directions so as to minimize the overall length of the path.

The most straightforward way to accomplish this optimization is to discretize the space and choose the path of shortest length. Note that every sequence of waypoints can be divided into subsequences such that each subsequence has endpoints with specified tangents and interior waypoints with unspecified tangents. Each such subsequence presents an independent problem. The complexity of the calculation depends on the length of the largest such subsequence.

Supposed the unspecified tangent angles in such a subsequence are given by $\{\theta_i\}$, $i = 1..k$, and that each θ_i is discretized into N distinct values. Then a naive algorithm that discretizes the waypoint tangent angles in sequential order to find the shortest path has complexity $O(N^k)$. By always choosing to discretize a central waypoint first, the complexity can be reduced to $O(N^{\log k})$. Furthermore, by precomputing a matrix of subpath lengths for every pair of adjacent points and then sequentially reducing them by vector-matrix "multiplication" to find the shortest subpath as a function of endpoint tangent angles, the complexity is further reduced to $O(kN^2)$.

Once the machinery is in place for generating optimal paths with waypoints in the plane, three dimensional paths that are near optimal can be generated in exactly the same way as was done for the simple paths in section 3.2.3. The x and y components are generated first. This curve is then used to define a surface by extending it into the $\pm z$ directions. A second iteration then adds z coordinates to the already determined path in x and y.

3.3 Finding the Velocity of the Vehicle

Once the shape of the path has been determined, and hence its length is known, the problem reduces to finding a satisfactory $v(t)$ to meet the given constraints on arrival time and velocity.

Consider a subpath connecting two consecutive waypoints. In order to determine an appropriate velocity profile for that subpath, the following problem must be solved:

GIVEN:

- length of the subpath, L
- desired transit time, T
- initial and final velocities, v_0 and v_f

FIND:

velocity $\{v(t) : t \in [0, T]\}$ such that:

- $v(0) = v_0$ and $v(T) = v_f$.
- $v(t)$ is continuous and positive $\forall t \in [0, T]$
- $\int_0^T v(t)dt = L$

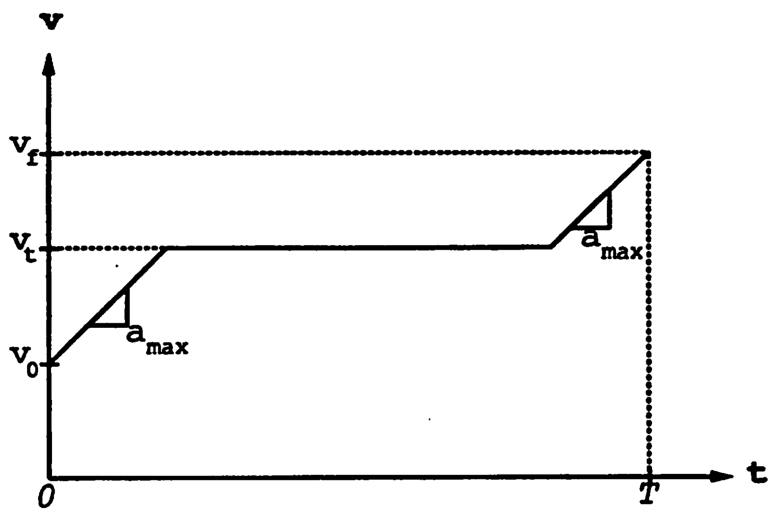


Figure 3.4: a "trapezoidal" velocity profile

In general, there are many choices of v(t) that will solve the above problem, all of which are optimal using our criterion since the pathlength has already been determined. We choose our velocity profiles to be "trapezoidal" as shown in figure (3.4). A steady transit velocity, v_t, is maintained during the middle portion of the trajectory. The vehicle accelerates or decelerates at maximum rate near the endpoints in order to meet the specified initial and final velocities. Note that because v_0, v_f, a_{max}, and T are all fixed, choosing v_t completely determines $v(t)$. Furthermore, there is a monotonic one-to-one relationship between v_t and L.

