6-1-2011

Minimum time kinematic trajectories for self-propelled rigid bodies in the unobstructed plane

Andrei A. Furtuna
Dartmouth College

MINIMUM TIME KINEMATIC TRAJECTORIES FOR SELF-PROPELLED RIGID BODIES IN THE UNOBSTRUCTED PLANE

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Andrei Furtuna

DARTMOUTH COLLEGE

Hanover, New Hampshire

June 2011

Examining Committee:

(chair) Devin Balkcom

Chris Bailey-Kellogg

Peter Winkler

Yuliy Baryshnikov

Brian W. Pogue, Ph.D.
Dean of Graduate Studies

Abstract

The problem of moving rigid bodies efficiently is of particular interest in robotics because the simplest model of a mobile robot or of a manipulated object is often a rigid body. Path planning, controller design and robot design may all benefit from precise knowledge of optimal trajectories for a set of permitted controls.

In this work, we present a general solution to the problem of finding minimum time trajectories for an arbitrary self-propelled, velocity-bounded rigid body in the obstacle-free plane. Such minimum-time trajectories depend on the vehicle's capabilities and on and the start and goal configurations. For example, the fastest way to move a car sideways might be to execute a parallel-parking motion. The fastest long-distance trajectories for a wheelchair-like vehicle might be of a turn-drive-turn variety.

Our analysis reveals a wide variety of types of optimal trajectories. We determine an exhaustive taxonomy of optimal trajectory types, presented as a branching tree. For each of the necessary leaf nodes, we develop a specific algorithm to find the fastest trajectory in that node. The fastest trajectory overall is drawn from this set.

Contents

List of Figures

Chapter 1

Introduction

The problem of moving rigid bodies efficiently is of particular interest in robotics because the simplest model of a mobile robot or of a manipulated object is often a rigid body. Path planning, controller design and robot design may all benefit from precise knowledge of optimal trajectories for a set of permitted controls.

In this work, we present a general solution to the problem of finding minimum time trajectories for a self-propelled, velocity-bounded rigid body in the obstacle-free plane. Such minimum-time trajectories depend on the vehicle's capabilities and on and the start and goal configurations. For example, the fastest way to move a car sideways might be to execute a parallel-parking motion. The fastest long-distance trajectories for a wheelchair-like vehicle might be of a turn-drive-turn variety.

Our mathematical model is fully developed in Chapter 3. We assume the controls that can be applied to a rigid body to be velocities with $\dot{x}$, $\dot{y}$ and $\dot{\theta}$ components. The accelerations are assumed to be instantaneous, allowing direct control over the velocity. The problem is to find minimum time control policies that cause the vehicle to reach the goal.

A control policy is a function $u(t)$ that specifies which velocity is chosen at each time t. The velocities are chosen from within a convex polyhedral set of velocities U. Reflecting the assumption that the vehicle moves autonomously, the control set is fixed in the body's frame of reference. Among all the control policies that take us from an initial configuration $q_0 = (x_0, y_0, \theta_0)$ to a goal configuration $q_g = (x_g, y_g, \theta_g)$, we wish to find one of the least duration possible. The minimum-time curves ("brachistochrones") characterized in the present work are a generalization of some previously studied minimum-length curves ("geodesics"): the Dubins [12] and the Reeds-Shepp [19] bounded-curvature trajectories.

An example minimum time curve found by our algorithm, for an omnidirectional vehicle with three

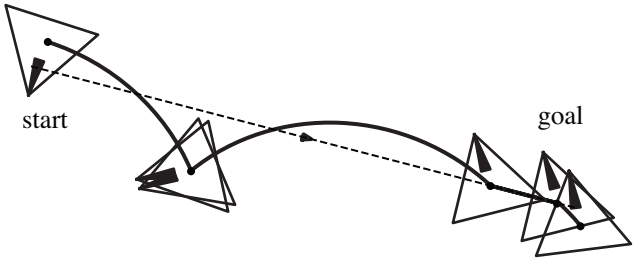


Figure 1.1: The fastest trajectory connecting the two configurations shown for a vehicle with three powered omni-wheels. This trajectory is *singular*, as it contains a translation parallel to the *control line*.

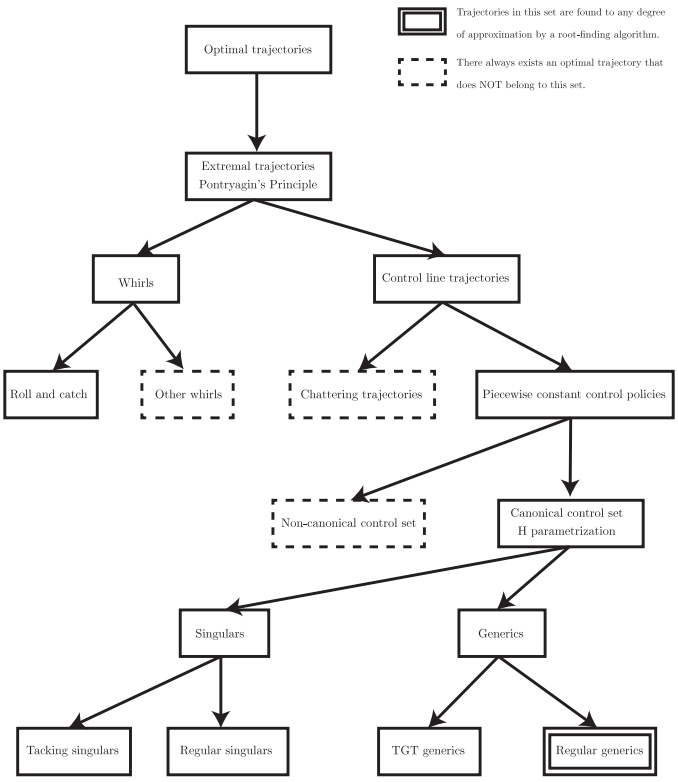


Figure 1.2: Taxonomy of time optimal kinematic trajectories for self-propelled rigid bodies in the plane, represented as a tree. We give a search algorithm for each of the leaf nodes, with the exception of the types within dashed border boxes, which are not necessary for optimality.

symmetrically placed omniwheels, is shown in Figure 1.1.

1.1 Summary of results

Our analysis reveals a wide variety of types of optimal trajectories. One of the main results of the present work is an exhaustive taxonomy of optimal trajectory types. Figure 1.2 shows this taxonomy as a tree. Some of the leaf nodes of this tree are such that there always exists an optimal trajectory outside the corresponding class. For each of the remaining, necessary leaf nodes, we develop a specific algorithm to find the fastest trajectory in that node. The fastest trajectory overall is drawn from this set.

Chapter 3 defines the model and the problem and gives formulas for the forward kinematics of rigid bodies in the plane. A simple planner is developed as a constructive proof of controllability. The existence of optimal trajectories then follows from Fillipov's existence theorem.

Chapter 4 applies the Pontryagin Principle, which sets necessary conditions for optimality, to the problem. Optimal trajectories that verify the Pontryagin Principle are shown to fall into two classes: the more general class for which there exists a directing line in the plane, called a *control line*, and the more restricted class of *whirls*. Whirls are trajectories that maintain a constant angular velocity. By applying a different version of the Pontryagin Principle, as well as some geometrical analysis, we find a canonical subclass of

whirls, roll-and-catch trajectories, that is sufficient for optimality. We conclude the chapter by giving an algorithm for finding the fastest roll-and-catch trajectory.

Chapter 5 studies optimal control policies for the more general class of control line trajectories. The main result of this chapter is that piecewise constant control policies that only use controls from a canonical, finite subset of the polyhedral control set are sufficient for optimality.

Chapter 6 prepares the stage for an efficient algorithmic search of the space of canonical control policies. We first show that, except on well-characterized singular intervals, the position of the control line uniquely determines the optimal control policy. We then give an algorithm that generates the optimal control policy on non-singular intervals, when given the position of the control line.

Chapter 7 makes extensive use of the generator developed in Chapter 6 in order to develop specific algorithms for each of several further subclasses of canonical control line trajectories. *Singulars* are trajectories that may contain singular intervals. Singular trajectories may be either regular, in which singular intervals effect translations parallel to the control line, or *tacking*, which may contain a translation-translation control switch. The algorithm for finding the fastest tacking trajectory is only slightly modified from the algorithm for finding the fastest regular singular.

Generics are trajectories where the control policy is everywhere uniquely determined by the position of the control line. *TGT* generics begin and end with translations, and in this case it is possible to find the exact position of the control line. For all other generics, we give a parametrization of the control line, and a root finding approximation algorithm.

Finally, Chapter 8 presents some experimental results obtained from implementing the algorithms for the more general cases. We have implemented algorithms for the more common subtypes (except whirls and tacking) and obtained trajectories, which we believe to be optimal, for both previously studied vehicles, as well as a vehicle for which the boundary problem had not been solved before (the symmetric three-omniwheeled robot). We also discuss possible future directions of development and applications.

Chapter 2

Related work

The earliest origins of optimal control theory can be said to be almost as old as the calculus of variations: with the benefit of hindsight, we can place Bernoulli's 1696 brachistochrone problem within the realm of optimal control theory ([27]). A firm theoretical basis for this discipline was, however, only established in the 1950s and 1960s, initially through the work of Richard Bellman at RAND. Bellman formulated the fundamental problem in terms of searching for a "control" function, i.e. a function of time, while given a system of differential constraints and an objective function to maximize. Initially analyzing the discrete case (i.e. multistage processes), Bellman's first result was the well-known method of dynamic programming. His later extension of the fundamental equation to the continuous case generated the Hamilton-Jacobi-Bellman equation, a sufficient condition for optimality and the basis of so-called "direct" methods in modern control theory (for a concise summary of Bellman's results, see [4]). The HJB equation is generally difficult to solve analytically except in relatively simple instances of the control problem, although at least one of these cases (linear quadratic control) has important practical applications. The HJB equation has, however, proven very suitable as a basis for numeric methods for optimal control, e.g. the direct collocation methods implemented in multiple FORTRAN and MATLAB software packages.

At roughly the same time as Bellman's work, and on the other side of the Iron Curtain, the second foundation stone of modern optimal control theory was being laid at the Moscow Steklov Mathematical Institute, by a team under the leadership of Lev Pontryagin. The main result of their "indirect" method ([18]) has become known as the Pontryagin Principle (called the Maximum Principle by Pontryagin himself, and the Pontryagin Minimum Principle in some recent works). The Principle is a strong necessary condition on the local structure of optimal control functions (and their corresponding trajectories in state space) and has provided an easier path to analytically solving non-linear optimal control systems than the HJB equation. Numerical optimizers based on the PMP also exist (e.g. BNDSCO [16]).

While the applications of dynamic programming, the HJB equation, and the Pontryagin Principle are widespread and diverse, we will restrict the current work to geometric considerations inspired by the movement of planar robots. Planar curves (whether considered as vehicle trajectories, or not) have also been subject to mathematical interest for a very long time. Without dwelling too long on subjects such as Galileo's characterization of the cycloid, we will trace the roots of the current inquiry to a problem posed by Markov in 1887 ([15]): what are the shortest planar curves, of bounded maximum curvature and connecting two given tangent vectors? Seventy years later, the solution to this problem was characterized by L. E. Dubins ([12]), who described a class of curves he called *R-geodesics*. In 1975, Cockayne and Hall ([9]) showed how to synthesize the shortest R-geodesics to any given configuration.

In 1990, Reeds and Shepp ([19]), explicitly motivated by a robotics problem (finding the optimal trajec-

tories for a robotic cart), and by Dubins' success at giving, in effect, the shortest paths for a car that can only travel forwards, gave a characterization of the geodesics for a car that can travel both forwards and backwards. Their result revealed the scope for the application of optimal control theory into mobile robotics, and shortly afterwards two independent papers (by Sussmann and Tang [28] in 1991 and Boissonnat *et al.* [5] in 1992, respectively) re-established (and even slightly tightened) the Reeds and Shepp results on the more general basis of classical optimal control theory. The optimal trajectory synthesis, i.e. the set of optimal trajectories from all starting configurations, for the Reeds-Shepp car was given by Souères and Laumond in 1996 [25]. The Pontryagin Principle constituted the main theoretical basis of these papers, and at this point we see coming together the approach of which the present work is the logical continuation: analyzing planar motion problems through the prism of the Pontryagin Principle.

The chassis of mobile robots is, of course, not limited to the cart design. The most popular arrangement, at least for small robots, seems to be the differential drive: two independent motors driving parallel wheels. Such vehicles, unlike cars, can spin in place, thus rendering a straight application of the "geodesic" criterion into a rather uninteresting problem - the shortest paths for the center of the robot are always to spin in place, and drive straight to the destination. There are, however, two optimality criteria for which the problem becomes nontrivial. A "fuel consumption" criterion (the sum of the distances traveled by each wheel) was studied by Chitsaz *et al.* [7], and surprisingly enough the optimal paths, in this case, turn out to be identical to the Reeds-Shepp paths. Also, a "brachistochronic" (time optimal) criterion was used in several analyses.

While time optimality is the simplest case for the Pontryagin Principle (as one of the terms in the main PMP equation becomes extremely simplified), such a criterion raises, on the other hand, the problem of dynamic effects. In classical mechanics, the "controls" are forces, which result in accelerations; thus, one controls the second derivative of the vehicle's position. A model of the differential drive in which the accelerations of the two wheels are the controls (within the $[-1, 1]$ interval) has been extensively studied. This model, associated with the LAAS-CNRS robot Hilare, was proposed by Jacobs *et al.* in 1991 [17]. In 1994, Reister and Pin [20] presented a numerical analysis of trajectories for the Hilare model that contained at most four switches. In 1997, Renaud and Fourquet [21] showed that some optimal trajectories for this vehicle contain more than four control switches; thus only the shapes of the optimal segments seem to be known for this system (they are pieces of clothoids and involutes of circles [24]).

An alternate time-optimal model of the differential drive was studied, with more extensive results, by Balkcom and Mason in 2002 [3]. They obtained a novel set of curves, as well as an optimal trajectory synthesis for the differential drive, by considering the controls to be wheel velocities rather than accelerations. There is a good case to be made that such a "kinematic" (as opposed to dynamic) model is not only convenient to analyze, but also fairly accurate. Common electric motors respond to a given voltage by quickly settling to a well-determined velocity. The input can thus be considered to be, in effect, this velocity; and the time to cover a given distance d under a given control is better estimated through dividing the distance by the "steady" velocity corresponding to the control, rather than through the $\frac{1}{2}ma^2$ formula. For small robots, the acceleration times are expected to be quite short. Furthermore, the optimal control problem for dynamic models appears to be very difficult, as the differential equations describing the trajectories do not have recognizable analytic solutions, and, in some cases, the optimal trajectories appear to involve chattering ("the Fuller phenomenon"), an infinite number of control switches within a finite time. Such issues occured in analyses of the bounded acceleration Dubins car (studied by Sussmann in 1997 [26] and by Souères and Boissonnat in 1998 [24]) and of a dynamic model of an underwater vehicle (studied by Chyba and Haberkorn in 2005 [8]).

On the other hand, the combination of kinematic models and the Pontryagin Principle seems more amenable to analytic solution. Such a model was used by Balkcom *et al.* in 2006 to characterize the time

optimal paths for a symmetric omnidirectional vehicle [2] and by Furtuna and Balkcom in 2008 [14] to characterize the optimal paths for a vehicle with three arbitrarily placed omniwheels (which was also a generalization of the previous results by Dubins and Reeds and Shepp). Furtuna and Balkcom extended the analysis of the kinematic model to cover the time optimal trajectories for any rigid body in the plane with velocity controls in [13]. These results will be discussed at length in a subsequent section. But first, we will introduce a few other works that are related to the current work.

Analyzing the optimal trajectories in the absence of obstacles has limited direct applications, as few mobile robots will operate within boundless, empty parking lots. But the structures thus derived do have a significant place as part of the general problem of moving efficiently within a cluttered environment. In fact, all optimal trajectory segments that do not exactly follow the contour of an obstacle should have their shape characterized by the obstacle-free model, by the principle that a subsegment of an optimal trajectory must itself be an optimal trajectory. Thus, a solution to the problem of optimal motion in the presence of obstacles consists of not only the mentioned work, but also two additional components: a characterization of which shapes of obstacle boundaries can be followed, and a theory of the junction points between free and obstacle-bound segments. Such a comprehensive solution has not been, so far, completed for any of the studied vehicles.

The closest work to our intended approach is Chitsaz's Ph. D. thesis [6]. By considering the differential drive with a "total wheel motion" metric among obstacles, Chitsaz developed a generalization of the visibility graph that he called a *nonholonomic bitangency graph*. This graph (which has to be generated numerically, based on the shape of the obstacles) encodes the structure of the optimal trajectories among obstacles; it is then to be processed by a standard shortest-path graph algorithm.

The other vehicle model that has been extensively studied among obstacles is the Reeds-Shepp car (with the standard distance optimization metric). In 1996, Desaulniers [11] showed that the optimal trajectories in such a case do not exist for some instances of the problem (or, more precisely, that such optimal trajectories involve infinite chattering). For cases where the optimal trajectories do exist, and the obstacles only consist of a room's walls (i.e., the work space is within a convex polygon), Agarwal *et al* gave a $O(n^2 \log n)$ algorithm for finding such trajectories [1].

Optimal distance metrics may also give some useful information about how obstacles may interfere with desired motions. Vendittelli *et al.* [29] developed an algorithm to obtain the shortest non-holonomic distance from a robot to any point on an obstacle. Optimal paths between pairs of points in configuration space may not exist in the presence of visibility constraints. Salaris *et al.* [22] give the optimal control words for a unicycle with a limited FOV camera.

Chapter 3

Model and kinematics

The object of this chapter is to study the mathematical model and the kinematics of the optimization problem that we are concerned with. The main focus is on trajectories that reach the goal, by choosing velocities from the vehicle's control set. Figure 3.1 shows an example of two such trajectories for a simple, car-like vehicle.

In the first section, we give a precise mathematical definition of the problem setting and of what an admissible trajectory is. The second section provides a means of verifying that a trajectory reaches the goal, by showing how to build a vehicle simulator that maps control policies to the configurations reached by applying those control policies. The third section gives a constructive proof that a trajectory that reaches the goal always exists for our model, by developing a simple planner that can reach the goal while applying only two distinct controls. Finally, the last section uses all of these results to prove that time optimal trajectories always exist for the studied system.

3.1 Problem setting

The vehicles we study are rigid bodies that can propel themselves in the unobstructed Euclidean plane. The configuration of a rigid body in the plane is fully given by three quantities: two coordinates for a reference point on the rigid body, with respect to the plane's origin, and an angular quantity that indicates the body's

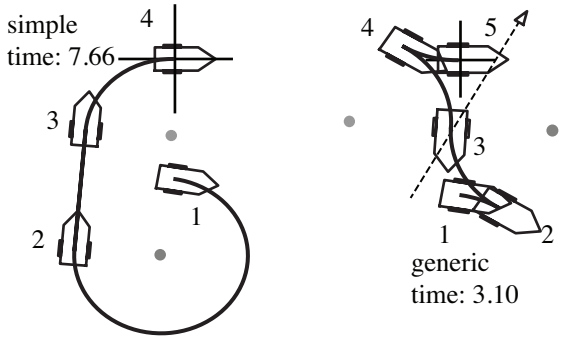


Figure 3.1: Two trajectories that reach the goal for a simple car-like vehicle under control set $U = [-1, 1] \times 0 \times [-1, 1]$. Both trajectories go from configuration $(0.04, -1.55, -0.17)$ to configuration $(0, 0, 0)$. Both control policies begin with a sharp right turn forwards, $\dot{q} = (\dot{x}, \dot{y}, \dot{\theta}) = (1, 0, -1)$.

orientation with respect to the x axis. We collect these three quantities into a *state* vector $q = (x, y, \theta)$.

Our model is fully kinematic, assuming that acceleration happens so fast that its time can be neglected. Thus, the vehicle is controlled by directly choosing velocities of the form $\dot{q} = (\dot{x}, \dot{y}, \dot{\theta})$. The control set U is simply the set of all allowable velocities. We assume this control set to be a convex polyhedron in $\mathbb{R}^3$. Specifying U fully specifies the vehicle's capabilities.