The pathlength L can be expressed as a function of v_t as follows:

$$\begin{aligned} L &= \int_0^T v(t)dt \\ &= v_t T + \frac{1}{2a_{max}}\Big[|v_t - v_0|(v_0 - v_t) + |v_t - v_f|(v_f - v_t)\Big] \end{aligned} \tag{3.11}$$

We wish to invert equation (3.11) so that v_t can be expressed as a function of L. In order to do so, the following values must be precomputed:

$$\begin{aligned} L_{min} &\triangleq L\,\big|_{v_t = v_{min}} \\ L_{max} &\triangleq L\,\big|_{v_t = v_{max}} \\ L_{v_0} &\triangleq L\,\big|_{v_t = v_0} \end{aligned}$$

$$L_{v_f} \triangleq L\,|_{v_t=v_f}$$

where v_{min} and v_{max} are determined either by the limitations of the vehicle or by the cases in which the vehicle must undergo continuous acceleration followed immediately by continuous deceleration or vice versa. Now v_t, and therefore $v(t)$, can be determined as follows:

case 1: $L < L_{min}$ or $L > L_{max}$

The trajectory can not be achieved due to the performance limitations of the vehicle.

case 2: $L_{min} < L < min\{L_{v_0}, L_{v_f}\}$

$$v_t = \frac{v_0 + v_f}{2} - \frac{Ta_{max}}{2} + \frac{1}{2}\sqrt{T^2a_{max}^2 - 2Ta_{max}(v_0 + v_f) - (v_0 - v_f)^2 + 4La_{max}}$$

case 3: $min\{L_{v_0}, L_{v_f}\} < L < max\{L_{v_0}, L_{v_f}\}$

$$v_t = \frac{1}{2}\left(\frac{2La_{max} - |v_0^2 - v_f^2|}{Ta_{max} - |v_0 - v_f|}\right)$$

case 4: $max\{L_{v_0}, L_{v_f}\} < L < L_{max}$

$$v_t = \frac{v_0 + v_f}{2} + \frac{Ta_{max}}{2} - \frac{1}{2}\sqrt{T^2a_{max}^2 + 2Ta_{max}(v_0 + v_f) - (v_0 - v_f)^2 - 4La_{max}}$$

There are two primary advantages to the "trapezoidal" form of the velocity profiles presented here. First of all, it maximizes the duration of the trajectory in which the velocity is steady. In fact, for long distance transits, the path shape found using the method presented in 3.2 will have a large segment in which the the heading and pitch angle are constant as well. Thus a large portion of the transit will be at a steady course and speed, thereby making the control problem much easier. The second advantage of velocity profiles of this form is that it allows the motion planner to take full advantage of the vehicle's capabilities. For example, in order to maximize the distance traveled in a given amount of time, the vehicle should accelerate at maximum rate to its maximum velocity and then decelerate if necessary at the end of the trajectory to reach a specified final velocity.