The assumption that the vehicle is self-propelled translates into the fact that U is specified, and constant, in the vehicle's own frame of reference, rather than in the world frame. For instance, if U contains the vector $(\dot{x}, \dot{y}, \dot{\theta}) = (1, 0, 0)$, this indicates that the vehicle can always translate "forwards", no matter how it is oriented. (This assumption does not hold for vehicles that rely on an external source of power, such as a sailboat.) In order to transform a velocity vector from the robot frame to the world frame, when the vehicle's orientation is θ with respect to the x axis, we need to multiply the velocity vector by a rotation matrix:

$$\dot{q}_W = R_\theta \dot{q}_R. \tag{3.1}$$

Here, R_θ is a rotation matrix that is simply augmented with an extra row and column that leave the $\dot{\theta}$ quantity unchanged:

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

The next section discusses transformations between the vehicle frame and the world frame in more detail.

The possibility of choosing among controls raises the issue of what choices are made as time passes. A *control policy* specifies these control choices. In mathematical terms, we define a control policy of duration t_f to be a Lebesgue integrable function $u : [0, t_f) \to U$. The control policy takes three-dimensional values: $u(t) = (u_x(t), u_y(t), u_\theta(t))$. The trajectory corresponding to this control policy is simply defined to be a function of time $q(t)$ that indicates the vehicle's state at each moment in time. In order to obtain the trajectory, given an initial state q_0 and a control policy, the control policy is integrated in two steps. First, we obtain the $\theta(t)$ component of the trajectory:

$$\theta(t) = \theta_0 + \int_0^t u_\theta(\tau) d\tau. \tag{3.3}$$

Next, $\theta(t)$ is used to calculate the other components of the vehicle's state:

$$q(t) = q_0 + \int_0^t R_{\theta(\tau)} u(\tau) d\tau. \tag{3.4}$$

Given initial state q_0 and goal state q_g, a goal-reaching trajectory under control set U is a trajectory $q(t)$, corresponding to some control policy under U, such that $q(t_f) = q_g$.

With these definitions, we can mathematically specify what an instance of optimization problem that we are concerned with is. Such an instance is specified by an initial state q_0, a goal state q_g and a polyhedral convex control set U. (Strictly speaking, it is sufficient to specify q_0, as the goal can always be assumed to be the world frame origin. As we will see in Section 3.3, U also needs to contain at least two controls, out of which at least one needs to be a rotation.) A solution to the problem is a Lebesgue integrable control policy $u(t)$ under U of duration t_m such that

1. The trajectory $q(t)$, starting from q_0 and corresponding to $u(t)$ as characterized by equation 3.3 reaches q_g.

2. There exists no control policy under U of duration $t < t_m$ such that the trajectory starting from q_0 and corresponding to this control policy reaches q_g.

3.2 Trajectories generated by sequences of controls

This section is concerned with computing the positions attained by a vehicle as several distinct controls are applied in sequence. We will proceed in three stages: first by analyzing the stationary vehicle, next by studying motion when a single control is applied, and finally by showing how to compose arbitrary sequences of controls.

3.2.1 Stationary vehicle

Giving a state of the vehicle only specifies the position of a reference point and the vehicle's orientation. Where are the other points of the vehicle located in the world frame, when the vehicle is in this position? The immediate application will be that our simulator becomes capable of feats such as drawing the complete vehicle (including, for example, its wheels) at arbitrary positions in the plane. In the longer run, it is also useful to determine the locations of points of importance for locomotion, such as rotation centers.

The standard toolset used for frame transformations is constituted by homogeneous coordinates. Point (x_p, y_p) in the robot frame gets its position vector expanded by appending 1 to it: $p = (x_p, y_p, 1)^T$. This expanded position vector is then multiplied by frame transformation matrices, in order to obtain the point's position in various frames of reference.

The transformation matrix from the robot frame to the world frame is

$$T_{WR} = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

This transformation matrix is so often used that, when the context is unambigous, we will simply designate it by T. It also contains the same information as the state vector q; for this reason, we will often use T and q in an interchangeable manner. q is easier to interpret, and T is usually more convenient for computations.

Sometimes we need to transform the positions of points in the world frame into positions in the robot frame. For this purpose, we transform these points into homogeneous coordinates as above and multiply their position vectors by the transformation matrix from the world frame to the robot frame:

$$T_{RW} = \begin{bmatrix} \cos\theta & \sin\theta & -x\cos\theta - y\sin\theta \\ -\sin\theta & \cos\theta & x\sin\theta - y\cos\theta \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

In our further analysis, we also consider a reference frame tied to a "control line" in the plane. The control line will be specified, in the world frame, as a line with heading specified by unit-length vector (k_1, k_2) and signed distance k_3 from the origin (k_3 thus being the coordinate of the world frame origin in the control line

frame). Then the transformation matrix from the world frame to the control line frame is

$$T_{LW} = \begin{bmatrix} k_1 & k_2 & 0 \\ -k_2 & k_1 & k_3 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.7}$$

Its inverse is

$$T_{WL} = \begin{bmatrix} k_1 & -k_2 & k_2k_3 \\ k_2 & k_1 & -k_1k_3 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.8}$$

3.2.2 Applying one control

What is the effect of applying, for time t, a velocity vector u, specified in the robot frame, when the vehicle is in world frame configuration q? We separate this question into two parts. First, we show how to transform velocities among frames, and we develop a useful alternative notation for controls in the process. Second, we develop a unified method of integrating velocites, which works for both rotations and translations.

The main idea that we use for transforming velocities among frames is to express these velocities in terms of the corresponding rotation centers. Since these rotation centers are points at easily determined locations in the robot frame, we then simply use the same transformation matrices that we have given above in order to transform velocities as well as points.

If control $u = (\dot{x}, \dot{y}, \dot{\theta})$ is a rotation (i.e., $\dot{\theta} \neq 0$), the rotation center is at location $(-\dot{y}/\dot{\theta}, \dot{x}/\dot{\theta})$. In homogeneous coordinates, it is equivalent to represent this point as either $(-\dot{y}/\dot{\theta}, \dot{x}/\dot{\theta}, 1)$ or

$$c = (-\dot{y}, \dot{x}, \dot{\theta}). \tag{3.9}$$

The second representation is particularly convenient. Not only is it the position of a point, so it can be passed through transformation matrices; but it also contains, as the third component of the vector, the angular velocity $\dot{\theta}$. Since the third component of a homogeneous coordinates vector is left unchanged by multiplications with frame transform matrices, it is possible to thus retrieve the full control (location of rotation center, and the angular velocity at which the rotation proceeds) after passing this representation through the regular frame transformations.

The following lemma places this insight on a mathematical basis, and shows that the method also applies to translations. Note that the rotation center representation of a control, $(-\dot{y}, \dot{x}, \dot{\theta})$ is easily obtained from the standard velocity representation $(\dot{x}, \dot{y}, \dot{\theta})$ by left multiplying the latter with a rotation matrix $R_{\pi/2}$.

Lemma 1 *Assume a moving rigid body in the plane and consider two reference frames, A and B. In frame A, the current velocity is $\dot{q}_A = (\dot{x}_A, \dot{y}_A, \dot{\theta})$ and let T_{BA} be the transformation matrix from A to B. Then the velocity in frame B is*

$$\dot{q}_B = R_{-\pi/2} T_{BA} R_{\pi/2} \dot{q}_A. \tag{3.10}$$

Proof: If $\dot{q}$ is a translation, T_{BA} has the same effect as a pure rotation matrix. Therefore the right-hand side of equation 3.10 has the same effect as an application of T_{BA} to the velocity vector, which is the correct result.

If $\dot{q}$ is a rotation, $R_{\pi/2}\dot{q}_A$ gives us the A frame coordinates of the rotation center. Passing these coordinates through T_{BA} gives us the B frame coordinates of the rotation center, and then we apply a $R_{-\pi/2}$ matrix to obtain the B frame velocity. ■

This lemma suggests that it will often be convenient, for transformation purposes, to represent controls in a rotation center notation. For any control $u = \dot{q}$, we define the rotation center representation to be

$$c = R_{\pi/2}u. \tag{3.11}$$

With this notation, equation 3.10 becomes

$$c_B = T_{BA}c_A. \tag{3.12}$$

We have, at this point, achieved our first objective for the current section, i.e. the transformation of velocities among reference frames. We will now proceed with the second objective: showing how to integrate a given velocity for an arbitrary time t.

Knowing the position of the rotation center allows integration of the control for an arbitrary point in the robot frame. Selig [23] gives us the rotation matrix around point **c**:

$$\begin{bmatrix} R & (I_2 - R)\mathbf{c} \\ 0 & 1 \end{bmatrix}, \tag{3.13}$$

where R is a $2X2$ rotation matrix and I_2 is the identity matrix.

By replacing into equation 3.13 the coordinates of the rotation center from equation 3.9, we obtain the following transformation matrix that corresponds to the application of control $u = (\dot{x}, \dot{y}, \dot{\theta})$ for time t:

$$T(u,t) = \begin{bmatrix} \cos\dot{\theta}t & -\sin\dot{\theta}t & \dot{x}t\ \mathrm{sinc}\ \dot{\theta}t - \dot{y}t\ \mathrm{verc}\ \dot{\theta}t \\ \sin\dot{\theta}t & \cos\dot{\theta}t & \dot{x}t\ \mathrm{verc}\ \dot{\theta}t + \dot{y}t\ \mathrm{sinc}\ \dot{\theta}t \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.14}$$

where

$$\mathrm{sinc}\ (x) = \begin{cases} \frac{\sin x}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \tag{3.15}$$

is the well-known cardinal sine function and

$$\mathrm{verc}\ (x) = \begin{cases} \frac{1-\cos x}{x}, & x \neq 0 \\ 0, & x = 0 \end{cases} \tag{3.16}$$

is a differentiable "cardinal versine" function, developed by analogy with the cardinal sine.

This result has some implications that are not necessarily limited to the study of optimal trajectories. Since both the cardinal sine and the cardinal versine are defined for $\dot{\theta} = 0$, Formula 3.14 also works for integrating translations and we are, in effect, presenting a unified method for integrating both rotations and translations in the plane. The smooth behavior of this formula around the $\dot{\theta} = 0$ point leads to increased numerical stability for the analysis of vehicles that sometimes move in almost a straight line (e.g., wheeled vehicles for which the wheel diameters are not exactly equal). Furthermore, the following section will show that the integration matrices $T(u,t)$ can be easily composed, which leads to facile modeling of sequences

of rotations and translations.

The final result of this section will be an analysis of the velocities achieved by arbitrary points in the vehicle frame, when a control is applied. This operation is necessary if we wish to update the control space to reflect a change in reference point. We determine a matrix that can be multiplied by any new reference point's coordinates in the old frame, in order to determine its $(\dot{x}, \dot{y})$ velocity vectory (the $\dot{\theta}$ velocity always being the same as that of the original control). In order to build this matrix, we first translate the rotation center to the origin and then apply a skew-symmetric matrix. We obtain the following:

$$S = \begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dot{y}/\dot{\theta} \\ 0 & 1 & -\dot{x}/\dot{\theta} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\theta} & \dot{x} \\ \dot{\theta} & 0 & \dot{y} \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.17}$$

This matrix is then multiplied by the new reference point's coordinates to obtain the reference point's velocity, i.e. the $\dot{x}$ and $\dot{y}$ components of the control in the new frame. Note that this transformation matrix is also valid for translations ($\dot{\theta} = 0$).

3.2.3 Sequences of controls

A trajectory with a piecewise constant control law can be given as a sequence of (u_i, t_i) pairs, where each consecutive control $u_i = (\dot{x}_i, \dot{y}_i, \dot{\theta}_i)$ is applied for time t_i. Given such a sequence, we first assemble the $T(u_i, t_i)$ integration matrices as above. These matrices compose by post-multiplication. Thus, the final state of trajectory $[(u_1, t_1), (u_2, t_2), \ldots (u_n, t_n)]$, starting from state q_0 (equivalently specified by the robot frame to world frame transform matrix T_0) is:

$$T_f = T_0 T(u_1, t_1) T(u_2, t_2) \ldots T(u_n, t_n). \tag{3.18}$$

The active control at time $t \leq \sum_{j=0}^{n} t_j$ is the control corresponding to the largest index k such that $t \geq \sum_{j=0}^{k} t_j$. The state at time t is thus:

$$T(t) = T_0 T(u_1, t_1) T(u_2, t_2) \ldots T(u_k, t_k) T(u_{k+1}, t'), \tag{3.19}$$

where k is the largest index such that $t' = t - \sum_{j=0}^{k} t_j \geq 0$. This equation proves our initial statement in this section: if a trajectory has piecewise constant controls, its representation as a sequence of (control, time) pairs is equivalent with our earlier definition of a trajectory as a $q(t)$ function.

3.3 A simple method for finding a trajectory that reaches the goal

Being able to find a trajectory that reaches the goal is an important step in our work. The existence of such a trajectory is a critical requirement for the proof of the existence of optimal trajectories at the end of this chapter. Furthermore, search methods described later in this work construct trajectories and test if those trajectories pass close enough to the goal. Consideration of a trajectory is terminated if it has not yet reached the goal in time given by a known trajectory to the goal.

In this section we describe a simple and fast technique for always finding a trajectory to the goal, so that controllability is established and there is always an upper bound on trajectory time. We will see that such a trajectory to the goal always exists, even if we have as few as two controls (without both of them being translations).

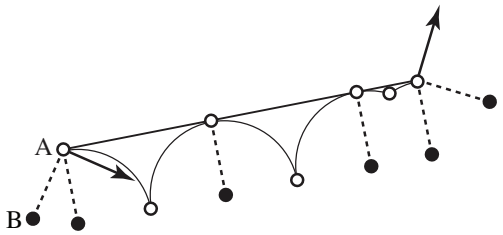


Figure 3.2: A trajectory found for a robot with just two rotation controls centers (black and white circles). Start and goal configurations are given by arrows; the path of the white rotation center is shown.

The basic idea, if we have a rotation and a translation available, is to build a turn-drive-turn trajectory. We use the rotation to achieve the proper orientation, drive "straight" with the translation, and turn again at the goal. If only two rotations are available, we will show a way to simulate the "driving straight" part by alternating these two rotations (see Fig. 3.2 for an example). By careful choice of rotation directions and careful choice of the controls used to construct the translation segment (in the case that there are more than two controls available), this method can be used to quickly construct trajectories that are often not too much slower than the optimal.

If a trajectory to any goal always exists, we will say that the system is *controllable*. The following lemma gives the geometry behind the main result of this section:

Lemma 2 *A rigid body, controlled by velocities chosen from a set U that is constant in the body's own frame of reference, is controllable in SE(2) if and only if U contains two or more distinct velocities, at least one of which is a rotation.*

The remainder of this section is dedicated to a constructive proof of this lemma.

It is evident that the body is not controllable if either U has cardinality less than two or U only contains translations. We will next give a constructive proof for the converse.

Let u_1 and u_2 be two distinct controls in U. Without loss of generality, assume u_1 is a rotation of center c_1. Let c_{10} and c_{1f} be the positions, in the world frame, of c_1 at states q_0 and q_f respectively. Let g be the displacement vector $c_{1f} - c_{10}$. There are two cases, according to whether u_2 is a translation.

The two cases will be discussed in detail below. The basic idea of the proof is the folllowing. If u_2 is a translation, then the trajectory will be easy to construct: choose the reference point of the robot to be centered on the rotation center, spin in place until the translation direction is lined up with the vector from the start to the goal, drive to the goal, and rotate to the required angle. If both velocities are rotations, we will replace the translation section of the trajectory with a sequence of rotations.

u_2 is a translation

The goal is achieved by a turn-drive-turn trajectory. Let $v_0 = (\dot{x}_{20}, \dot{y}_{2f})$ be the velocity vector corresponding to u_2 at the initial state and v_f be the velocity vector corresponding to u_2 at the final state. We connect q_0 to q_f as follows:

1. Apply u_1 until v_0 becomes parallel to g.
2. Apply u_2 until the displacement distance $||g||$ is achieved.
3. Apply u_1 until g becomes parallel to v_f.

u_2 is a rotation

We will develop a controller very similar to the one above, by simulating the middle translation with an alternation of the two rotations provided. Thus let v_0 be the velocity vector of c_{10} when u_2 is applied at q_0, and let v_f be the velocity vector of c_{1f} when u_2 is applied at q_f. We connect q_0 to q_f thus:

1. Apply u_1 until v_0 becomes parallel to g.
2. Apply an alternation of u_2 and u_1 until the translational displacement $||g||$ is achieved in the direction of the vector g.
3. Apply u_1 until g becomes parallel to v_f.

We still need to show how to achieve the middle step. Let A and B represent rotation centers (see Fig. 3.2); choose the origin of the robot frame coincident with A. Let the distance between the two rotation centers be l. Rotate B around A until the line segment from A to B is perpendicular to the line from the start to the goal. Then repeat a series of segments, where each segment is of the form $B_{\pi/2}A_{\pi}B_{\pi/2}$, achieving a pure translation of distance $2l$ in the direction of the line segment from start to goal. If such a translation would overshoot the goal, adjust the angles in the BAB segment to exactly reach the goal (details left to reader); finally rotate about A to the goal angle.

To translate for a distance $d < 2l$, we apply the following sequence of actions. Align the world x axis with the $c_{20} - c_{10}$ vector. Then

1. Apply u_2 until c_1 achieves a y coordinate of $d/2$.
2. Apply u_1 until c_2 achieves a y coordinate of d.
3. Apply u_2 until c_1 achieves a y coordinate of d.

Thus we obtain an alternation of u_1 and u_2 that translates the body for a displacement of g. With this module, the universal planner is complete. This concludes our constructive proof of Lemma 2.

3.4 Existence of optimal trajectories

The existence of optimal trajectories for the system studied in this work is quickly implied by a corollary to Fillipov's existence theorem given in [24] (pp. 98 - 99). We slightly restate this corollary with the notation used in the present work, as follows.

Let q_0 and q_g be two states in $\mathbb{SE}(2)$. If all the following conditions are satisfied, there exists a minimum time trajectory under U from q_0 to q_g:

1. There exists a function $g(q)$ such that $\dot{q} = g(q)u$.
2. $g(q)$ is locally Lipschitz continuous.
3. The control set U is a compact convex subset of $\mathbb{R}^m$.
4. There exists an admissible trajectory from q_0 to q_g.

5. Given any initial state q_0 and control law $u(t)$, there exists a corresponding trajectory $q(t)$, defined for the whole duration of the given control law.

It is easily shown that our system verifies these conditions:

1. In our case, $g(q) = R_\theta$ (see eq. 3.3).
2. $R_\theta(q)$ is locally Lipschitz continuous as it only contains the functions $\sin\theta$ and $\cos\theta$, which are Lipschitz continuous.
3. In our case, U is a closed convex polyhedron in $\mathbb{R}^m$.
4. The existence of an admissible trajectory has been proven in the preceding section.
5. Since $u(t)$ is Lebesgue integrable, the integral in equation 3.3 always exists, and thus there always exists a trajectory $q(t)$ corresponding to $u(t)$.

Therefore, optimal trajectories always exist for the system model analyzed in this work. The next chapter will apply the methods of optimal control theory to study these optimal trajectories.

Chapter 4

Necessary conditions for optimality

We start our analysis of optimal trajectories by applying the Pontryagin Principle to the problem stated in Section 3.1. This application immediately separates optimal trajectories into two classes, one of which (whirls) is less general, as it only contains trajectories with a constant angular velocity. We develop a geometric, local condition on the optimal control policy for trajectories belonging to the general case, and based on this condition we call such trajectories *control line trajectories*. For the particular case of the *whirls*, we use a modified version of the Pontryagin Principle to completely solve this case, thus leaving control line trajectories the sole subject of subsequent chapters.

4.1 The Pontryagin Principle

The Pontryagin Principle [18] places several strong necessary conditions on optimal control policies. The conditions are local in trajectory space, and the entire approach may be placed in analogy with the study of the maxima and minima of differentiable real functions. At all such extremal points, the derivative of the function must be zero. After solving the equation $f'(x) = 0$, the value of the function is compared at all the points thus found in order to find the global maximum or minimum.

Analogously, consider the space of all valid trajectories that reach the goal. Let $f(x)$ be a function that computes the time of a trajectory x. The Pontryagin Principle is a local condition in this space, corresponding to the $f'(x) = 0$ condition in simpler spaces. We call the trajectories that satisfy the Pontryagin Principle *extremal*, and there is usually a limited number of them, just as usually $f'(x) = 0$ has a limited number of solutions.

Our approach is to find all extremal trajectories linking a start to a goal, and to compare their times in order to pick the fastest. We will find that there exist several very distinct types of extremal trajectories, and different methods will be needed for constructing a shortest time trajectory within each of these classes.

Let us now examine the details. The Pontryagin Principle requires, first, that for each time optimal trajectory $q(t)$ there must exist a corresponding *adjoint* $\lambda(t)$ that represents a privileged direction in velocity space at each point on the trajectory. All along the optimal trajectory, the control applied must maximize, among all possible controls, the dot product of the adjoint and the generalized velocity of the body. Second, this dot product $H(t) = \langle \lambda(t), \dot{q}(t) \rangle$, called the *Hamiltonian* of the optimal trajectory, is a constant function $H(t) = \lambda_0 > 0$.

Finally, the Pontryagin Principle places restrictions on the adjoint. $\lambda(t)$ must be a non-zero continuous

function and must satisfy the following differential equation:

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial q}. \tag{4.1}$$

For a given vehicle control space U, we will call the *extremal ensemble* of an extremal trajectory $q(t)$ the components of the proof that it is indeed extremal: the control policy $u(t)$, the adjoint $\lambda(t)$ and the value of the Hamiltonian. Only trajectories for which an extremal ensemble exists can be time optimal.

In the following, we integrate the Pontryagin differential equation for the adjoint, as applied to kinematic rigid bodies in the plane, and we examine the immediate implications of the formula thus obtained.

4.1.1 Integration of the adjoint function

This section is concerned with proving the following theorem:

Theorem 1 *Consider a rigid body with an attached convex polygonal control set U, moving in the unobstructed plane on a trajectory $q(t) = (x(t), y(t), \theta(t))$ from initial state $q_0 = q(0)$ to final state $q_f = q(t_f)$. The trajectory is generated by an acceptable control policy $u(t)$ according to the definitions in Section 3.1.*

If control policy $u(t)$ is time optimal among all acceptable control policies that generate trajectories from q_0 to q_f, then there exist constants k_1, k_2 and k_3 such that at every time t the value of the control policy $u(t)$ is a point in U that maximizes the value of the Hamiltonian function

$$H(t) = \begin{pmatrix} k_1 \\ k_2 \\ k_1 y(t) - k_2 x(t) + k_3 \end{pmatrix}^T R_{\theta(t)} u(t). \tag{4.2}$$

Furthermore, the Hamiltonian is a constant function $H(t) = \lambda_0 > 0$.

Proof: At any time t, from the Pontryagin Principle (as stated above in equation 4.1), the adjoint equation is

$$\dot{\lambda} = -\frac{\partial}{\partial q}\langle \lambda, \dot{q}(q,u)\rangle \tag{4.3}$$

$$= -\begin{pmatrix} 0 \\ 0 \\ \lambda^T (\frac{\partial}{\partial \theta} R_\theta) u \end{pmatrix}. \tag{4.4}$$

The zeros occur because $\dot{q} = R_\theta u$ does not depend on $\dot{x}$ or $\dot{y}$. Therefore, by direct integration, $\lambda_1 = k_1$ and $\lambda_2 = k_2$. Let $(u_x, u_y, u_\theta) = u$ and substitute these values back into the definition for $\dot{\lambda}_3$:

$$\dot{\lambda}_3 = k_1(s u_x + c u_y) - k_2(c u_x - s u_y), \tag{4.5}$$

where c and s are shorthand for $\cos\theta$ and $\sin\theta$. From equation 3.1,

$$\dot{x} = c u_x - s u_y \tag{4.6}$$

$$\dot{y} = s u_x + c u_y \tag{4.7}$$

Substitute into equation 4.5,

$$\dot{\lambda}_3 = k_1\dot{y} - k_2\dot{x}, \tag{4.8}$$

and integrate:

$$\lambda_3 = k_1 y - k_2 x + k_3. \tag{4.9}$$

The Hamiltonian to be maximized along time-optimal trajectories is thus

$$H = k_1\dot{x} + k_2\dot{y} + \dot{\theta}(k_1 y - k_2 x + k_3). \tag{4.10}$$

■

4.1.2 The control line

Since the adjoint must be non-null, it is not possible for any trajectory to have an adjoint with all three integration constants equal to zero. We will, however, distinguish two kinds of adjoints, and two corresponding kinds of time optimal trajectories.

For adjoints with $k_1 = k_2 = 0$, the control $u = (u_x, u_y, u_\theta)$ only needs to maximize $H = k_3 u_\theta$. Thus, for all control policies that use exclusively either controls from U of maximum angular velocity ($k_3 > 0$) or of minimum angular velocity ($k_3 < 0$), it is always possible to find a very simple adjoint that verifies the Pontryagin Principle. Let this kind of trajectories be called *whirls*. This particular case does not have much in common with the rest of our analysis (besides the kinematic model already discussed), and, since it is thus self-contained, it will be solved separately in the next section.

In the rest of the current work, we study the more general case where at least one of k_1 or k_2 is not null. Since the conditions set by the Pontryagin Principle are invariant with respect to scaling of the adjoint by a positive constant, we assume without loss of generality, in this case, that $k_1^2 + k_2^2 = 1$.

The optimal trajectories that admit this kind of adjoint have a peculiar geometrical property. Define the *control line* to be a line with heading (k_1, k_2), and signed distance k_3 from the origin. The first part of the Hamiltonian

$$k_1\dot{x} + k_2\dot{y} \tag{4.11}$$

is the component of the translational velocity of the rigid body along the vector (k_1, k_2), and the term $-k_2 x + k_1 y + k_3$ is the distance from the reference point of the rigid body to the control line.

Therefore, let the the more general type of optimal trajectories that are not whirls be called *control line trajectories*.

We thus have a geometric interpretation of the Hamiltonian for non-whirls. Define the "control line frame" L to be a frame attached to the control line with the x axis aligned with the control line (see Fig. 4.1). Then in the control line frame, y_L is the distance of the rigid body from the control line, and θ_L is the angle the body frame makes with the control line. $\dot{x}_L$ is the component of the body's velocity along the control line. In these coordinates, the Hamiltonian of a control that imposes velocity that resolves to $(\dot{x}_L, \dot{y}_L, \dot{\theta})$ in the control line frame is

$$H = \dot{x}_L + y_L\dot{\theta}. \tag{4.12}$$

It will sometimes be convenient to write the above expression of the Hamiltonian for controls expressed in polar coordinates. For a control that sets velocity v in direction α and angular velocity ω in the frame of the

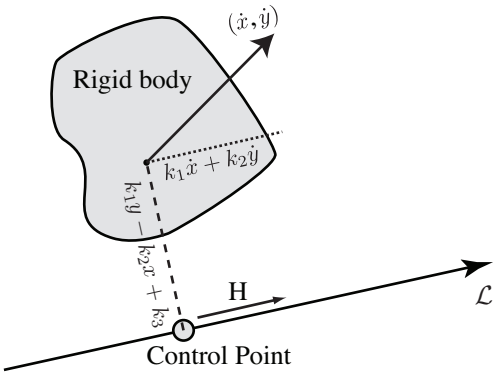


Figure 4.1: A rigid body instantaneously following a control line optimal trajectory. Two possible reference points attached to the rigid body are shown. The optimal control law needs to choose a control that maximizes the Hamiltonian with respect to the control line. For a reference point crossing the control line, the Hamiltonian is equal to the component of this point's velocity that is parallel to the control line.

body, (thus $u_x = v\cos\alpha$, $u_y = v\sin\alpha$ and $u_\theta = \omega$), the following formula is obtained immediately:

$$H = v\cos(\theta_L + \alpha) + y_L\omega. \tag{4.13}$$

In particular, if the reference point on the vehicle happens to be crossing the control line at some time, then $y_L = 0$ and the necessary condition is to simply choose a control that maximizes the reference point's velocity in the control line's direction (see Figure 4.1).

As the following lemma shows, we are, in fact, able to change the point of reference for the purpose of calculating the Hamiltonian along a trajectory.

Lemma 3 *Given a rigid body trajectory that obeys the Pontryagin Principle, the same value of the Hamiltonian will be obtained for any point of reference in the body frame.*

Proof: Instantaneous motions for planar rigid bodies can be either rotations or translations. If the instantaneous motion is a translation, the result is immediate. Let the instantaneous motion be a rotation of center O and angular velocity ω. Let y_O be the distance between O and the control line. Let O' be O's projection onto the control line and P be an arbitrary point in the frame of the vehicle. From the perspective of P, the Hamiltonian is

$$H_P = \dot{x}_P + y_P\omega = ||OP||\omega\cos\angle POO' + y_P\omega \tag{4.14}$$

$$H_P = (||OP||\cos\angle POO' + y_P)\omega = y_O\omega = H_O. \tag{4.15}$$

Therefore calculating the Hamiltonian at any point in the frame of the body is the same as calculating it at the center of rotation. ■

An interesting result is obtained by choosing the reference point, during a rotation control, to be the rotation center, for which $\dot{x} = 0$. Therefore, given a value for H, we can compute the distance of the active rotation center from the control line. A similar result is obtained describing the angle a translation makes with the control line in terms of the value of H.

Corollary 1 *If the control corresponding to rotation center O and angular velocity ω is active at time t on an extremal trajectory of Hamiltonian value H, then at this time the signed distance from O to the control line is $y_O = \frac{H}{\omega}$.*

Corollary 2 *If a translation control of velocity v and forming angle α with the horizontal axis is active at time t on an extremal trajectory of Hamiltonian value H, then at this time $\cos(\alpha + \theta) = \frac{H}{v}$, where θ is the orientation of the body frame with respect to the control line.*

4.2 Whirls

Since, in the case of whirls, two constants from equation 4.2 are null ($k_1 = k_2 = 0$), the expression of the Hamiltonian is simple:

$$H = k_3\dot{\theta}. \tag{4.16}$$

Either k_3 is greater than zero, or less than zero. ($k_3 \neq 0$, since the Pontryagin Principle restricts H from being identically zero.) If k_3 is positive, then any control with maximum $\dot{\theta}$ satisfies the Pontryagin Principle. Otherwise, any control with minimum $\dot{\theta}$ satisfies the principle.

In the simplest case, the controls for which the minimum and maximum values of $\dot{\theta}$ are attained are unique. Then these trajectories are simple: constant controls, corresponding to pure rotations around a fixed rotation center. The more interesting case is when multiple controls maximize or minimize $\dot{\theta}$. The Pontryagin Principle, as applied above, does not directly give any information about when to switch between the controls in this case.

Under what circumstances might such a trajectory be optimal? The classic example is a Reeds-Shepp car that can reverse as well as go forwards. Consider the goal of spinning this car in place. A direct spin is not an available control, and a human driver would execute a three-point turn. The driver might move forwards around the left rotation center, with positive angular velocity, then backwards around the right rotation center, with positive angular velocity, then forwards again around the left rotation center.

How long does the three-point turn take? It is simply the angle to be traversed divided by the angular velocity. Assuming that the car is controlled by an electronic system that can effect very quick control changes, it could also follow a four-, five-, or six-point turn, taking the same time, but following a very different trajectory. Therefore, we expect that there may be many optimal trajectories between configurations for which the amount of angle to turn through is the limiting factor, rather than the distance to be travelled. Rather than constructing all such trajectories, we will show that there is a canonical trajectory structure, which we call 'roll-and-catch' (see Fig. 4.2 for an example) that we can use to always find one optimal trajectory. We also show that that for this canonical trajectory structure, we can find the precise control policy of the optimal trajectory for every start and goal.

Since the original control space U is a convex polyhedron, all the controls that can be used for whirls are on a single polygonal face of this polyhedron. The problem of finding the optimal whirl trajectories can be restated equivalently in the following way. Consider a convex polygonal surface of rotation centers Z in the plane, containing at least two distinct points, and a vehicle that this surface is attached to. The vehicle can rotate at angular velocity 1 around any point in Z. (The clockwise case is symmetric.) Find an algorithm to construct an optimal trajectory for given start and end configurations, q_0 and q_f respectively.

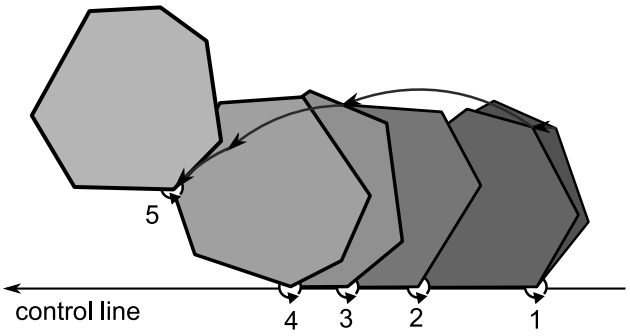


Figure 4.2: Example of a roll and catch trajectory. The polygonal control surface rolls along the control axis with constant angular velocity. When the last rotation center is put in place, the last motion is an off-axis rotation around this point (the "catch" stage). The trajectory of the last rotation center is shown, as well as the locations in the world frame of all the rotation centers used along the trajectory.

4.2.1 A sufficient family of whirls for optimality

In a direct application, the Pontryagin Principle does not place any constraints on whirl trajectories. We will identify a class of optimal trajectories that always exist and are composed of two stages, with the objective of applying the Pontryagin Principle to characterize the shape of the first stage. Given a whirling vehicle with convex control surface Z, we will call an *xy stage trajectory for point A* ($A \in Z$) a trajectory between two configurations q_0, q_f that satisfies the following two conditions:

1. The first stage of the trajectory places A on its correct position in q_f in as short a time as possible. We will call this the xy stage, as only the x and y coordinates for A need to be attained.

2. The second stage of the trajectory is a rotation around A, until q_f is attained.

xy stage trajectories always exist between a given pair of configurations, as the vehicle is controllable. Given two such configurations, q_0 and q_f, consider an optimal trajectory and an xy stage trajectory between them, respectively. Let t_f be the time taken by the optimal trajectory, let t_1 and t_2 be the respective times taken by the two stages of the xy stage trajectory. Since the optimal trajectory does place A in its correct location, $t_1 \leq t_f$; therefore $t_f \leq t_1 + t_2 < t_f + 2\pi$. But the times of two trajectories between the same pair of configurations must differ by a multiple of 2π; therefore $t_1 + t_2 = t_f$ and the xy stage trajectory is optimal. xy stage trajectories are therefore a class of optimal trajectories that always exist. In the following, we will confine our efforts to characterizing this class of optimal trajectories and finding a method to always construct one such trajectory.

4.2.2 The non-autonomous version of the Pontryagin Principle

The configuration space for the xy stage is two-dimensional, containing only the x and y coordinates. This makes it possible to remove the θ coordinate from the state, and re-apply the Pontryagin Principle. However, removing θ from the state makes the configuration space velocity depend on time: $\dot{q} = f(q, u, t)$. To deal with this problem, we will apply the non-autonomous version of the Pontryagin Principle.

The Pontryagin Principle for time-optimal trajectories for non-autonomous vehicles ([18] p. 60) is very similar to the version used previously in this work, with the exception that the function H is only required to be positive and not necessarily constant. Taking the final rotation center as a reference point in the frame

of the body, we obtain the condition that, along the xy stage, the function $H(x, u, t)$ needs to be maximized by the chosen control at each point along the trajectory, where

$$H = \lambda_1 \dot{x} + \lambda_2 \dot{y} \tag{4.17}$$

and the (λ_1, λ_2) vector is non-null. Since $\frac{d\lambda_1}{dt} = \frac{\partial H}{\partial x} = 0$, $\lambda_1(t)$ is a constant function. Similarly, $\lambda_2(t)$ is a constant as well, therefore

$$H = k_1 \dot{x} + k_2 \dot{y}, \tag{4.18}$$

with $k_1^2 + k_2^2 > 0$. Therefore, for each xy stage optimal trajectory, there exists a control direction in the plane, given by the vector (k_1, k_2) such that the optimal control policy, at all times, chooses a control that maximizes the body's velocity component in this direction. Since the control velocities form a polygon that is always turning at constant velocity, it follows that optimal control policies are piecewise constant.

4.2.3 Shape of the xy stage

Consider, along an xy stage optimal trajectory, the time when the control switches from u_i to u_{i+1}, corresponding to rotation centers R_i and R_{i+1} respectively. Consider the functions $H_i = H(x, u_i, t)$ and $H_{i+1} = H(x, u_{i+1}, t)$. Both these functions are continuous. Immediately before the switch, $H_i \geq H_{i+1}$ and immediately after the switch $H_i \leq H_{i+1}$. Therefore, H_i and H_{i+1} are equal at the time of the switch.

Furthermore, at the switching time, let P be the reference point and v_i and v_{i+1} its velocity vectors immediately before and after the switch respectively. Since the angular velocity is constant, the lengths of the vectors v_i and v_{i+1} are proportional to the lengths of the segments PR_i and PR_{i+1} respectively. The angle between the pre-switch and post-switch velocity vectors is furthermore equal to the angle $\angle R_i P R_{i+1}$, as the velocities are perpendicular to the radii. So the triangle formed by the two velocity vectors is proportional to the triangle $\Delta R_i P R_{i+1}$ and these two triangles form an angle of $\frac{\pi}{2}$.

Since $H_i = H_{i+1}$ at the time of the switch, and these two functions are the projections of v_i and v_{i+1} onto the control direction, the third side of the triangle formed by v_i and v_{i+1} is perpendicular onto the control direction. This side corresponds to $R_i R_{i+1}$ in the proportional and rotated by $\frac{pi}{2}$ triangle; therefore $R_i R_{i+1}$ is parallel to the control direction. This holds for all switches along the xy stage. Therefore, in the world frame, all the rotation centers used during the xy phase are found on a line passing through the first rotation center and parallel to the control direction.

Setting the world reference frame on this axis, we notice by a similar argument that, if R_j is placed higher, in respect to the control line, than R_k, then $H_j > H_k$. Since the first rotation center used is on the control line, assuming the control direction points right to left, all the other rotation centers must be above the control line in the initial state; this condition is then propagated along the trajectory. We have therefore proven the following:

Lemma 4 *For each xy stage optimal trajectory, there exists a* control line *such that the xy stage is a rolling of control surface Z in the positive direction along the control line.*

Thus, we have found that a control line exists even for a subclass of whirls that is sufficient for optimality. Because the xy stage is thus shown to be analogous to a sideways view of a rolling motion on a flat surface, with a subsequent "catch" on the final rotation center, we will alternatively call xy stage trajectories "roll and catch" trajectories.

4.2.4 The position of the control line for known initial and final controls

Number the corners of the convex hull consecutively clockwise as R_1, R_2, ..., R_m. Assume we knew that the initial control is a rotation around R_1 and the final two controls are rotations around R_k and R_f respectively. The optimal control is therefore $R_1, R_2, \cdots, R_m$ repeated n times (where n is an unknown integer) and then $R_1, R_2, \cdots, R_{k-1}, R_k, R_f$.

In the world frame, let R'_1 be the initial location of R_1, let R'_f be the final location of R_f and let R'_k be the location, at the final switch, of R_k. We are given the position of R'_1, and we know the position of R'_f as the final motion is a rotation around this point. Let d'_{1f} be the length of the segment $R'_1R'_f$. In order to determine the structure of the trajectory (if it exists), it is sufficient to find the positon of R'_k, which determines the control line $R'_1R'_k$.

Let r_{ij} be the distance, in the body frame, between two arbitrary rotation centers R_i and R_j. Let $l_i = r_{i,i+1}$, i.e. the length of the ith side of the control surface. Let $p = \sum_{i=1}^{m} l_i$ be the perimeter of the control surface Z. Since the trajectory is a roll along the control line, the length of the segment $R'_1R'_k$ is:

$$d'_{1k} = np + l_1 + l_2 + \cdots + l_{k-1}. \tag{4.19}$$

In the triangle $\Delta R'_1R'_kR'_f$, the triangle inequality must hold:

$$d' - r_{kf} < np + l_1 + l_2 + \cdots + l_{k-1} \leq d' + r_{kf}. \tag{4.20}$$

The left-hand side is a strict inequality because, as shown above, if two rotation centers are on the control line at the same time, the one that is used on the immediately preceding interval must have a smaller x coordinate.