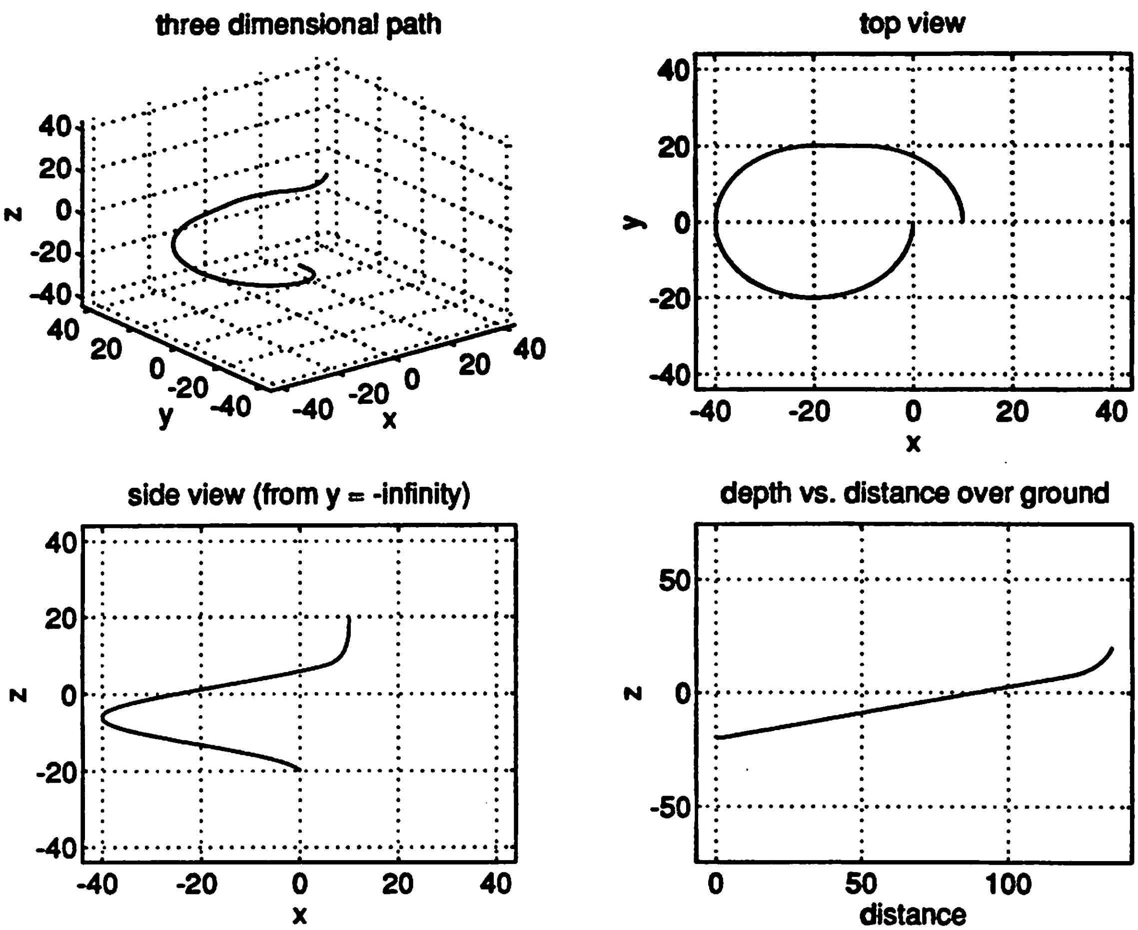
So far, this discussion has centered around the problem of determining the velocity profile between a pair of consecutive waypoints with times and velocities specified for each. The extension of this method to a series of points is straight forward. In general, a velocity profile can be found for each pair of adjacent points and the results can be concatenated to generate a velocity profile for the whole path. A point specified without regard to time of arrival or velocity can be dealt with by simply ignoring that point and deriving the velocity profile for a larger portion of the path. For cases in which the velocity is not specified at one or both of the endpoints, a profile can be found by substituting v_t for v_0 and/or v_f in equation (3.11) as appropriate and then inverting. For cases in which the time is not specified for a given waypoint, a time should be chosen based on the times specified for the preceding and following points and the relative distances to each.

Chapter 4

Examples of Paths

The figures on the following pages show examples of paths generated using the method presented in this paper. The paths were generated using MATLAB and required approximately 250 lines of code.

In each case, the algorithm was given fully specified initial and final configurations as shown. The limits on the vehicle's turning and pitching rates were set at $R_t = 20$ and $R_p = 20$ respectively.