Since r_{kf} is a section through Z, $2r_{kf} \leq p$. Note that Relation 4.20 has a span of $2r_{kf}$ between the leftmost and rightmost side, and the middle changes in increments of p. Therefore Relation 4.20 has at most one solution for the unknown integer n, which is obtained by subtracting and dividing appropriately and taking the floor function:

$$n = \lfloor \frac{d' + r_{kf} - (l_1 + l_2 + \cdots + l_{k-1}}{p}) \rfloor. \tag{4.21}$$

Furthermore, n needs to satisfy the left-hand side of 4.20 above. By replacing this solution into equation 4.19 above, we determine the length of d'_{1k}. This fully determines the triangle $\Delta R'_1R'_kR'_f$ and, by extension, the position of the control line and the structure of the trajectory. In order for the xy stage to be extremal, we only need to check observance of the Pontryagin Principle at its final point, i.e. calculate the configuration at the switch from R_k to R_f and verify that no point of Z is above the control line in this configuration.

Since there is at most one solution for the location of the control line, if such a solution exists then the corresponding xy stage is the fastest way to get R_f into its final position by using R_1 as the first rotation center and R_k as the last.

4.2.5 Constructing an xy stage trajectory for given initial and final configurations

Given a control surface and initial and final configurations q_0 and q_f respectively, we have proven above that there exists a "roll and catch" optimal trajectory between these configurations, for any choice of reference point on the control surface (the reference point being the location of the last rotation center that is used on the trajectory). We have also shown how to find this trajectory, if the initial and the second to last controls

were known; these two controls have to be rotations around corners of the control surface, which we have shown can be assumed without loss of generality to be a convex polygon. Therefore, the following simple algorithm is valid:

1. Enumerate all possible ordered pairs of corners of the control surface.
2. For each such pair, construct the "roll and catch" trajectory (there can exist at most one) that corresponds to the chosen initial and second to last controls.
3. Pick the fastest trajectory.

The algorithm runs in time that is $O(m^2)$, where m is the number of corners of the polygonal control surface. For any given control surface, the running time is constant.

This result concludes our analysis of whirls. For the rest of the current work, we limit ourselves to the case of control line trajectories.

Chapter 5

A class of control policies sufficient for optimality

The sole requirement that the control policy $u(t)$ be Lebesgue integrable leaves open a wide variety of candidate solutions for the optimal control problem defined in Section 3.1. A direct search of this infinitely ($\aleph_1$) dimensional space is not feasible. In this chapter, we considerably narrow down the search space by characterizing a class of control policies that are sufficient for optimality. These control policies are piecewise continuous and only use a finite set of controls. We will begin by presenting the main result and some applications, and will use the second half of the chapter to prove the result by a sequence of lemmas.

5.1 Control space discretization

The main result that will be proven in this chapter is the following:

Theorem 2 *There exists a canonical finite subset U_c of control set U, containing the vertices of U and at most one point on each face or edge of U that intersects the $\dot{\theta} = 0$ plane, such that any optimal control problem of the type defined in Section 3.1 has a solution that is a control policy that*

1. *is piecewise continuous*
2. *only takes values in U_c.*

The proof for this result will be given in the second half of the chapter. In the first half, we will first examine how optimal control policies that are not piecewise constant can occur in some cases, which will reinforce our motivation to search for piecewise constant policies. Second, we will show how control space discretization and the existence of piecewise constant control policies are intertwined, by discussing a few examples for the application of Theorem 2 to several well-studied vehicles.

5.1.1 Optimal control policies that are not piecewise constant

Control policies that are not piecewise constant can be difficult to implement in practice. They are, however, mathematically optimal in some cases.

For example, consider a rigid body in the plane that can translate in any of the four directions aligned with the axes, north, south, east, and west, with speed one, as shown in Figure 5.1. The fastest trajectory

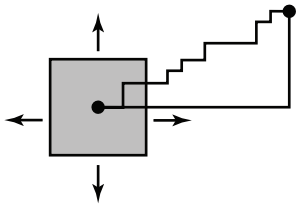


Figure 5.1: Two optimal trajectories for a translational platform.

to move to a point to the east is unique and simple to describe: drive east, with no control switches. How about a trajectory to drive to the point $(2, 1)$? The minimum time required is 3 (the Manhattan distance to the point from the origin), and one optimal trajectory is to drive east for time two, then north for time one. However, any other trajectory to the goal that uses only the controls 'east' and 'north' is also optimal, even if these controls are applied in a manner that is not piecewise continuous. For this system, optimal trajectories clearly exist; but it is easiest to describe the trajectories with a minimum number of switches, with the understanding that permutations of translation sections of the trajectory will also be optimal.

There are worse situations that can arise. Consider a refrigerator whose projection onto the plane is a square, with supporting legs at the vertices of the square. Assume we can rotate the refrigerator about any of the legs with angular velocity 1 or -1. What is the fastest way to move the refrigerator along the positive x axis? It turns out that the solution may be to "chatter", or switch infinitely quickly, between two rotation controls, approximating a straight-line motion.

We will show that, because of the convexity requirement for the control space, it is always possible to find a piecewise constant optimal control policy. This is particularly easy to see for the refrigerator example above: since the control space includes a hard turn left and a hard turn right, it must also include all intermediate controls between these two, and in particular a straight line translation that can be used to replace any "chattering" segment. For the translational platform, we show that all "chattering" segments can be replaced by a piecewise constant control policy with at most one switch.

In general, we will show that for any polyhedral convex control space U it is possible to find a finite subset of the control space that is sufficient for optimality, and furthermore that it is always possible to find a piecewise constant optimal control policy corresponding to this finite subset.

5.1.2 Examples of control space discretization

Before presenting a proof for Theorem 2, let us examine the way it applies to several well-known vehicles. This sequence of examples will also show how control space discretization and the existence of piecewise constant control laws are intertwined in our analysis. We begin with the Dubins car, for which the control space (see Fig. 5.2) is the line segment $1 \times 0 \times [-1, 1]$ (i.e., the vehicle always drives straight with a velocity $\dot{x} = 1, \dot{y} = 0$ in its own frame of reference, and the only control is choosing the bounded turning radius). The finite canonical control set contains the two vertices of the control set and the one point on the single edge where $\dot{\theta} = 0$:

$$U_c = \{(1, 0, -1), (1, 0, 1), (1, 0, 0)\} \tag{5.1}$$

Intuitively, the necessity of including the vertices is easily demonstrated by recalling that the Pontryagin Principle requires that, at an arbitrary time t, only controls that maximize the projection of the vehicle's velocity onto the adjoint vector may be chosen. If the adjoint happens to be, in the vehicle's frame of reference, at position a_1 depicted in Figure 5.2, then only the upper vertex $u_+ = (1, 0, 1)$ can be maximizing.

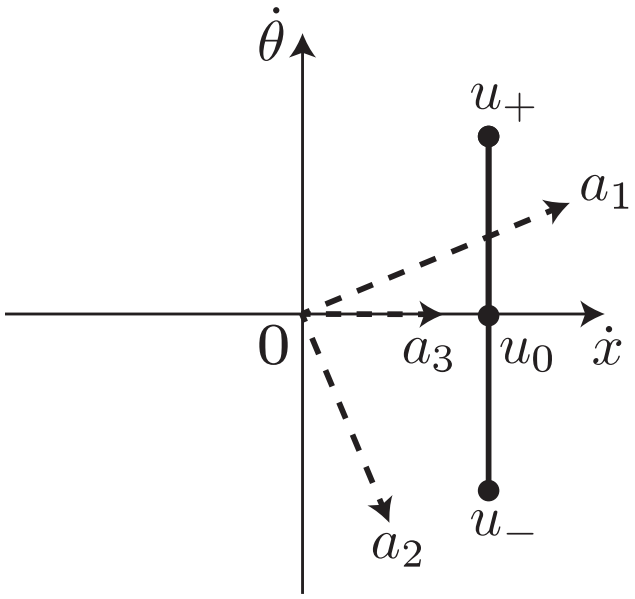


Figure 5.2: The control space for a Dubins car is a vertical line segment in the $(\dot{x}, \dot{y}, \dot{\theta})$ space. Extremal trajectories for this vehicle will use, at most times, either control u_+ (a hard turn left, e.g. for the adjoint at position a_1) or control u_- (a hard turn right, e.g. for the adjoint at position a_2). In the particular case when the adjoint a_3 is perpendicular to one edge of the control space, all the controls on this edge can be chosen; however, choosing, in this case, any control except the translation u_0 will cause the adjoint to move from this position, thus causing one of the two corners of U to become the single maximizing control.

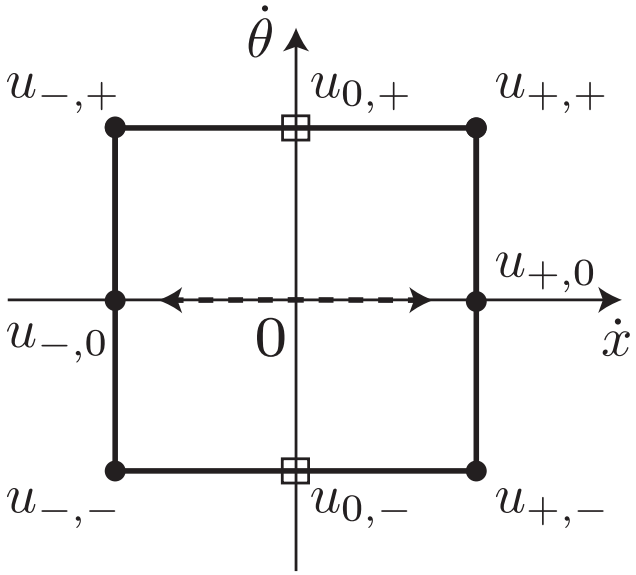


Figure 5.3: The control space for a convexified Reeds-Shepp car is a square centered on the origin in the $\dot{y} = 0$ plane. All the four corners maximize H for some position of the adjoint. Furthermore, the left and right edges contain points on both sides of the $\dot{\theta} = 0$ plane, a fact that allows singular translations. The top and bottom edges do not have this property.

If the adjoint is at position a_2, then only the lower vertex u_- is maximizing.

The possibility that other controls might also be used on an optimal trajectory is indicated by considering the adjoint in position a_3, i.e. perpendicular onto the edge of the control space. In such a case, all the controls on the edge are maximizing, and may be chosen without violating the Pontryagin Principle. However, it turns out that it is sufficient to add to the canonical control set the point on the edge where $\dot{\theta} = 0$, i.e. the translation $u_0 = (1, 0, 0)$.

The necessity of having this point in the control set that is sufficient for optimality is easily demonstrated. Consider the very simple situation where the Dubins car needs to drive from $q_0 = (-1, 0, 0)$ to $q_g = (0, 0, 0)$. If only u_+ and u_- were available, then there would exist no optimal control policy that is piecewise constant. Instead, the solution would be to "chatter", i.e. to alternate infinitesimal applications of u_+ and u_- respectively, achieving a continuous straight translation from this alternation of rotations. However, if we also add u_0 to the canonical control set, it is always possible to replace such "chattering" segments with a single application of u_0 instead. Deeper analysis will show that no other additions are necessary in this case.

Figure 5.3 shows the discretization of the control set for another well-studied vehicle, the convexified Reeds-Shepp car. The control set is, in this case, a square centered on the origin $U = [-1, 1] \times 0 \times [-1, 1]$. Note that such a convexified control set gives our car capabilities (e.g. spinning in place $u_{0+} = (0, 0, 1)$) that are not usually associated with car-like vehicles, a reason for which the original Reeds-Shepp car [19] only allowed controls belonging to a subset of this square.

However, it turns out that the convexified Reeds-Shepp car always has an optimal control policy that only uses six controls: straight forward, forward left, forward right, straight backwards, backwards left, backwards right. Since all of these controls are also available to the non-convexified Reeds-Shepp car, it turns out that the canonical optimal trajectories are, in fact, identical for these two apparently distinct

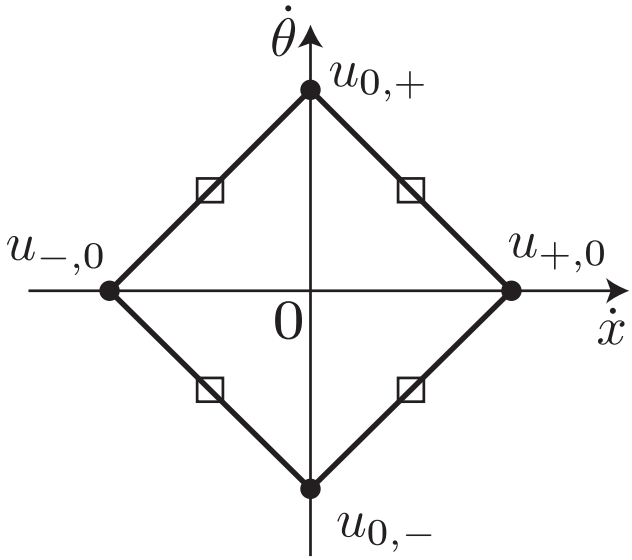


Figure 5.4: The control space for a differential drive vehicle is a diamond centered on the origin in the $\dot{y} = 0$ plane. Since none of the edges cross the $\dot{\theta} = 0$ plane, only the corners can be maximizing H in a sustainable manner, when the adjoint varies.

vehicles. (This discretization result was first presented in [28], which also first introduced the notion of the convexified Reeds-Shepp car.)

In our framework, we obtain the canonical control set for the convexified Reeds-Shepp car by adding to it the original control set's four corners and, by arguments analogous with those presented above for the Dubins car, also one point on each of the right and left edges, corresponding to translating straight forward ($u_{+,0} = (1, 0, 0)$) and backwards ($u_{-,0} = (-1, 0, 0)$) respectively. Note that the control set has two more edges, an upper and a lower edge. However, none of these edges intersects the $\dot{\theta} = 0$ plane (i.e., there are no translations on these edges), so the canonical control set does not contain points from these two edges.

Figure 5.4 shows the discretization of the control set of the differential drive, a wheelchair-like vehicle that can be driven by setting different velocities on two parallel wheels. In this case, none of the four edges crosses the $\dot{\theta} = 0$ plane, so the canonical control set contains simply the four vertices of the control space (two of which are translations).

Finally, Figure 5.5 shows a quadrilateral face of the control set for a hypothetical omnidirectional vehicle, projected onto the $(\dot{x}, \dot{\theta})$ plane. In a manner similar to the convexified Reeds-Shepp car, the canonical control set will contain, corresponding to this face of U, the four vertices and the two points where the edges intersect the $\dot{\theta} = 0$ plane. In addition, the control set will also contain a translation u' that corresponds to the face itself (as opposed to the edges). Intuitively, this translation corresponds to a "chattering" of three or more of the face's vertices.

Our analysis is also applicable to some control spaces that are not polyhedral. For example, if presented with an arbitrary, non-convex control set (e.g, a finite control set or the original control set of the Reeds-Shepp car), we would first build the convex hull of this control set, and then discretize the convex hull in order to obtain a canonical control set. If the canonical set is a subset of the original control set, then optimal control laws that are piecewise constant always exist, and can be obtained through the same methods that we use in the rest of our analysis. If the canonical control set is not a subset of the original control set, then

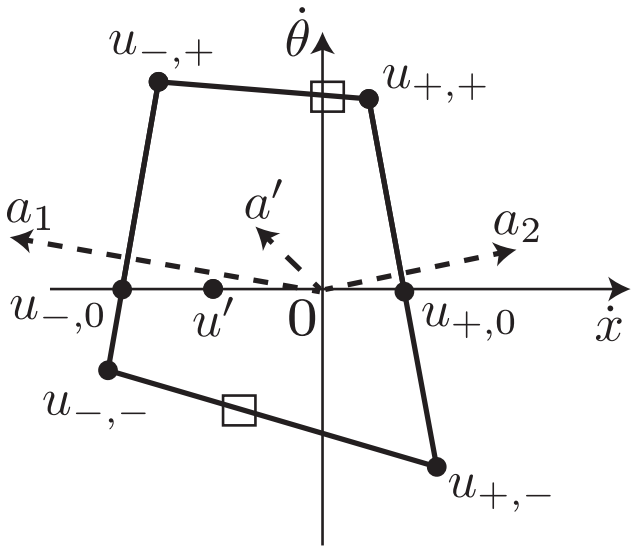


Figure 5.5: A projection onto the $\dot{y} = 0$ plane of a quadrilateral face of an arbitrary polyhedral control space. The corners can always be maximizing. The upper and lower edges do not cross the $\dot{\theta} = 0$ plane, and therefore contain no extra maximizing controls. When the adjoint a' is perpendicular onto the polyhedron's face, there exists at most one sustainable control u' on the face whose application will keep the adjoint in position a_3.

piecewise constant optimal control laws may not exist for that vehicle.

Let us now detail our analysis by presenting the mathematics behind the statements above.

5.2 Sufficiency of piecewise constant control policies that take values in the canonical control set

This section will present a proof for Theorem 2. The proof will be built up through a sequence of lemmas, some of which also contain results that will be used in subsequent chapters. The main outline of the proof is the following. First, we characterize the maximizing controls that need to be used by extremal control policies, and we will prove a few properties that such maximizing controls exhibit. Next, we prove that extremal control policies need to be piecewise constant outside so-called *singular intervals*. Singular intervals are time intervals during which multiple maximizing controls exist everywhere. Finally, we show that singular intervals can, in fact, be equivalently replaced with piecewise constant control policies. This will complete our proof that piecewise constant control policies are sufficient for optimality.

5.2.1 Maximizing controls

Let us first examine the kinds of controls that maximize the Hamiltonian during extremal trajectories. For an extremal trajectory $q(t)$ and its ensemble $u(t), \lambda(t), H$, let the times t where there exists a single control that maximizes the Hamiltonian be called *unimax*. Let the times where there exist multiple maximizing controls *multimax*. In this section, we will prove a sequence of lemmas that lead to the proof, in the next section, that piecewise constant control laws are sufficient for optimality.

Consider an arbitrary time $t' \in [0, t_f]$. The Pontryagin Principle requires that $u(t')$ be a solution of

the following problem: maximize the Hamiltonian $H = (\lambda(t')R_{\theta(t')})u$ for u in U. Since U is assumed to be a convex polyhedron, the condition $u \in U$ can be written as a linear system of inequalities, and the Hamiltonian maximization problem is thus a classical linear program: out of all the points in a convex polyhedron, find those points that maximize their position vectors' projections onto a given direction.

This simple observation, in conjunction with the facts about the adjoint proven in Theorem 1, allows us to make a few further observations that hold at all times along an optimal trajectory, no matter what value the adjoint may take.

Fact 1 *The optimal control policy can only contain controls that are on the outer hull of convex polyhedron U.*

Proof: The proof is by replacing any other control by its positive scaling to the convex hull to obtain a larger value for the Hamiltonian. ■

The next two observations follow immediately from the properties of linear programs and the fact that $\lambda(t)$ is a continuous function:

Fact 2 *If t is unimax, then $u(t)$ must be one of the vertices of U. Furthermore, there exists an entire neighborhood of t on which $u(t)$ is also the only maximizing control.*

Fact 3 *If t is multimax, then there exists an entire edge or face of U such that all the controls on this edge or face are maximizing.*

We are now ready to prove several lemmas about multimax points. The case where the multimax point involves an edge of U that is entirely made up of translations needs to be treated separately.

Lemma 5 *On an extremal trajectory with Hamiltonian H, there exists only one value of θ at which all the controls on an edge of U that is entirely in the $u_\theta = 0$ plane can be maximizing. Furthermore, there exists only one value of H for which this may happen.*

Proof: Let u_i and u_j be the two distinct vertices of the edge of U under consideration. Note that u_i and u_j are both translations. At the multimax point, we write equation 4.13 for both controls:

$$H = v_i \cos(\theta_L + \alpha_i) \tag{5.2}$$

$$H = v_j \cos(\theta_L + \alpha_j). \tag{5.3}$$

We have two equations in two unknowns, H and θ_L. Their right-hand sides are also equal:

$$v_i \cos(\theta_L + \alpha_i) = v_j \cos(\theta_L + \alpha_j). \tag{5.4}$$

Use of the cosine addition identity and some algebraic manipulation leads to:

$$\tan\theta_L = \frac{v_j \cos\alpha_j - v_i \cos\alpha_i}{v_j \sin\alpha_j - v_i \sin\alpha_i}. \tag{5.5}$$

We choose the solution for θ_L satisfying the requirement that $H > 0$, and compute the unique value for H from $H = v_i \cos(\theta_L + \alpha_i)$. ■

For edges of U that do not exclusively contain translations, we obtain

Lemma 6 *On a control line trajectory of Hamiltonian H, there exist only two pairs of values (y_L, θ_L) at which all the controls on an edge of U that is not entirely in the $u_\theta = 0$ plane can be maximizing.*

Proof: Let the two distinct vertices on the edge of U under consideration be labelled as u_1, u_2. Note that not both of these can be translations.

At the multimax point, we write equation 4.13 for both controls:

$$H = v_1 \cos(\theta_L + \alpha_1) + y_L \omega_1 \tag{5.6}$$
$$H = v_2 \cos(\theta_L + \alpha_2) + y_L \omega_2. \tag{5.7}$$

By multiplying the top equation by ω_2, the bottom by ω_1, and subtracting, we may eliminate y:

$$\omega_2 H - \omega_1 H = \omega_2 v_1 \cos(\theta_L + \alpha_1) - \omega_1 v_2 \cos(\theta_L + \alpha_2). \tag{5.8}$$

Using the cosine addition formula and rearranging, this equation can be written in the form

$$a \cos\theta_L + b \sin\theta_L = c, \tag{5.9}$$

where

$$a = \omega_2 v_1 \cos\alpha_1 - \omega_1 v_2 \cos\alpha_2 \tag{5.10}$$
$$b = -\omega_2 v_1 \sin\alpha_1 + \omega_1 v_2 \sin\alpha_2 \tag{5.11}$$
$$c = \omega_2 H - \omega_1 H. \tag{5.12}$$

Equation 5.9 is of standard form. Craig [10] gives the solution as

$$\theta_L = 2 \operatorname{atan}\left(\frac{b \pm \sqrt{a^2 + b^2 - c^2}}{a + c}\right). \tag{5.13}$$

(A solution also exists for $a + c = 0$: $\theta_L = \pi$.) Note that there are at most two solutions for θ_L. For each solution for θ_L, we can plug the values of θ_L, H, and the rotation control into equation 4.13 to compute the unique value of y. ■

5.2.2 Piecewise continuity outside singular segments

We are now in a position to define singular segments, and to show that extremal trajectories are piecewise continuous outside the singular segments. The next lemma will give a lower bound on the duration of unimax intervals.

Lemma 7 *For any extremal trajectory $q(t) : [0, t_d] \to \mathbb{SE}(2)$ with extremal ensemble $u(t), \lambda(t), H$, there exists a constant quantity $k_c > 0$ such that all unimax times are contained within unimax time intervals, each interval of duration at least k_c.*

Proof: Consider a unimax time t_v. Because of fact 2, there must exist $\varepsilon > 0$ such that all the points in $(t_v - \varepsilon, t_v + \varepsilon)$ are unimax. There must also exist a vertex u_v of U such that for all $t \in (t_v - \varepsilon, t_v + \varepsilon)$, $u(t) = u_v$.

There are two cases: either u_v is a rotation, or u_v is a translation.

If u_v is a rotation, then $\theta(t)$ is monotonic on $(t_v - \varepsilon, t_v + \varepsilon)$ and therefore $\theta(t)$ is not constant on any superset of this interval. Let $\Theta_c = \{\theta(0), \theta(t_d)\} \cup \{\theta |$ there exists an edge or face of U that can be maximizing at $\theta\}$. According to lemmas 6 and 5, Θ_c is a finite set.

Let $t_{c1} = \max\{t < t_v | \theta(t) \in \Theta_c\}$ and $t_{c2} = \min\{t > t_v | \theta(t) \in \Theta_c\}$. Then all the points in (t_{c1}, t_{c2}) are unimax, and furthermore $u(t) = u_v$ on this interval. Since $\dot{\theta}(t)$ thus exists and is constant on (t_{c1}, t_{c2}), there must exist a quantity $k_r > 0$, depending only on the trajectory and its extremal ensemble, such that the length of the (t_{c1}, t_{c2}) interval is at least k_r. This concludes the case where u_v is a rotation.

If u_v is a translation, then $y(t)$ is monotonic on $(t_v - \varepsilon, t_v + \varepsilon)$. Let $Y_c = \{y(0), y(t_d)\} \cup \{y |$ there exists an edge or face of U, not entirely in the $u_\theta = 0$ plane, that can be maximizing at $y\}$. (Since t_v is not multimax, and u_v does not modify θ, we do not need to consider the case when an edge of U entirely in the $\dot{\theta} = 0$ plane might become maximizing.) By an argument analogous to that used in the case where u_v was a rotation, there exists a quantity $k_t > 0$, depending only on the trajectory and its extremal ensemble, such that the length of the (t_{c1}, t_{c2}) interval is at least k_t.

The proof is concluded by letting k_c be the minimum of k_r and k_t. ■

We are now able to prove the central result of this section:

Lemma 8 *For any extremal trajectory $q(t) : [0, t_d] \to \mathbb{SE}(2)$ with extremal ensemble $u(t), \lambda(t), H$, the function*

$$u_s(t) = \begin{cases} u(t), & \text{if } t \text{ is unimax} \\ 0, & \text{if } t \text{ is multimax} \end{cases} \tag{5.14}$$

is piecewise constant. Furthermore, $u_s(t) = 0$ at all discontinuity points.

Proof: Any discontinuity point will contain both unimax and multimax points arbitrarily close to it; because of fact 2, such discontinuity points can only be multimax, and therefore u_s is 0 at all discontinuity points.

We will prove by contradiction that u_s is piecewise continuous. Assume that u_s is not piecewise continuous; then it must have an infinity of discontinuity points on $[0, t_d]$. Therefore there must exist a point t_a within $[0, t_d]$ such that there are discontinuity points arbitrarily close to t_a. However, according to Lemma 7, there exists k_c such that no unimax point can exist within $(t_a - k_c, t_a + k_c)$. Therefore, there exists an interval containing t_a in which all the points are multimax, and therefore u_s is 0, and thus continuous, on this interval; which is a contradiction.

Therefore, u_s is piecewise continuous. Since all the unimax points form intervals of a certain minimum length (Lemma 7), this also implies that u_s is piecewise constant. ■

The lemma above implies that $(0, t_d)$ is composed of open unimax intervals that are separated by either isolated multimax points or by closed intervals that are multimax at all points. Let a unimax interval be called a *generic interval*. The isolated multimax points separate generic segments, indicating a change of controls; we will therefore call isolated multimax points *control switches*. The nondegenerate multimax intervals can contain irregular behaviors for the control policy (e.g. nowhere continuous functions etc.); let such nondegenerate multimax intervals be called *singular intervals*.

5.2.3 Replacing singular segments

We will next be concerned with a more detailed characterization of singular intervals. We will show that singular intervals only occur at certain values of the Hamiltonian, and that singular intervals can always be

replaced by a time-equivalent piecewise constant control law, with at most two pieces. This will conclude our proof of Theorem 2.

Fact 4 *Consider an arbitrary extremal control policy $u(t)$ that contains a singular interval $[t_1, t_2]$ on which a face of U, which contains an edge e_t in the plane $\dot{\theta} = 0$, is maximizing everywhere. Then $u(t)$takes values from e_t almost everywhere.*

Proof: From Lemma 5, it is not possible for two distinct edges of U in the $\dot{\theta} = 0$ plane to be simultaneously maximizing. Therefore, all the other edges on the maximizing face need to be on the same side of the $\dot{\theta} = 0$ plane. Choosing $u_\theta(t) \neq 0$ at points that form a set of measure greater than zero would therefore result in $\theta(t_2) \neq \theta(t_1)$, which, in conjunction with Lemma 5, contradicts the initial assumption that e_t is maximizing everywhere. ■

Lemma 9 *Any singular interval $[t_1, t_2]$ for which the set $\{t \in [t_1, t_2] | u_\theta(t) \neq 0\}$ has measure zero, and during which multiple translation controls are maximizing, can be equivalently replaced by a control policy of the form*

$$u'(t) = \begin{cases} u_1, & t \in [t_1, t'] \\ u_2, & t \in [t', t_2] \end{cases} \tag{5.15}$$

for some $t' \in [t_1, t_2]$, where u_1 and u_2 are vertices of U that are translation controls.

Proof: The fact that $u_\theta(t) = 0$ almost everywhere, coupled with the requirement that the interval be singular and that multiple translation controls must be maximizing, indicates that there exists an edge e_t of U, contained in the $\dot{\theta} = 0$ plane, that is maximizing at all points on $[t_1, t_2]$.

The value of the Hamiltonian is determined in this case, as shown by Lemma 5; also, at no point in $[t_1, t_2]$ can there exist any other translations-only edge of U that is also maximizing.

Let u_1 and u_2 be the two corners of e_t. As a first step, we replace the value of $u(t)$ at all points where $u_\theta \neq 0$ with u_1. Since all these points form a set of measure zero, the shape of the trajectory $q(t)$ remains unchanged.

The requirement that all the controls on e_t be maximizing indicates, by application of equation 4.13, that for any $u \in e_t$ the $\dot{x}_L$ velocity is the same. Therefore, any control policy on $[t_1, t_2]$ that only uses controls from e_t achieves that same x_L displacement as the original $u(t)$.

Furthermore, any control policy that only uses controls from e_t will attain a y_L displacement that is intermediate between those achieved by $u_1(t) = u_1$ and $u_2(t) = u_2$ everywhere on $[t_1, t_2]$. Therefore, the following equation always has a solution in $[t_1, t_2]$:

$$u_{yL1} t' + u_{yL2}(t_2 - t_1 - t') = y_L(t_2) - y_L(t_1). \tag{5.16}$$

Thus, the control policy $u'(t)$ that is constructed by using the value of t' obtained above in 5.15 is an equivalent substitute for the original control policy $u(t)$.

■

We can thus lay aside the case of singular intervals that feature a maximizing translations edge e_t. There are two cases left to consider: intervals where an entire face of U, with no translations-only edge, is maximizing; and intervals where only one edge of U, which does not contain only translations, is maximizing. The following two lemmas address these two cases.

Lemma 10 *Any singular interval $[t_1, t_2]$ for which exactly one edge of U, e_r, which contains points on both sides of the $\dot{\theta} = 0$ plane, is maximizing can be equivalently replaced by a control policy of the form*

$$u'(t) = u_0, t \in [t_1, t_2], \tag{5.17}$$

where u_0 is the translation control in e_r. Furthermore, the value of the Hamiltonian is uniquely determined by e_r in this case.

Proof: The requirement that e_r must be maximizing everywhere on $[t_1, t_2]$, along with Lemma 6, indicate that y_L and θ are constant on $[t_1, t_2]$ and thus the trajectory is a translation parallel to the control line on this interval. The replacement control policy 5.17 thus leaves the trajectory $q(t)$ unchanged. From equation 4.13, the value of the Hamiltonian is thus calculated as the Euclidean length of the u_0 vector. ■

Lemma 11 *Any singular interval $[t_1, t_2]$ for which an entire face of U, F, which contains points on both sides of the $\dot{\theta} = 0$ plane, is maximizing can be equivalently replaced by a control policy of the form*

$$u'(t) = u_0, t \in [t_1, t_2], \tag{5.18}$$

where u_0 is a translation control in F. Furthermore, the value of the Hamiltonian is uniquely determined by F in this case.

Proof: The requirement that any given edge on F must be maximizing everywhere on $[t_1, t_2]$, along with Lemma 6, indicate that y_L and θ are constant on $[t_1, t_2]$ and thus the trajectory is a translation parallel to the control line on this interval.

Let u_i, u_j and u_k be three distinct corners of F that are not all on the same side of the $\dot{\theta} = 0$ plane. Since F is convex, these three vertices are not collinear.

For any control $u = (u_x, u_y, y_\theta)$, we can re-write equation 4.13 as

$$u_x c - u_y s + u_\theta y_L - H = 0, \tag{5.19}$$

where $c = \cos\theta_L$ and $s = \sin\theta_L$. By writing equations of the form 5.19 for u_i, u_j and u_k respectively, we construct the linear system

$$\begin{bmatrix} u_{xi} & -u_{yi} & u_{\theta i} & -1 \\ u_{xj} & -u_{yj} & u_{\theta j} & -1 \\ u_{xk} & -u_{yk} & u_{\theta k} & -1 \end{bmatrix} \begin{pmatrix} c \\ s \\ y_L \\ H \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5.20}$$

Since u_i, u_j and u_k are not collinear, the matrix is of rank three. By elementary row operations on the system, we eliminate variables H and y_L and are left with an equation of the form $ac + bs = 0$, where a and b are constants, not both zero. In conjunction with $c^2 + s^2 = 1$, we calculate two possible values of the (c, s) pair; these two values add up to zero. Replacing into the original system, we obtain two critical values of H, one of which is the negative of the other. Only the positive solution for H and its corresponding (c, s) pair are valid.

This value of H is the critical value corresponding to F. Once c, s and H are thus determined, u_0 is the only translation on F for which equation 5.19 holds:

$$u_{x0} c - u_{y0} s = H \tag{5.21}$$

■

This concludes our proof of Theorem 2. Based on the lemmas stated above, we can also summarize the following useful fact:

Fact 5 *There exists a finite set of singular values of the Hamiltonian, computed in the proofs of lemmas 5, 10 and 11, such that singular segments do not occur for extremal ensembles that do not contain a singular value of the Hamiltonian.*

Chapter 6

Generating canonical control policies

We have shown, in the previous chapter, that piecewise constant control policies, using controls from a finite, canonical control set, are sufficient for optimality, as far as control line trajectories are concerned. Control policies of this type are fully specified by a sequence of the type

$$[(u_1, t_1), (u_2, t_2), \ldots, (u_n, t_n)], \tag{6.1}$$

where the u_i are canonical controls and the t_i are the times for which each control is applied.

This result, in conjunction with our analysis of the kinematics of control sequences in Section 3.2.3, opens up the possibility of algorithmic search for the fastest trajectory. A direct search, however, will be very slow. Let the number of canonical controls be $|U_c|$; there are $|U_c|^n$ possible structures and n degrees of freedom for trajectories of the type described in Formula 6.1, and we have no upper bound on n.

The characterization of a control line in Section 4.1.2 offers the potential of considerably speeding up the search. There are only two degrees of freedom involved in specifying the position of a line and, once this line is fixed, many if not most of the control switches and control application times involved in Formula 6.1 are strictly determined.

The purpose of this chapter is to explore the use of the control line for generating canonical control policies. The first part of the chapter will delineate the conditions under which the position of the control line uniquely determines an extremal trajectory: we show that it does uniquely determine the trajectory in most cases. In the second part of the chapter, we develop a precise algorithm that builds canonical trajectories from the position of the control line.

6.1 Trajectories uniquely determined by the control line

For each trajectory segment on a canonical trajectory of type 6.1, there are two basic issues that any trajectory generation algorithm needs to address: what is the control to be applied, and for what duration is it to be applied? We will show that, for extremal trajectory ensembles that contain *generic*, i.e. non-singular, values of the Hamiltonian, the answers to both these questions are uniquely determined by the position of the control line.

Regarding the choice of controls, we will introduce the notion of *sustainable controls*. Even though an entire edge of U is maximizing at multimax points, we will show that it is usually the case that only one control can be applied sustainably, i.e. without contradicting the Pontryagin Principle after an infinitesimal application.

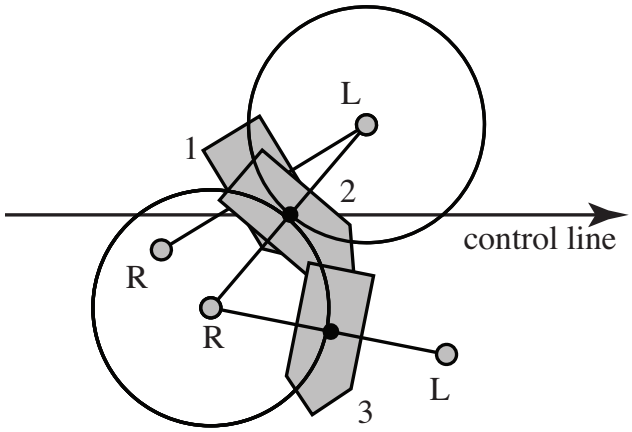


Figure 6.1: The control line uniquely determines a section of an optimal trajectory for a Dubins car.

For determining the duration of unimax segments, we will introduce the notion of *switching points*. We will show that every pair of canonical controls has a corresponding switching point in the vehicle's frame, such that the two controls' Hamiltonians are only equal when their corresponding switching point is on the control line. This analysis framework will be used twice: first, to show that the durations of control application are uniquely determined on generic trajectories; and, in the second part of the chapter, to calculate these durations.

6.1.1 Switching points

For any two controls of different angular velocities there will exist a point that has the same planar velocity (but different angular velocities) under both controls. For instance, consider controls $u_1 = (0, 0, -1)$ (a spin in place counter-clockwise) and $u_2 = (1, 0, 0)$ (forwards translation). The point $s_{12} = (0, 1, 1)$ has the same planar velocity $\dot{x} = 1, \dot{y} = 0$ under both controls (but different angular velocities).

As shown in Lemma 3, the Hamiltonian can be calculated at any point on the vehicle. Calculating the Hamiltonians of u_1, u_2 in a reference frame centered on s_{12}, by applying equation 4.12 we obtain

$$H_1 = \dot{x}_{1L} + y_L \dot{\theta}_1 \tag{6.2}$$

$$H_2 = \dot{x}_{2L} + y_L \dot{\theta}_2. \tag{6.3}$$

Since the planar velocity of s_{12} is equal under both controls, $\dot{x}_{1L} = \dot{x}_{2L}$. Therefore, the above two formulas only differ in $\dot{\theta}$:

$$H_2 - H_1 = y_L(\dot{\theta}_2 - \dot{\theta}_1). \tag{6.4}$$

Since $\dot{\theta}_2 - \dot{\theta}_1 > 0$ in this example, $H_2 \geq H_1$ iff $y_L \geq 0$. Therefore, for any extremal ensemble, the control policy can only take value u_2 when s_{12} is above the control line. u_1 can only be chosen when s_{12} is below the control line, and the control can only switch between u_1 and u_2 when s_{12} is on the control line. Since s_{12} thus indicates switches between u_1 and u_2, we call it a *switching point* between u_1 and u_2.

Figure 6.1 shows a switching point for the Dubins car. The car can turn left $u_L = (1, 0, 1)$ and right $u_R = (1, 0, -1)$; the switching point is $s_{RL} = (0, 0, 2)$ which is the midpoint between the two rotation centers. In the initial configuration, s_{RL} is above the control line, and thus u_L has to be chosen, as there

are no other available control and the Hamiltonian of u_R is smaller. After u_L is applied for some time, s_{RL} crosses the control line. At this point, both controls are maximizing the Hamiltonian.

However, any application of u_L will cause s_{RL} to immediately cross to below the control line, thus causing the Hamiltonian of u_R to be larger than that of u_L. Thus, only u_R is a *sustainable control* at the multimax point. The position of the control line thus uniquely determines the trajectory everywhere, including at the multimax point.

Let us now give a precise mathematical formulation of the intuitions above. We will first establish the existence of switching points, and then use this result to show unique determination of generic segment length. In the next section, we will examine sustainable controls.

Lemma 12 *Consider an extremal trajectory and its ensemble. For any pair of distinct controls u_i, u_j with $\dot{\theta}_i \leq \dot{\theta}_j$, there exists a switching point in the vehicle frame at homogeneous coordinates*

$$s_{ij} = \begin{pmatrix} \dot{y}_i - \dot{y}_j \\ \dot{x}_j - \dot{x}_i \\ \dot{\theta}_j - \dot{\theta}_i \end{pmatrix}, \tag{6.5}$$

such that

1. *The Hamiltonians of the two controls, H_i and H_j, are equal iff the world frame position of s_{ij} is on the control line.*
2. *$H_i < H_j$ iff s_{ij} is above the control line, i.e. iff the y_L coordinate of s_{ij} is strictly positive.*

Proof:
In the inequality

$$H_i \leq H_j, \tag{6.6}$$

we develop both sides according to Formula 4.10 and subtract the left-hand side from the right-had side. The inequality thus obtained is identical to the one we obtain by developing the formula

$$0 \leq (0,1,0) T_{LW} T_{WR} s_{ij}, \tag{6.7}$$

where T_{LW} and T_{WR} are the world-frame-to-control-line-frame and robot-frame-to-world-frame transformation matrices, respectively.