INITIAL CONFIGURATION

$x_0 = 0$

$y_0 = 0$

$z_0 = -20$

$\alpha_0 = 180$

$\beta_0 = 0$

FINAL CONFIGURATION

$x_f = 10$

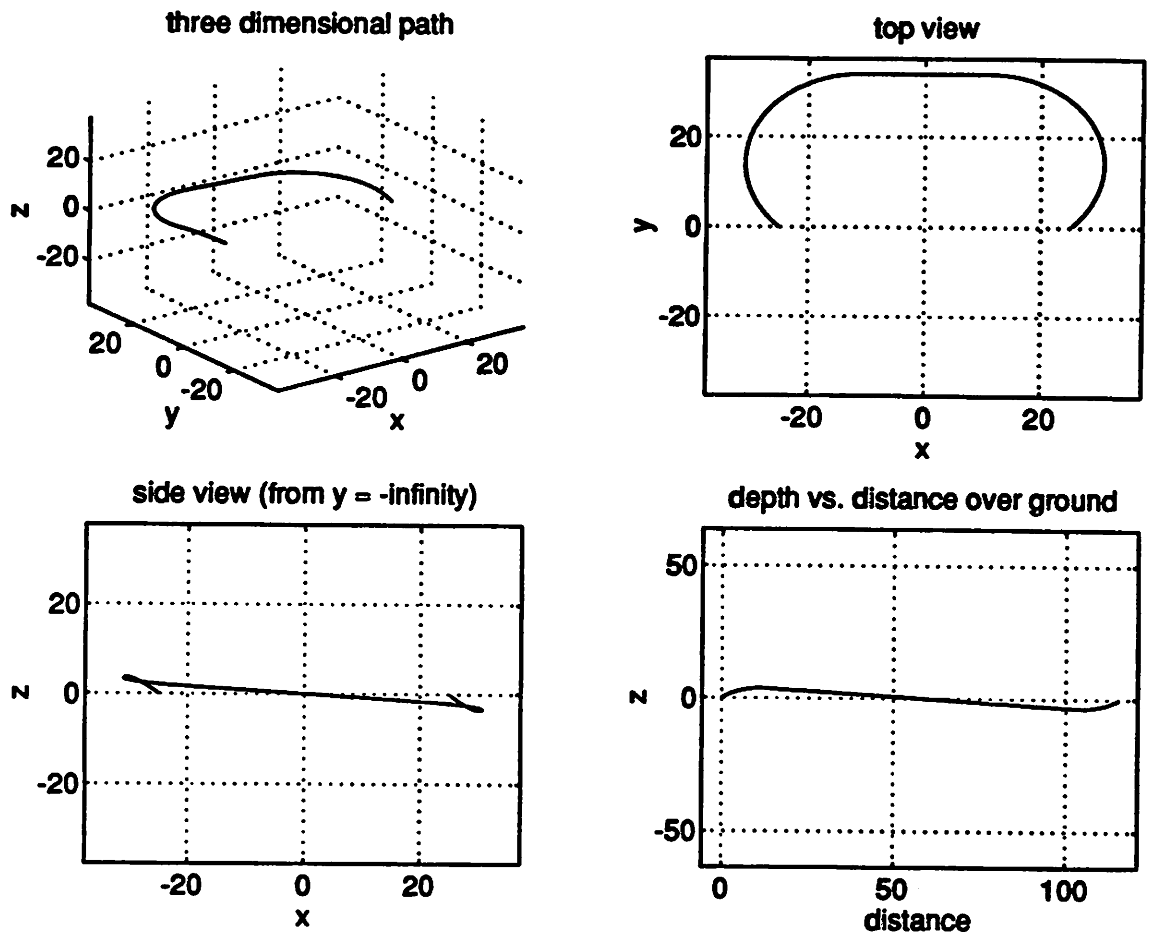
$y_f = 0$

$z_f = 20$

$\alpha_f = 180$

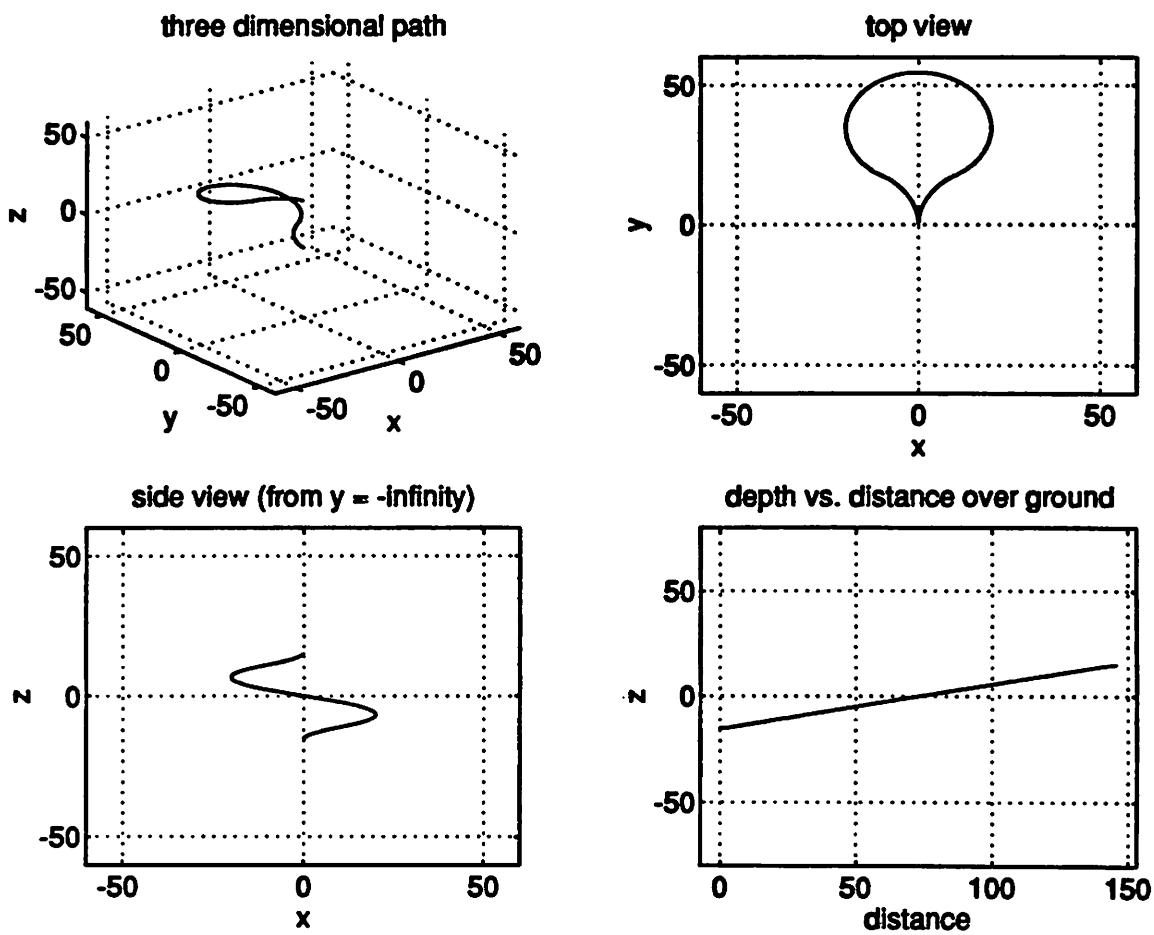
$\beta_f = 70$

Figure 4.1: path planning example



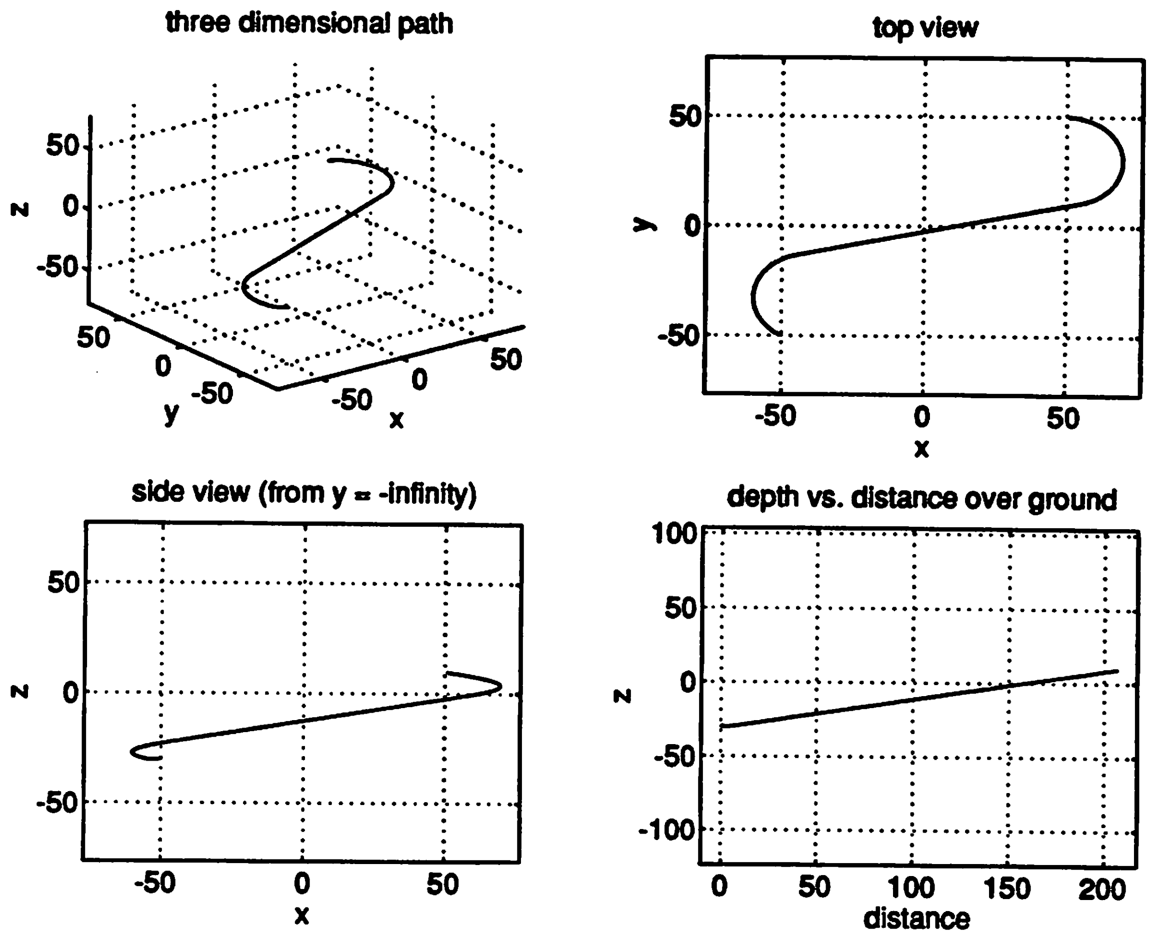
INITIAL CONFIGURATION	FINAL CONFIGURATION
$x_0 = -25$	$x_f = 25$
$y_0 = 0$	$y_f = 0$
$z_0 = 0$	$z_f = 0$
$\alpha_0 = 315$	$\alpha_f = 225$
$\beta_0 = 35$	$\beta_f = 35$

Figure 4.2: path planning example



INITIAL CONFIGURATION	FINAL CONFIGURATION
$x_0 = 0$	$x_f = 0$
$y_0 = 0$	$y_f = 0$
$z_0 = -15$	$z_f = 15$
$\alpha_0 = 0$	$\alpha_f = 180$
$\beta_0 = 0$	$\beta_f = 0$