■

Fact 6 *Given two controls, u_i and u_j, let their rotation center representations (see eq. 3.11) be c_i and c_j. Then c_i, c_j and the switching point s_{ij} are collinear.*

Proof: From equations 6.5 and 3.11,

$$s_{ij} = c_j - c_i. \tag{6.8}$$

Let $L = (l_1, l_2, l_3)$ be a homogenous coordinates line through c_i and c_j. Then $Lc_i = Lc_j = 0$, and therefore $Ls_{ij} = 0$ and the switching point is collinear with the two rotation centers. ■

Fact 7 *Let e be an edge of U. Then all pairs of distinct controls on e have the switching point in the same location (with possibly different homogeneous weights).*

Proof: Let u_i and u_j be the two ends of e. It is sufficient to show that for any u_k on the edge, s_{ik} and s_{ij} are in the same location.

Since u_k is on the edge, there exists $a \in (0, 1)$ such that

$$u_k = au_j + (1 - a)u_i. \tag{6.9}$$

Multiplying by $R_{\pi/2}$,

$$c_k = ac_j + (1 - a)c_i. \tag{6.10}$$

Applying equation 6.8,

$$s_{ik} = c_k - c_i = a(c_j - c_i) = as_{ij}. \tag{6.11}$$

Therefore, s_{ij} and s_{ik} are homogenous multiples and thus represent the same location. ■

Fact 8 *Let s_e be the switching point for an edge e of U. Then the planar point represented by s_e point has the same $(\dot{x}, \dot{y})$ planar velocity under all controls on e.*

Proof: If all the controls on e have the same angular velocity, Formula 6.5 indicates s_e is at infinity. Any control on e imparts the same motion velocity on s_e.

If e is not parallel to the $\dot{\theta} = 0$ plane, then we calculate the planar velocity of s_e under each of the two ends of e by assembling the velocity matrix from Formula 3.17 and multiplying by s_e as given by equation 6.5. We obtain the same result; since s_e's planar velocity under all the other controls on e is intermediate between these two, it follows that the planar velocity of s_e is the same under all controls on e. ■

Lemma 13 *The duration of each generic interval is uniquely determined by the position of the control line.*

Proof: Consider an arbitrary generic interval (i.e. a unimax segment with possible isolated multimax points at its ends) with a fixed starting configuration and a maximizing control u. If the interval is the last generic interval in the trajectory, then the duration is limited by the final configuration.

If the generic interval under consideration is not the last, then consider the positions of all the switching points at the fixed starting configuration. When u is applied, for each switching point it is the case that it either never collides with the control line, or there exists a minimum positive time until it collides with the control line. Since the interval is not the last on the trajectory, there must be a control switch after it. The time at which the control switch occurs is one of the times at which one of the switching points collides with the control line. ■

The next section will show that control switches on generic trajectories are also uniquely determined by the position of the control line. Section 6.3 will show how to calculate the duration of generic intervals.

6.1.2 Sustainable controls

On piecewise constant trajectories, each applied control must be maximizing H not only instantaneously, but also for at least some short time interval. We formalize this fact in the concept of sustainable controls. Let a *sustainable control* at time t be a maximizing control u for which there exists $\varepsilon > 0$ such that the application of u on time interval $[t, t + \varepsilon)$ keeps u as a maximizing control.

For canonical trajectories, since the control set is finite, sustainability analysis can be done on a pairwise basis. A control u_i is sustainable if and only if it is sustainable with respect to any other control u_j: u_i must

be maximizing, and its immediate application must not cause the Hamiltonian of u_j, H_j, to become higher than H_i.

It is evident that only sustainable controls may be chosen by piecewise constant extremal control policies. If the control policy has in its ensemble a generic value of the Hamiltonian, i.e. not one of the singular values characterized in Section 5.2.3, then the control policy cannot contain any singular intervals. The control policy is therefore made up of unimax intervals separated by isolated multimax points.

On such generic trajectories, despite there being multiple maximizing controls at the multimax points, the choice is, in fact, deterministic at these points.

Lemma 14 *For any extremal trajectory, if its ensemble contains a generic value of the Hamiltonian then there exists only one sustainable control at all points.*

Proof: The lemma is evident for unimax points. We only need to consider multimax points in this proof. According to the analysis in Section 5.2.1, at a multimax point there may be multiple maximizing edges, or a single maximizing edge.

As shown in the proof of Lemma 11, if two or more distinct edges of U are simultaneously maximizing then the value of the Hamiltonian must be singular. This case is therefore ruled out.

Thus, only one edge e can be maximizing. If e contained multiple translations, then Lemma 5 would yield a singular value for the Hamiltonian. There are, therefore, only two possibilities: either e contains exactly one translation, or it contains none.

Case 1: If e contains no translation, then the edge doesn't cross the $\dot{\theta} = 0$ plane. e therefore contains only rotations that have the same sign on their angular velocities.

Consider a pair of rotations u_i and u_j that have the same sign on their angular velocities. Let s_{ij} be their corresponding switching point. As observed above, the two centers of rotation c_i and c_j are collinear with s_{ij}. At a time when both u_i and u_j are maximizing, s_{ij} must be on the control line. Also, on account of corollary 1, both rotation centers are on the same side of the control line. Given this geometry, there exists a half-plane bounded by the control line, so that, no matter whether u_i or u_j is applied, s_{ij} instantaneously moves into this half-plane. On account of Lemma 12, only one of the two controls u_i and u_j is associated with this half-plane; therefore, only one of the two rotations is sustainable.

By repeating this argument for all rotations on e, only one rotation on e can be sustainable in this case.

Case 2: If e does contain a translation, assume e contains multiple sustainable controls. According to the analysis in the previous case, no two of these sustainable controls can be on the same side of the $\dot{\theta} = 0$ plane. Therefore, there can be at most three sustainable controls: the two extremes of e, of opposite sign angular velocities, and the translation that is contained by e.

If a rotation and a translation are both sustainable, then the translation has to be parallel to the control line, in order to keep s_{ij} on the control line. As shown in Section 5.2.3, the Hamiltonian has a singular value in this case. If two rotations of opposite angular velocities are both sustainable, then the translation on e is also sustainable, which implies, as above, that H is singular. ■

6.1.3 Periodicity of generic trajectories

In this section, we show that generic trajectories are periodic by proving the following lemma:

Lemma 15 *There exists a partitioning of the values of the Hamiltonian into a finite set of open intervals and a finite set of critical values, such that every canonical trajectory with a Hamiltonian within a single interval, containing the same control switch, will contain the same sequence of control switches, following the switch that the trajectories have in common.*

Proof: The partitioning of the Hamiltonian is effected by a finite set of *critical values* of the Hamiltonian. To obtain the set of critical values, we start with the set of singular values (see fact 5) and add to them the magnitudes of the velocities of all switching points. Fact 7 indicates that there is only one switching point per edge of U, and fact 8 indicates that this switching point has the same $(\dot{x}, \dot{y})$ planar speed under all controls on its edge. Therefore, the number of velocity magnitudes to be added to the set of critical values is no larger than the number of edges of U.

Now we want to show that outside of these critical values, continuous perturbation of the value of H starting from a trajectory with some control switch u_1, u_2 does not change the structure of the trajectory.

Consider an extremal with a value of H not in the partitioning. Since values of H where a $\dot{\theta} = 0$ edge of U is maximizing are singular, this extremal may not contain any translation-translation switches. If the trajectory contains control switch u_1, u_2, then the (y_L, θ_L) configuration at this switch may be uniquely computed from Lemma 6 and from the analysis in Section 6.3.1. This (y_L, θ_L) configuration furthermore changes continuously with changes in H.

Starting from switch u_1, u_2, consider each possible switch u_2, u_k with $k \neq 2$. Each of these switches will happen when the corresponding switching point hits the control line. Continuously perturbing H, and thus the initial configuration, will not immediately change the order in which the switching points hit the control line. Changes in this order need to pass through configurations in which either:

1. Two or more switching points hit the control line simultaneously. In this case, at least two distinct edges of U are maximizing; thus an entire vface of U is maximizing, and the value of H is singular.

2. The trajectory of a switching point becomes tangent to the control line. At the tangency point, this switching point's velocity is parallel to the control line. Calculating the Hamiltonian at this switching point at the tangency moment, it is equal to its velocity, therefore H is critical.

Thus, as long as the change in H does not cross a critical value, the trajectory structure is constant. Since there are a finite number of possible control switches on generic trajectories (two for each edge of U), all generic trajectories beyond a certain number of switches must become periodic in their switch sequence. Since each switch corresponds to a fixed (y_L, θ_L) configuration, these trajectories are also periodic in (y_L, θ_L) space. ■

We will discuss the (y_L, θ_L) space at more length in Section 6.2. For now, let us prove a limitation on the number of periods for a certain class of extremal trajectories. We will need the following fact:

Fact 9 *Generic trajectory segments with constant $\theta(t)$ have constant controls.*

Proof: Since $\theta(t)$ is constant, the control must be a translation. On non-singular trajectories, there may only be one maximizing translation for any given θ. ■

Lemma 16 *Generic trajectories for which the image of $\theta(t)$ is not S^1, and for which $\dot{\theta}(0) \neq 0$, contain no more than one period.*

Proof: Consider a candidate trajectory that contains more than a full period. We will prove that this trajectory is not optimal by constructing another trajectory from the start to the goal that takes an equal amount of time, but does not satisfy the Maximum Principle (and is therefore not optimal). Figure 6.2 illustrates the idea.

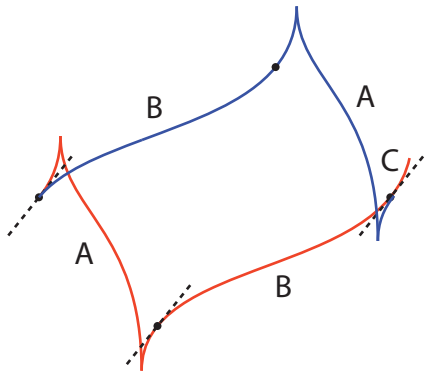


Figure 6.2: Construction showing that optimal trajectories for which the image of $\theta(t)$ is not S^1, and for which $\dot{\theta}(0) \neq 0$, contain no more than one period.

Any trajectory achieves both a minimum and a maximum value for θ. For now, assume that $\theta(0)$ is not either of the extreme values. By fact 9, trajectories for which the image of $\theta(t)$ is a point contain zero periods.

Let T be the duration of the first period. Since $\theta(t)$ is continuous, it must achieve all the values between the minimum and maximum values of θ in $(0, T)$. Therefore there exists $t_1 \in (0, T)$ such that $\theta(t_1) = \theta(0)$. Also, $y(0) \neq y(t_1)$, since $t_1 < T$. Let A be the section of the trajectory on the interval $[0, t_1]$, let B be the section of the trajectory on $[t_1, T]$, and let C be the remainder of the trajectory. The controls at the start of A and the start of C are the same.

Now construct the trajectory BAC. This trajectory takes the same duration as ABC, is feasible, and reaches the goal. On this new trajectory, we have the same controls at the beginning of A (time $T - t_1$) and beginning of C (time T), but different y values. If we compute the Hamiltonians at these times with equation 4.12,

$$H(T - t_1) = \dot{x}_L(T) + \dot{\theta}(T) y_L(T - t_1) \tag{6.12}$$

$$H(T) = \dot{x}_L(T) + \dot{\theta}(T) y_L(T), \tag{6.13}$$

we find that the Hamiltonian is not constant of the trajectory, since $y_L(T) \neq y_L(T - t_1)$ and $\dot{\theta}(t) \neq 0$. ■

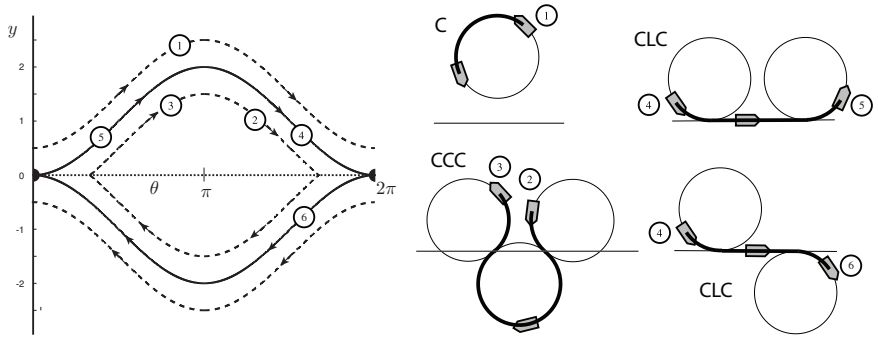
Let us now examine a way in which generic trajectories may be easily visualized.

6.2 Visualizing trajectory segments uniquely determined by the control line

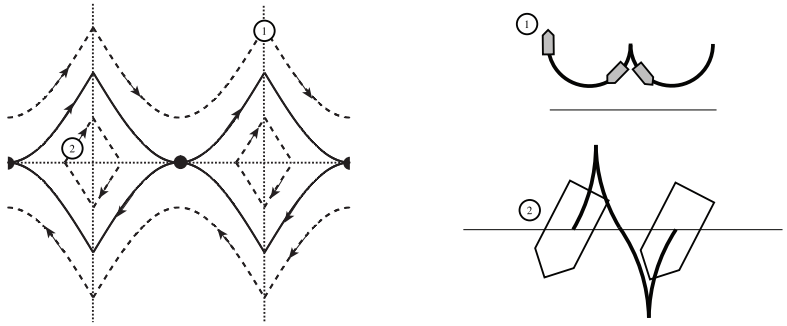
From a topological point of view, the requirement that the Hamiltonian be constant on extremal trajectories (Theorem 1) makes it possible to represent extremal trajectories as level curves of the Hamiltonian. In the control line frame of reference, the level curves of the Hamiltonian are two-dimensional, only depending on y_L and θ_L (eq. 4.12).

It is not particularly difficult to represent these two-dimensional level curves graphically. Figure 6.3 shows some examples. Specifically, assume control

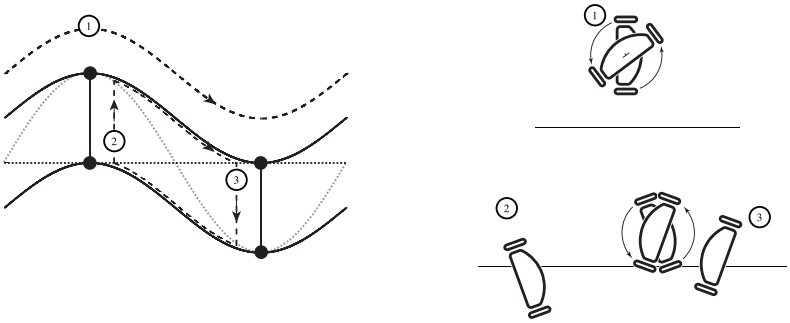
$$u_i = (v_i \cos \alpha_i, v_i \sin \alpha_i, \omega_i) \tag{6.14}$$



(a) The Dubins car, a car that can only drive forwards.



(b) The Reeds-Shepp car.



(c) The differential-drive, a wheelchair-like vehicle.

Figure 6.3: Switching spaces and example trajectories for standard robotic vehicles. For each vehicle, the figure on the left side shows level curves of the Hamiltonian in the (y_L, θ_L) space. The figure on the right side shows some extremal trajectories, in the plane, that correspond to portions of the Hamiltonian level curves on the left.

is maximizing for some (y_L, θ_L) configurations that are on the level curve for a specified H. Then, from equation 4.13,

$$H - v_i \cos(\theta_L + \alpha_i) = y_L \omega_i. \tag{6.15}$$

This is a sinusoid curve. Repeating this analysis for all the controls in the canonical control set U_c, we obtain the level curve of Hamiltonian value H as a union of (possibly degenerate) segments of sinusoids. The bounds of these segments are on (y_L, θ_L) configurations that are multimax, and thus can be calculated through the techniques of lemmas 6 and 5.

This representation of level curves makes it easy to visualize the periodicity of generic trajectories. For the Dubins car (Fig. 6.3 (a)), the periodicity is evident for both trajectories (1) (a turning motion that, as far as constrained by the control line, may continue indefinitely) and (2) - (3) (a sequence of left and right turns). The continuous level curve that contains configurations (4), (5) and (6) contains a singular point at the origin (showed twice since the (y_L, θ_L) space has a cylindrical topology). In the $(0, 0)$ configuration, corresponding to the car being on the control line and parallel to the control line, there is a choice of:

1. A multimax segment of indefinite length, driving straight along the control line.
2. A left turn.
3. A right turn.

The level curve passing through this configuration is thus singular. Similar analyses apply to the Reeds-Shepp car (Fig. 6.3 (b)) and the differential drive (Fig. 6.3 (c)). Each level curve contains a potentially infinite number of optimal trajectories, since it is possible to start and end anywhere on it, as long as travel proceeds continuously, with the possibility of periodicity as well. Some level curves (e.g. the one containing configuration (2) for the Reeds-Shepp car) may consist of multiple disjoint segments.

6.3 An algorithm that generates trajectories based on the control line position

The approach described in Section 6.2 is useful for visualizing extremal trajectories, but not particularly suitable for generating them. An algorithm based on this approach would need to determine connections between the ends of the various sinusoids, and perform graph searches in order to determine the sequence of switches and switching configurations. This section presents a simpler, more direct (if less visual) method to achieve the same goal.

A very simple algorithm for building a generic trajectory, given a position of the control line, would be to

1. Determine the sustainable control, by calculating H and $\frac{d}{dt}H$ for all controls in the canonical control set U_c.
2. Integrate the sustainable control for some very small time t_ε.
3. Repeat.

This algorithm has a number of problems, notably error accumulation and the danger of missing out on controls that are only applied for a very short time. We will spend the remainder of this section building a much more precise algorithm. Its basic outline is

1. Determine the sustainable control u, by calculating H and $\frac{d}{dt}H$ for all controls in the canonical control set U_c.

2. For each control u_i in U_c, determine the time to switch to u_i, i.e. the time that u can be applied before u_i becomes sustainable.

3. Pick the minimum of the switching times, t_{min}.

4. Integrate u for t_{min} to obtain a new state, using Equation 3.4.

5. Repeat.

Step 3 is trivial, and step 4 is achieved by the method described in Section 3.2.2. Step 1 will be described in the following section. Step 2 is the most challenging.

In the context of the characterization of switching points in Lemma 12 above, step 2 is an instance of a more general kinematics problem. Given a planar vehicle that is moving at velocity u, constant in its own frame of reference, and a point P rigidly attached to the vehicle (the switching point, in our case), how long will it take until point P collides with a given line in the plane? We solve this more general problem in Section 6.3.2.

6.3.1 Determining all sustainable controls

As defined above, sustainable controls need to maximize the Hamiltonian, and to change state in a manner that doesn't increase the Hamiltonian of other maximizing controls. We have already shown how to calculate a control's Hamiltonian.

Assume two canonical controls, u_i and u_j, are both maximizing at some multimax point t: $H_i = H_j = H$. From equation 4.13,

$$H_i(q_L(t)) = v_i \cos(\theta_L + \alpha_i) + y_L \omega_i \tag{6.16}$$

Assume $u_j = (v_j \cos\alpha_j, v_j \sin\alpha_j, \omega_j)$ is instantaneously applied. Then the rate of change of the expression above is

$$\frac{d}{dt}H_i(q_L) = -\omega_j v_i \sin(\theta_L + \alpha_i) + v_j \omega_i \sin(\theta_L + \alpha_j) \tag{6.17}$$

If u_j is sustainable, its application must not immediately cause H_i to become greater than H. Therefore,

$$\frac{d}{dt}H_i \leq 0 \tag{6.18}$$

In a given configuration, we repeat this procedure for all pairs of maximizing canonical controls and eliminate all non-sustainable controls. As shown in Lemma 14, if the value of H is generic then there will be only one sustainable control left, when this analysis is finalized.