Figure 4.3: path planning example



INITIAL CONFIGURATION

$x_0 = -50$

$y_0 = -50$

$z_0 = -30$

$\alpha_0 = 300$

$\beta_0 = -10$

FINAL CONFIGURATION

$x_f = 50$

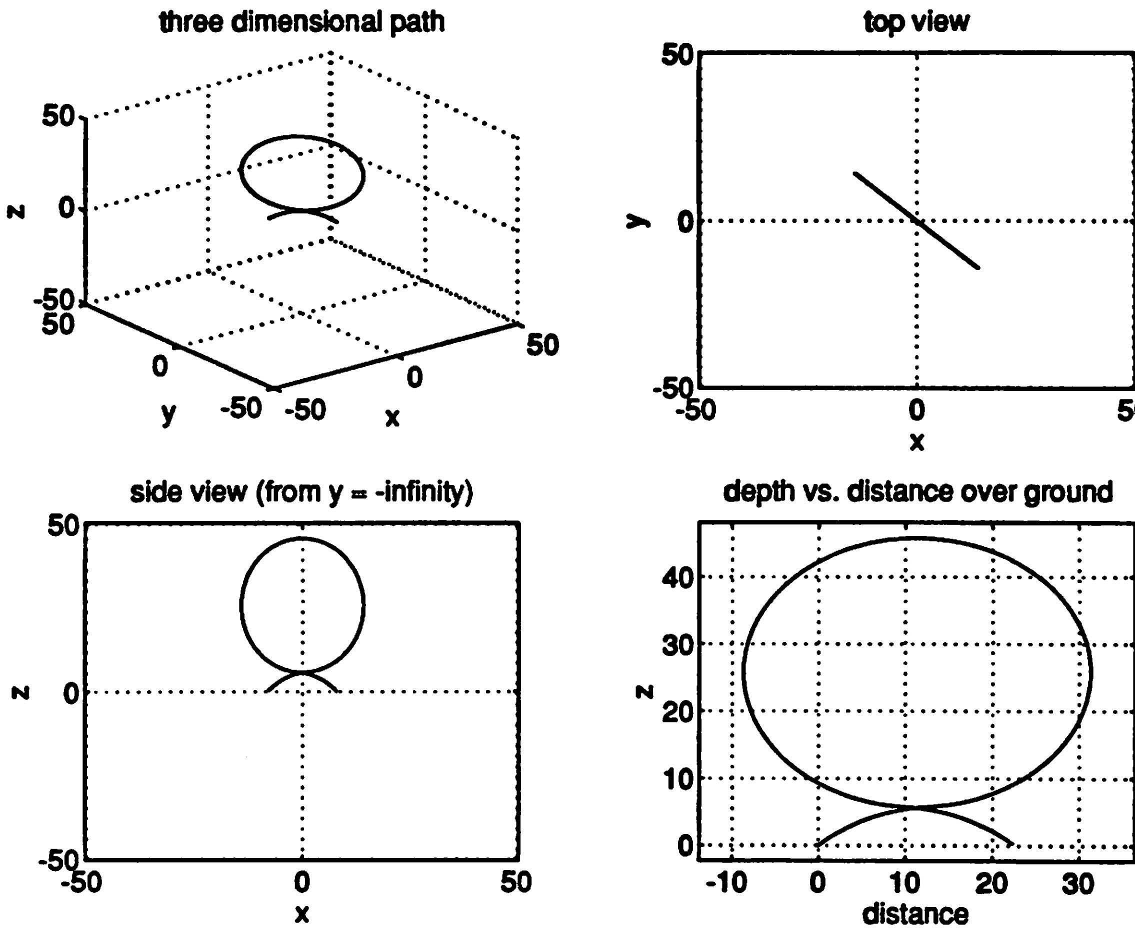
$y_f = 50$

$z_f = 10$

$\alpha_f = 270$

$\beta_f = 10$

Figure 4.4: path planning example



INITIAL CONFIGURATION

$x_0 = 8$

$y_0 = -8$

$z_0 = 0$

$\alpha_0 = 315$

$\beta_0 = 45$

FINAL CONFIGURATION

$x_f = -8$

$y_f = 8$

$z_f = 0$

$\alpha_f = 315$

$\beta_f = -45$

Figure 4.5: path planning example

Chapter 5

Conclusion

A general algorithm has been presented for planning trajectories for underwater vehicles. The trajectories satisfy the nonholonomic and performance constraints on the motion of the vehicle. Given a set of waypoints and associated specifications for time of arrival, heading, pitch angle, and/or velocity, the algorithm always returns a feasible path. The trajectory can then be realized as long as the specified times and velocities do not require the vehicle to exceed its limits on acceleration and/or velocity.

Given any two points in the vehicle's configuration space, the algorithm generates a path between them that is near optimal in the sense of minimizing the pathlength. The problem of characterizing the optimal (shortest) path itself remains open.

The paths are generated by expanding an optimal planar path into three dimensions. Generated paths with small pitch angles are approximately planar and hence near optimal. As it turns out, underwater vehicles typically operate at small pitch angles most, if not all, of the time. Many vehicles are limited by operating restrictions on the equipment on board. Manned vehicles are also limited by passenger comfort and safety. Furthermore, regardless of the limitations on the vehicle, for many applications, large pitch angles are not needed — the ocean environment can often be regarded as two and a half dimensional due to the limit on the vehicle's operating depth. Hence, distances to be covered in x and y typically far exceed those to be covered in z.

Note that, in order to use this algorithm, the performance limitations of the vehicle must be specified relative to a fixed inertial frame. For a vehicle that experiences a significant amount of roll, it would be more desirable to specify these limits relative to a body frame. This would require a reformulation of the kinematic model presented here such that the configuration space includes the roll angle as well and hence becomes $SE(3)$. Doing so makes the problem significantly more difficult.

In addition to its obvious applications for motion planners for AUVs and autopilots for manned submarines, the algorithm can also be implemented for fixed-wing aircraft.

Bibliography

[1] L. E. Dubins. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[2] T. Fujii and T. Ura. Development of Motion Control System for AUV Using Neural Nets. In *IEEE Symposioum on Autonomous Underwater Vehicle Technology*, pages 81–86, 1990.

[3] A. J. Healey and David Lienard. Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, 18(3):327–339, 1993.

[4] P. Jacobs and J. Canny. Robust motion planning for mobile robots. In *IEEE International Conference on Robotics and Automation*. IEEE, 1990.

[5] P. Jacobs, J-P. Laumond, and M. Taix. A Complete Iterative Motion Planner for a Car-like Robot. In *Journees Geometrie Algorithmique*, INRIA, 1990.

[6] Y. Kanayama and B. I. Hartman. Smooth Local Path Planning for Autonomous Vehicles. Technical Report TRCS88-15, University of California, Santa Barbara, June 1988.

[7] G. Lafferriere and H. J. Sussman. Motion Planning for Controllable Systems Without Drift: A Preliminary Report. Technical Report SYSCON-90-04, Rutgers Center for Systems and Control, June 1990.

[8] J-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[9] Z. Li and J. F. Canny, editors. *Nonholonomic Motion Planning.* Kluwer Academic Press, University of California, Berkeley, 1992.

[10] B. Mirtich and J. F. Canny. Using Skeletons for Nonholonomic Path Planning Among Obstacles. In *IEEE International Conference on Robotics and Automation*, pages 2533–2540, 1992.

[11] R. M. Murray and S. S. Sastry. Steering Nonholonomic Systems Using Sinusoids. In *IEEE Control and Decision Conference*, 1990.

[12] Y. Nakamura and S. Savant. Nonlinear Tracking Control of Autonomous Underwater Vehicles. In *IEEE International Conference on Robotics and Automation*, volume 3, pages A4–A9, 1992.

[13] Z. Li R. M. Murray and S. S. Sastry. A Mathematical Introduction to Robotic Manipulation. Preprint, 1993.

[14] J. A. Reeds and L. A. Shepp. Optimal Paths for a Car That Goes Both Forwards and Backwards. *Pacific Journal of Mathematics*, 145(2), 1990.