6.3.2 Time to collision with a line

In the next section, we will see that the solution to the switching time problem involves three cases. The most difficult case is that in which the motion is a nondegenerate rotation. For solving this case, we will prove an auxiliary lemma and will introduce calculations for some quantities.

The following lemma gives us the time to collision with vertical line $x = c_2$ when moving on the unit circle (starting at angle c_1).

Lemma 17 *The minimum $t \geq 0$ such that*

$$\cos(\omega t + c_1) = c_2 \tag{6.19}$$

where $\omega \neq 0$ and $c_2 \in [-1, 1]$, is

$$t = \frac{2\pi}{|\omega|} \min\left(\left\{ -\frac{c_1 - \operatorname{acos} c_2}{2\pi \operatorname{sign} \omega} \right\}, \left\{ -\frac{c_1 + \operatorname{acos} c_2}{2\pi \operatorname{sign} \omega} \right\} \right) \tag{6.20}$$

where $\{x\} = x - \lfloor x \rfloor$ is the fractional part of x.

Proof: The problem is restated equivalently as finding the minimum of a set:

$$t_{min} = \min\{t \geq 0 | \cos(\omega t + c_1) = c_2\} \tag{6.21}$$

Defining

$$A_+ = \{t \geq 0 \mid \exists k \in \mathbb{Z}\ \omega t + c_1 = \operatorname{acos} c_2 + 2k\pi\} \tag{6.22}$$

and

$$A_- = \{t \geq 0 \mid \exists k \in \mathbb{Z}\ \omega t + c_1 = -\operatorname{acos} c_2 + 2k\pi\} \tag{6.23}$$

we have

$$\begin{aligned} t_{min} &= \min(A_+ \cup A_-) && (6.24)\\ &= \min(\min A_+, \min A_-) && (6.25) \end{aligned}$$

Next we find $\min A_+$. In the case where $\omega > 0$

$$\begin{aligned} \min A_+ &= \frac{1}{\omega} \min\{t' \geq 0 \mid t' = -c_1 + 2k\pi + \operatorname{acos} c_2\} && (6.26)\\ &= \frac{-c_1 + \operatorname{acos} c_2}{\omega} + && (6.27)\\ &\quad + \frac{2\pi}{\omega} \min\{k \mid -c_1 + 2k\pi + \operatorname{acos} c_2 \geq 0\} && (6.28)\\ &= \frac{-c_1 + \operatorname{acos} c_2}{\omega} + \frac{2\pi}{\omega} \lceil \frac{c_1 - \operatorname{acos} c_2}{2\pi} \rceil && (6.29) \end{aligned}$$

Since $\lceil x \rceil = -\lfloor -x \rfloor$,

$$\begin{aligned} \min A_+ &= \frac{-c_1 + \operatorname{acos} c_2}{\omega} - \frac{2\pi}{\omega} \lfloor \frac{-c_1 + \operatorname{acos} c_2}{2\pi} \rfloor && (6.30)\\ &= \frac{2\pi}{\omega} \left(\frac{-c_1 + \operatorname{acos} c_2}{2\pi} - \lfloor \frac{-c_1 + \operatorname{acos} c_2}{2\pi} \rfloor \right) && (6.31) \end{aligned}$$

Therefore

$$\min A_+ = \frac{2\pi}{\omega}\left\{\frac{-c_1 + \ \text{acos}\ c_2}{2\pi}\right\} \tag{6.32}$$

In the case where $\omega < 0$, we analogously obtain (by using the identity $\min S = -\max(-S)$)

$$\min A_+ = -\frac{2\pi}{\omega}\left\{\frac{c_1 - \ \text{acos}\ c_2}{2\pi}\right\} \tag{6.33}$$

Therefore

$$\min A_+ = \frac{2\pi}{|\omega|}\left\{-\frac{c_1 - \ \text{acos}\ c_2}{2\pi\ \text{sign}\ \omega}\right\} \tag{6.34}$$

Analogously,

$$\min A_- = \frac{2\pi}{|\omega|}\left\{-\frac{c_1 + \ \text{acos}\ c_2}{2\pi\ \text{sign}\ \omega}\right\} \tag{6.35}$$

Equation 6.20 follows immediately. ■

6.3.3 Time to switch

We are now ready to give a general solution to the time to switch problem. In the control line frame, let $P(0) = (x_P, y_P, w_P)$ be the homogeneous coordinates of a point of interest (i.e. the switching point), possibly at infinity, attached to the robot, at $t = 0$. Let $R = (x_R, y_R, \omega)$ be the homogeneous coordinates of the center of a (possibly degenerate) rotation, corresponding to control $\dot{x} = y_R$, $\dot{y} = -x_R$, $\dot{\theta} = \omega$. We would like to find the minimum $t \geq 0$ such that $P(t)$ is on the x axis.

We will use Lemma 17 to solve the nondegenerate rotation case ($\omega \neq 0$). We calculate a few additional quantities to reduce the rotation case problem to the situation in Lemma 17. The forward kinematics from eq. 3.14 imply

$$y_P(t) = b_1 \sin\omega t + b_2 \cos\omega t + b_3 \tag{6.36}$$

where

$$b_1 = x_P - w_P x_R/\omega \tag{6.37}$$

$$b_2 = y_P - w_P y_R/\omega \tag{6.38}$$

$$b_3 = w_P y_R/\omega \tag{6.39}$$

Let $r = \sqrt{b_1^2 + b_2^2}$ (since P and R are different points, $r > 0$) and let $\varphi = -\text{atan2}(b_1, b_2)$. Then P is on the x axis when $y_P(t) = 0$, i.e.

$$r\cos(\omega t + \varphi) = -b_3 \tag{6.40}$$

To reduce this case to the situation in Lemma 17, let

$$c_1 = \varphi \tag{6.41}$$

and

$$c_2 = -b_3/r. \tag{6.42}$$

The other cases are trivial. The general solution, divided into three cases with several subcases, can thus be detailed as follows:

1. Degenerate cases
(P and R correspond to the same planar point, or $\omega = w_P = 0$)
- a) $y_P = 0$: $t = 0$
- b) $y_P \neq 0$: $t = \infty$

2. Translation
($\omega = 0$, $w_P \neq 0$)
- a) $y_P = 0$: $t = 0$
- b) $x_R = 0$ or $y_P/x_R < 0$: $t = \infty$
- c) $y_P/x_R > 0$: $t = y_P/x_R$

3. Rotation
($\omega \neq 0$)
Obtain c_1, c_2 as in equations 6.41 and 6.42.
- a) $c_2 \notin [-1, 1]$: $t = \infty$
- b) $c_2 \in [-1, 1]$: Obtain t from Lemma 17.

This concludes our analysis of how to generate non-singular trajectory segments.

Chapter 7

Finding minimum time trajectories

The generator algorithm in Chapter 6 constructs a trajectory for a given position of the control line. In this chapter, we use this result to find the fastest control line trajectories that reach the goal. For singulars, we are able to find the exact location of the control line. For generic trajectories, we vary the position of the control line until the goal is attained with a sufficiently small error.

The position of the control line is determined by three parameters: k_1, k_2 and k_3. Without loss of generality, we can restrict k_1 and k_2 such that $k_1^2 + k_2^2 = 1$. While positioning a line is thus generally a problem with two degrees of freedom, working with a finite control set allows us to consider all possible pairs for the initial and final controls, for any given instance of the problem. Setting the condition that the Hamiltonian needs to be equal for the initial and final controls allows us to reduce the problem of finding the control line to one degree of freedom. This condition is equivalent to determining one point, possibly at infinity, on the control line.

7.1 Parametrization of the control line position by the value of the Hamiltonian

A parametrization by the Hamiltonian value is convenient for search. For singulars, we showed in Chapter 5 how to calculate all possible singular Hamiltonian values, and this parametrization yields the position of the control line directly, in most cases. For generics, we have shown in Section 6.1.3 how to calculate a number of critical values of the Hamiltonian, such that the trajectory structure (i.e. the sequence of controls used) is constant when the Hamiltonian varies between two critical values.

We describe a procedure to obtain a function $f(H)$ that returns a maximum of two (k_1, k_2, k_3) tuples characterizing the position of the control line or, equivalently, (φ, k_3) tuples, as $(k_1, k_2) = (\cos\varphi, \sin\varphi)$.

This function of H is curried from a function that involves several additional parameters. Given initial and final states, q_0 and q_f, with corresponding robot-frame-to-world frame transformation matrices T_0 and T_f, and initial and final controls, u_0 and u_f, let c_0 and c_f be the rotation center representations (see eq. 3.11) of u_0 and u_f, respectively. Let c_{W0} and c_{Wf} represent the initial and final rotation centers in the world frame:

$$c_{W0} = T_0 c_0 = (a_0, b_0, g_0) \tag{7.1}$$

$$c_{Wf} = T_f c_f = (a_f, b_f, g_f). \tag{7.2}$$

The Hamiltonians of the initial and final controls need to equal the trajectory's Hamiltonian value:

$$(-\sin\varphi, \cos\varphi, k_3)c_{W0} = H \tag{7.3}$$
$$(-\sin\varphi, \cos\varphi, k_3)c_{Wf} = H. \tag{7.4}$$

Subtracting,

$$(-\sin\varphi, \cos\varphi, k_3)(c_{W0} - c_{Wf}) = 0. \tag{7.5}$$

Let $(a, b, g) = c_{W0} - c_{Wf}$. There are two cases, according to whether g is zero.

Case 1: $g \neq 0$

We solve for k_3 from equation 7.5 and replace into system 7.3 to obtain

$$-a' \sin\varphi + b' \cos\varphi = H, \tag{7.6}$$

where

$$a' = a_f - a\frac{g_f}{g} \tag{7.7}$$
$$b' = b_f - b\frac{g_f}{g}. \tag{7.8}$$

If $a' = b' = 0$, there exists no solution and $f(H)$ identically returns the empty set. Otherwise, let r' and α' be such that

$$a' = r' \sin\alpha' \tag{7.9}$$
$$b' = r' \cos\alpha'. \tag{7.10}$$

Then

$$r' \cos(\varphi + \alpha') = H. \tag{7.11}$$

We solve this equation and replace into 7.5 to obtain k_3:

$$\varphi = -\alpha' \pm \operatorname{acos}\frac{H}{r'} \tag{7.12}$$
$$k_3 = \frac{a\sin\varphi - b\cos\varphi}{g}. \tag{7.13}$$

The function $f(H)$ returns, for any value of H, the two (φ, k_3) tuples corresponding to this system.

Case 2: $g = 0$

In this case, the initial and final controls have the same angular velocity. If $a = b = 0$, then c_{W0} and c_{Wf} coincide. We will treat this case in Section 7.3. Otherwise, let r and α be such that

$$(a, b) = (r\sin\alpha, r\cos\alpha). \tag{7.14}$$

By algebraic manipulation,

$$r\cos(\varphi+\alpha)=0, \tag{7.15}$$

and therefore

$$\varphi=-\alpha\pm\frac{\pi}{2}. \tag{7.16}$$

As above, we have

$$(-\sin\varphi,\cos\varphi,k_3)c_{Wf}=H. \tag{7.17}$$

We will treat the case where $g_f=g_0=0$ (u_0 and u_f are translations) in Section 7.3. Otherwise,

$$k_3=\frac{H+a_f\sin\varphi-b_f\cos\varphi}{g_f}. \tag{7.18}$$

In conclusion, we have obtained a parametrization, in the general case, of the position of the control line by the value of H. There also exist two cases where this parametrization does not exist: when the initial and final motions are both translations, and when the initial and final world frame velocities are identical. We will begin by studying how to find optimal trajectories in the case where a parametrization by H exists. The cases when the parametrization does not exist will be studied in the second half of the chapter.

7.2 Finding optimal control policies corresponding to control lines parametrized by H

As shown previously, control line trajectories can be divided into singulars and generics. Singulars have singular values of H that were exactly determined in Chapter 5. Generics have control policies that are uniquely determined by the position of the control line.

7.2.1 Computing singulars

Since, for singulars, the value of H belongs to a finite set of singular values, we use the control line parametrization developed above to determine the exact position of the control line in this case.

Generic excursions

Singulars are not composed entirely of singular points. On non-singular segments, the control switches and durations are uniquely determined by the position of the control line. We call sub-trajectories that contain no singular point (i.e., no point with multiple sustainable controls) *generic excursions*. Since the position of the control line is known, generic excursions are constructed by the Chapter 6 generator. Generic excursions are of three types:

1. Type A: The initial generic excursion on a singular trajectory. It is generated by running the Chapter 6 generator from q_0, until a singular point is reached.
2. Type B: The final generic excursion on a singular trajectory. Generated by running the Chapter 6 generator in reverse from q_f, until a singular point is reached.
3. Type C: Intermediate generic excursions. Generated from one singular point to another singular point. Since there exist multiple sustainable controls at the initial singular point, the generator also needs to

be given an initial control (picked from the sustainable set) in order to generate a type C generic excursion.

Building singulars from generic excursions and singular segments

The following algorithms outputs a list of singular trajectories that reach the goal. For a given instance of the problem, let q_{AL} be the state, in the control line frame, reached at the end of the type A generic excursion, and let q_{BL} be the state that the type C generic excursion begins with. All possible singular trajectories need to fill the gap between q_{AL} and q_{BL} with type C generic excursions and with singular segments. Since all the singular segments are translations of $\dot{x}_L = H$, we can consolidate them into a single singular segment.

The algorithm for generating the middle part of a singular is thus to branch off and keep following each possible generic excursion from q_{AL}. If it is possible to connect the obtained state q_{CL} to q_{BL} by a singular segment, output the resulting trajectory into the list of results, and continue in order to find all other singulars. Keep applying the same procedure recursively q_{CL} becoming the new q_{AL}. Stop computation in each branch when the total time of the trajectory under consideration has surpassed a reference time, e.g. the time of the trajectory found by the simple planner from Chapter 3.

The only part that needs to be further detailed in the algorithm is how to determine the singular segment connection between q_{AL} and q_{BL}. The states can be connected by a singular segment only if $\theta_A = \theta_B$. Furthermore, if $y_{AL} = y_{BL}$ and there exists a sustainable control u_0 at q_{AL} that effects a translation parallel to the control line, then q_{AL} and q_{BL} are united by an application of u_0 for an appropriate length of time.

If $y_{AL} \neq y_{BL}$ and there exist two translation controls u_1 and u_2 that are sustainable at q_{AL}, then q_{AL} and q_{BL} may be united by a tacking segment that consists in an application of u_1 followed by an application of u_2. This is always possible if u_1 and u_2 have $\dot{y}_L$ velocities that are of opposite signs. If the $\dot{y}_L$ velocities have the same sign, there are some cases when such a connecting segment cannot be built (i.e., if the minimum $\dot{y}_L$ is either in the wrong direction or overshoots the difference in y_L coordinates between q_{AL} and q_{BL}). In such instances, we simply continue the computation without outputting a trajectory into the results list.

7.2.2 Approximating generics

For generic trajectories other than TGT, we have found no analytical method to determine the remaining parameter of the control line. Our best method is to approximate the optimal trajectory in this case by sampling all possible values of H at a fine resolution and picking the trajectory with the minimum error.

Sampling H works because small changes in H usually correspond to small changes in the trajectory. As we have shown in Section 6.1.3, as long as the initial state changes in a way that doesn't cross a critical value of the Hamiltonian, the trajectory structure will stay constant, and the only change is in the duration of the various controls applied on the trajectory.

Phase shift metric

In order to conduct a search, we need a way to determine how close a trajectory comes to reaching the goal. As shown in Section 6.1.3, all generics become periodic after a maximum number of control switches. Given a position of the control line, we use the following metric. Generate a trajectory starting from q_0, and generate a trajectory starting from q_f. The trajectory starting from q_0 and corresponding to the given control line reaches the goal if and only if these two curves are exactly on top of each other past a certain point. In the general case, the curves differ. If they differ in structure, then there is no small change in the position of the control line that can change that structure (we would need to cross a critical value of H for that). We

consider the metric to be infinite in this case. Usually, the two trajectories have the same structure, but differ from each other in their *phase shift*, i.e. the x_L coordinate displacement. In such a case, we consider the value of the phase shift to be the metric.

Sampling H space

The Hamiltonian cannot be negative. Also, there exists a superior bound on the Hamiltonian, corresponding to the highest critical value, beyond which all the trajectories are whirls (which we have studied already). Thus, given an instance of the problem, the following uniform sampling algorithm is feasible:

1. Divide the $(0, H_{max})$ interval uniformly, with a resolution fine enough to ensure a good coverage of the smallest interval between critical values.
2. For each sampled value of H, compute the two possible positions of the control line from the H parametrization.
3. For each possible position of the control line, generate trajectories from q_0 and q_f and compute the phase shift metric.
4. Out of the trajectories that reach the goal (i.e. have a value of the phase shift metric that is considered close enough to zero), pick the fastest.

Sampling does not have to be uniform, but the resolution can be varied so as to sample more densely around points with low values of the phase shift metric.

The phase shift evidently changes continuously with changes in H. As H approaches a value corresponding to an optimal trajectory that reaches the goal, the phase shift goes to zero. For future work, it would be a very useful result to bound the rate of change in the phase shift as H varies, and to show that this function is Lipschitz continuous.

7.3 Finding optimal control policies in the cases when the control line cannot be parametrized by H

In the notation of Section 7.1, we can safely ignore the case when $c_{W0} = c_{Wf}$. In this case, the initial and final control have the same center of rotation in the world frame (or are parallel translations). Since the controls need to have the same H, the angular velocities (or planar velocities, for translations) are also the same. Therefore, for any such trajectory, we can shift small amounts of movement from the initial to the final control, and generate an infinite number of trajectories that take the same time. At least one of these trajectories (zero time on the initial control) will be found by other means.

Thus, the remaining case when the control line cannot be parametrized by H is when both the initial and the final motion are non-parallel translations. Let $(\dot{x}_0, \dot{y}_0)$ and $(\dot{x}_f, \dot{y}_f)$ be the world-frame velocities of these translations. Since

$$H = k_1\dot{x}_0 + k_2\dot{y}_0 = k_1\dot{x}_f + k_2\dot{y}_f, \tag{7.19}$$

$H > 0$, and the translations are not parallel, both H and the (k_1, k_2) direction vector for the control line are uniquely determined in this case. We use this information to compute the control line frame velocities for the two translations, $(\dot{x}_{0L}, \dot{y}_{0L})$ and $(\dot{x}_{fL}, \dot{y}_{fL})$ repectively. It must further hold that $\dot{x}_{0L} = \dot{x}_{fL} = H$.

There are two cases: either the trajectory contains a singular point, or it does not.

7.3.1 Singular trajectories that begin and end with translations

We will show that considering the case of singular trajectories that begin and end with translations is not necessary for finding an optimal trajectory. Assume the trajectory contains a singular point. If at least one of the $\dot{y}_L$ velocities of the initial and final controls is zero, then that control effects a translation parallel to the control line and the trajectory will be found in the regular singular search.

If $\dot{y}_{L0}\dot{y}_{Lf} \neq 0$, there exists a regular singular that is just as fast as the trajectory under consideration. This regular singular is obtained by completely removing the shorter of the two translations, and replacing it by a combination of elongating the regular singular segment, at the singular point that exists within the trajectory, and either reducing or elongating (according to the sign of $\dot{y}_{L0}\dot{y}_{Lf}$) the longer translation.

7.3.2 Generic trajectories that begin and end with translations

The solution of this case (translation-generic-translation, or *TGT trajectories*) is based upon an idea due to our colleague Weifu Wang. Assume there exists no singular point along an optimal trajectory that begins and ends with translations. As shown above, both the direction of the control line and H are uniquely determined.

Rotate the world frame such that the x axis is parallel to the direction of the control line determined above. Consider the locations of all the switching points between u_0 and other controls at q_0. Let y_+ be the minimum y coordinate for a switching point that needs to be above the control line for u_0 to be maximizing; let y_- be the maximum y coordinate for a switching point that needs to be below the control line for u_0 to be maximizing. In the rotated world frame, the k_3 parameter for the control line (i.e. the y coordinate of the world frame origin in the control line frame) needs to be in the $[-y_+, -y_-]$ interval.

Choosing k_3 anywhere in this interval will lead to exactly the same sequence of control switches, and the same duration for the subsequent control applications (since H is fixed). We will show that a translation of vector $(\dot{x}_{fL}, \dot{y}_{fL})$ cannot appear twice in this sequence. If it appeared twice, we can build an equivalent time trajectory by entirely removing the second application and making the first application longer by the same amount. Consider the locations where the active switching points are at control switches on the new trajectory. The active switching points are on one line (the old control line) on the first section of the trajectory, and on a different line on the second section of the trajectory. Therefore this trajectory does not verify the Pontryagin Principle, so it cannot be time optimal. Since the original, two-translations trajectory is time equivalent, it cannot be time optimal either. Therefore two parallel translations cannot appear on the optimal trajectory under consideration.

Since u_f thus can only appear once in its correct orientation, place the control line anywhere in the valid interval (e.g. $k_3 = -y_+$) and use the Chapter 6 generator to build a trajectory until the switch to u_f at the correct body orientation. Let d be the distance travelled by the rigid body between the first and the final control switches on this trajectory. Then the position of the control line can be exactly determined.

7.4 Conclusion

In the case of TGT trajectories, the position of the control line is not determined by the value of the Hamiltonian and we use a specific method for saving this case. All other cases where position of the control line cannot be parametrized by H are not necessary to be considered for finding an optimal trajectory. When parametrization is possible, we have two cases. For singulars, we exactly determine the position of the control line, and we pursue and exhaustive search based on stringing together generic excursions. For generics,

we sample the H space and use the phase shift metric to find an approximation of the optimal trajectory. This algorithm, together with the algorithm for finding whirls in Chapter 4, constitutes a solution to the problem posed in Section 3.1.

Chapter 8

Implementation and results

This chapter describes the results we have obtained by implementing and running the more common cases among the algorithms we have described. The implementation did not contain the less common “whirls” and “tacking” cases.

8.1 Implementation challenges

The main implementation challenge consisted in the fact that many of the algorithms presented in this work are geometric. As such, they can be easily visalized in the general case. However, a purely geometric approach is not straightforward to implement in code, and often misses out on specific cases. Developing the series of geometric formulas presented in Chapter 3 was thus an essential foundation for the implementation. Similarly, the “collision with line” algorithm of Chapter 6 is easy to visualize, but corresponds to relatively complex algebraic formulas.

Finding a suitable implementation language has also proven somewhat difficult. In the initial stages, we used Matlab, but were unable to extend the code beyond a certain size while also keeping it manageable. The algorithms presented are actually fast enough that we were able to also develop a in-browser Javascript version of a large subsection of the codebase, at one point. Eventually, we’ve settled on Python with NumPy, and recently our colleague Weifu Wang has been working on porting substantial sections of this code base to the C language, with a resulting speed-up factor of about one hundred. Numerical precision was also a factor we had to fine tune on a few occasions.

8.2 Results

We have implemented the search algorithms described in about three thousand lines of Python code. We have compared the results with those obtained by a direct planner method implemented by our colleague Wenyu Lu, which uses reverse kinematics to find the fastest piecewise constant control policies with up to four segments. This direct plannning algorithm first computes exactly all the possible three segment trajectories, according to the eight possible cases (e.g. rotation-translation-rotation or translation-rotation-rotation etc.). The algorithm then samples the possible durations for a fourth segment, to find the fastest trajectory with up to four segments.

With a naive uniform sampling strategy for the generic trajectory finder, as well as speed limitations of interpreted code, running the code takes on the order of ten minutes per configuration on a desktop

computer (including the running time for the reverse kinematics planner). We have run two tests, with one hundred random starting configurations for the symmetric omnidrive robot for each test. For each tested configuration, the following results hold as we would expect:

- A generic or singular exists and achieves the minimum time over all methods.
- If the generic or singular has three or four actions, the direct planner always finds a trajectory of the same speed. This is because the direct planner, as described above, always finds the fastest trajectory with three or four segments.
- If the fastest generic or singular has five or more actions, it is typically faster than three- and four-action trajectories from the direct method.

The first test picked one hundred random starting configurations from the range $q_0 \in [-5, 5] \times [-5, 5] \times [0, 2\pi)$, with the goal at the origin. For this relatively short distance to the origin, we found that the direct method came close to the fastest trajectory 90% of the time. Of the extremals, generics were fastest 76% of the time, and singulars 24% of the time. The second test picked one hundred random starting configurations from the wider range $q_0 \in [-10, 10] \times [-10, 10] \times [0, 2\pi)$, with the goal still at the origin. For this longer distance, the singulars proved fastest 81% of the time, and generics only 19% of the time. The direct method became less accurate, as expected, with only 83% of the runs finding a trajectory comparable to the fastest extremal.

For about 2% of the configurations tested, the direct planner was initially found to be faster than extremal trajectories. Upon closer examination, all of these cases were found to be caused by insufficient sampling. Fine tuning the sampling resolution resulted in finding extremal trajectories that were at least as fast as the direct planner in all cases.

Figures 8.2 - 8.9 show some of the results obtained, both in the tests described above for the omnidrive and in isolated runs for other vehicles.

Our colleague Weifu Wang has since re-implemented the generic search code in C. This has reduced the average search time for a random starting configuration to less than a second, on a standard 2010 desktop computer, for the symmetrical omnidirectional vehicle with seventeen controls in the canonical control set. For larger control sets, we expect this duration to grow no worse than polynomially with the square of the number of controls.

The increased speed of the C code allowed for running a more extensive batch of tests, shortly before the finalization of this work. An extensive test was run for the differential drive, with over fourteen thousand starting configurations in the $[-3, 3] \times [-3, 3] \times \{\frac{\pi}{4}\}$ set. The results of this test are shown graphically in Figure 8.1. This figure generally corresponds with Figure 12 from [3] (see this work for a detailed description of the trajectory types indicated by the different colors). Numerical imprecision is causing some blurring and speckling, particularly around the isocost lines.

For each of the differential drive, Dubins car, and Reeds-Shepp car, we have also run a further batch of one thousand random starting configurations in the $[-3, 3] \times [-3, 3] \times [0, 2\pi)$ set. A further one thousand random starting configurations in $[-7, 7] \times [-7, 7] \times [0, 2\pi)$ were tried for the symmetric omnidirectional vehicle. Optimal trajectories were found in all cases, with an exception for the Reeds-Shepp car that we detail below.

For the differential drive, generics were fastest in 645 cases, and singulars in 355. For the Dubins car, generics were fastest in 252 cases, and singulars in the remaining 748. For the symmetric omniwheeled vehicle, generics were fastest in 477 cases, and singulars were fastest in 523 cases.

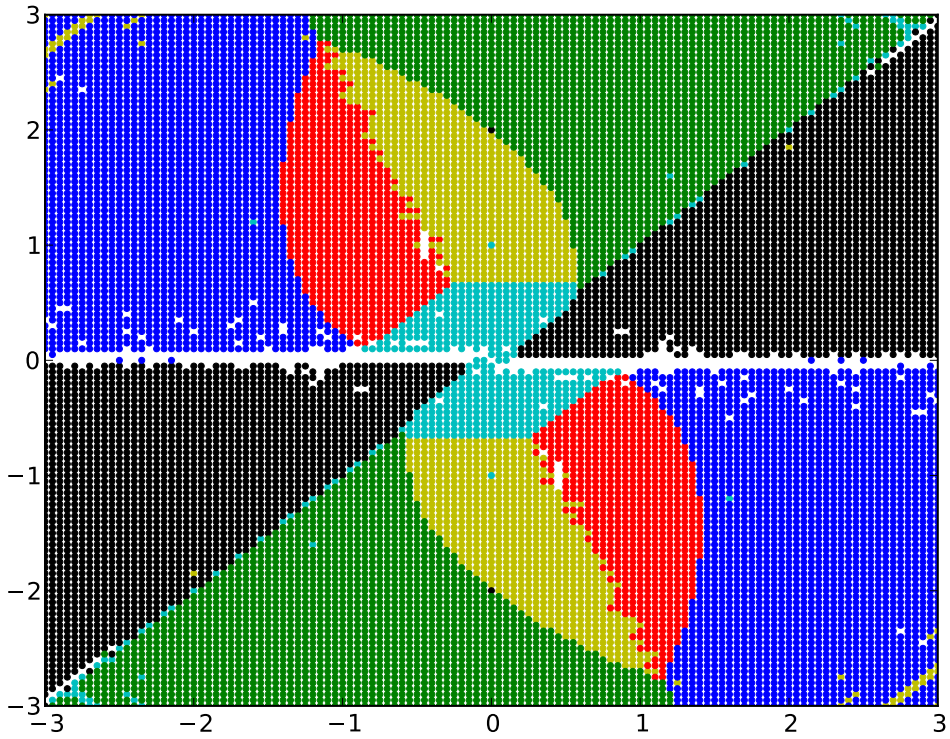


Figure 8.1: A summary of over fourteen thousand optimal trajectories for the differential drive, as found by our algorithm. Trajectories start with configuration $(x, y, \pi/4)$ and end at the origin. The colors indicate different trajectory structures.

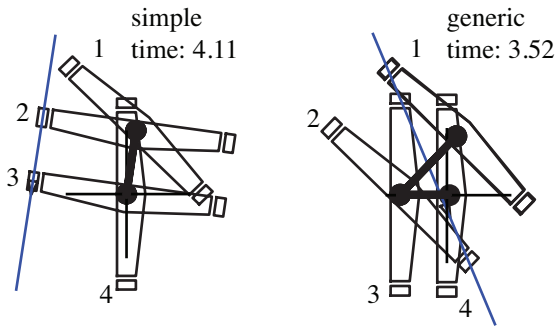


Figure 8.2: A simple planner trajectory for the differential drive compared to a generic trajectory.

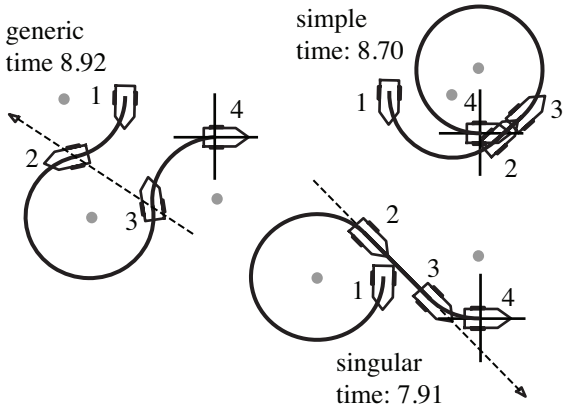


Figure 8.3: Comparison of three Dubins car trajectories: simple planner output, fastest singular and fastest generic.

For the Reeds-Shepp car, generics were fastest in 217 cases, and singulars in 673 cases. In the remaining 110 cases, neither generics nor singulars were found. We have confirmed that these cases were whirls (the whirl-finding algorithm not having been implemented) by running one thousand more tests for the Reeds-Shepp car for starting configurations in the $-\pi/2$ and $-3\pi/2$ planes and plotting out the configurations for which no generic or singular were found. The shapes and sizes of the resulting plots closely corresponded to the central regions (corresponding to whirls) in Figures 14 and 15, pp. 134-135 from [24].

In all cases, the trajectories found were at least as fast as those found by the simple universal planner. A comparison with the reverse kinematics planner is pending, based on a re-implementation of this code base in C as well.

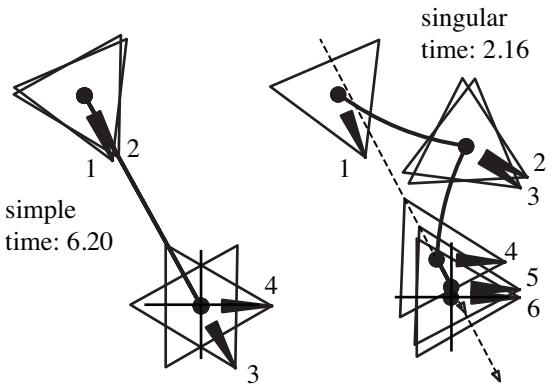


Figure 8.4: Simple planner output, compared to a generic trajectory for the omnidrive.

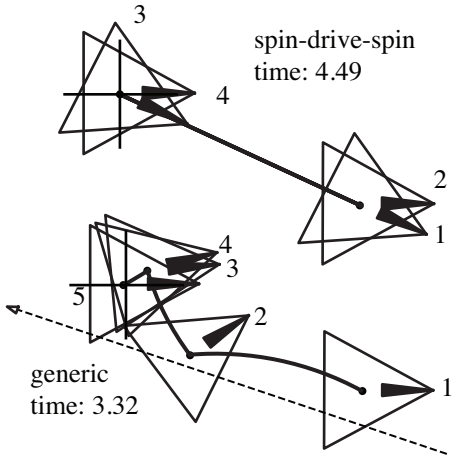


Figure 8.5: Simple planner output, compared to a singular trajectory for the omnidrive.

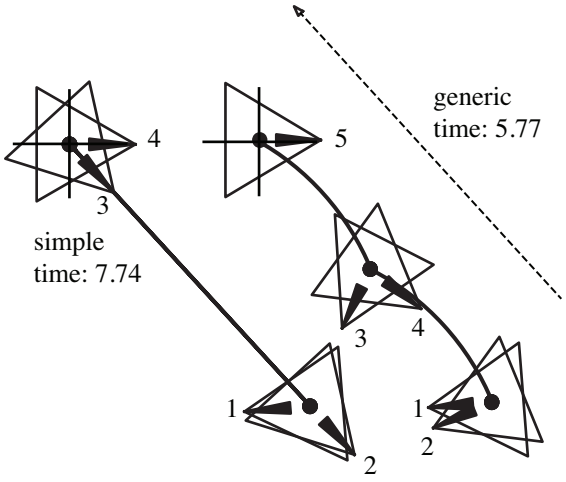


Figure 8.6: Simple planner output, compared to a generic trajectory for the omnidrive.

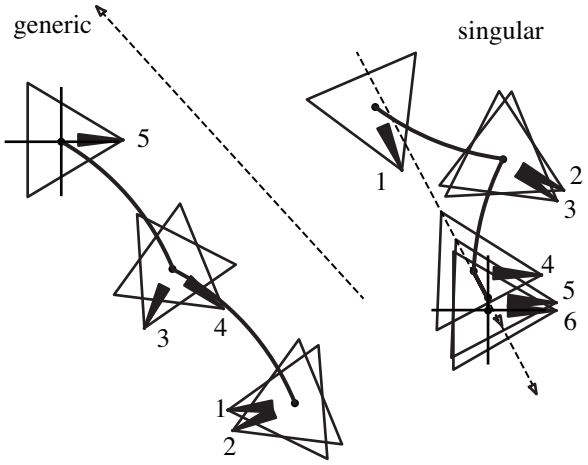


Figure 8.7: Fastest singular and fastest generic trajectories connecting a given pair of configurations for the omnidrive.

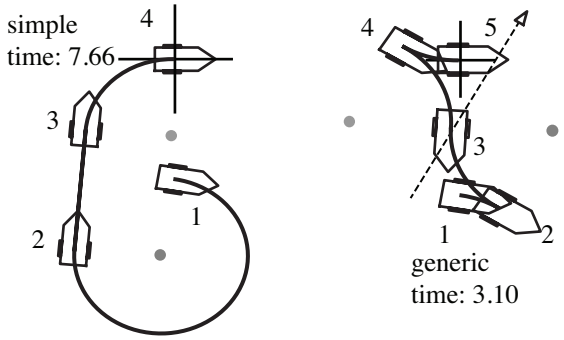


Figure 8.8: Simple planner output, compared to the fastest trajectory for the Reeds-Shepp car.

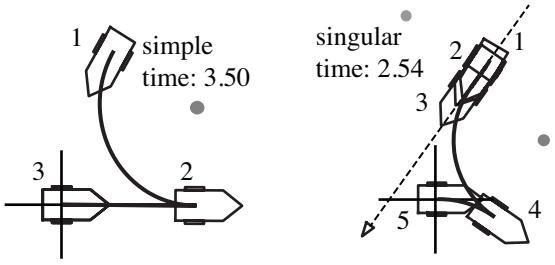


Figure 8.9: Simple planner output, compared to the fastest trajectory for the Reeds-Shepp car.

Chapter 9

Future work and conclusions

In this chapter, we discuss promising directions for future work based on the current results, and we conclude the present work with a series of general observations and lessons learned.

9.1 Future work

An immediate improvement for the work done would be to implement the whirls and tacking cases. Tacking trajectories do not occur for any of these vehicles, as they require an entire edge of U to be in the $u_\theta = 0$ plane. Whirls only occur in a very simple case for the Reeds-Shepp car (three-point turns).

The sampling strategy used for finding the fastest generic can be much improved. For uniform sampling, we have a sketch of a proof that, for any degree of approximation desired for the optimal trajectory time, there exists a corresponding sampling resolution for H that achieves a trajectory withing that approximation. The running time can also be considerably sped up by implementing non-uniform sampling and using, for example, a sampling strategy based on Newton's method.

Currently, we examine a model in which there are no obstacles and the controls switch instantaneously. There are reasons to believe that these restrictions can be lifted from the theoretical analysis. By applying additional results from classical optimal control theory ([18]) to an improved model, it seems possible to obtain results of very immediate practical relevance.

9.1.1 Costly switches

One of the more serious difficulties encountered by geometric optimal control analysis has been the so-called "chattering" phenomenon (see, for example, [28] and the work presented above). Some motion vectors can always be obtained from a combination of infinitesimal movements in other directions. There will, furthermore, exist optimal trajectories, for simple kinematic or dynamic models, that involve such chattering motions. Such trajectories can never be followed by a practical controller, as the controller's feedback loop will always involve some delay in changing the control, and the physical components themselves have inertia. One possible approach to this problem is to show that, for each chattering optimal trajectory, there exists a non-chattering equivalent trajectory; but this does not hold in all situations (see [11] for an example).

A more satisfying approach is to explicitly model the fact that control switches are inherently costly; in a practical situation, we will almost always prefer a slightly slower trajectory with three switches to another trajectory with billions of switches. In his seminal work that constitutes the basis of our analysis, Pontryagin ([18] p. 263) acknowledges this issue and suggests the following solution: "in a number of cases,

the controllers have a definite inertia [...]. Let us [then] make the parameter u a phase variable, and let us take the derivative of u to be our control parameter".

Such a model is similar, in some respects, to the delays involved in a driver's short range maneuvers with an automobile (e.g. three-point turn, parallel parking). To go from "forwards-left" to "backwards-right", the driver has to stop the vehicle and adjust the "shifting" controls (steering wheel, transmission stick) to different positions, which takes time. Some velocity shifts (e.g. "forwards-straight" to "backwards-straight") can be achieved faster than others.

Costly switches cannot be dealt with by the Pontryagin Principle in a direct manner. However, we can build an augmented mathematical model, suitable to analysis through the Pontryagin Principle, in the following way. We will augment the space of controls with some new dimensions consituting a "shifting space". Let the shifting space contain n privileged points, corresponding to the old controls ${}^R\dot{q}_1 \dots {}^R\dot{q}_n$. We will keep the old controls, and call them "movement" controls. Applying control i will result in a movement of velocity ${}^R\dot{q}_i$ in the plane if and only if the vehicle is at the point corresponding to control i in shifting space (in order to keep $\dot{q}$ differentiable with respect to q, as required by the Maximum Principle, we will in fact associate with each movement control a very sharp bump function, peaking at the corresponding point in shifting space). It is therefore never optimal to apply control i unless we are at the right point in shifting space. For each oriented pair of movement controls (i, j) we will furthermore create a "shifting" control, which keeps the rigid body motionless in the plane, but moves between the shifting space points corresponding to i and j in a time interval corresponding to the delay we want to model as involved in such a switch. The shifting control should, furthemore, be unable to switch between any other pair of movement controls.

While the presented model introduces a number of additional controls, we have no reason to suspect that these controls change the existing analysis, as the shifting controls have no impact on position in the plane, and therefore should not change the value of the components of the adjoint that lie in $\mathbf{SE}(\mathbf{2})$. This intuition needs to be, however, verified by rigorous proof.

A model with costly switches has additional benefits, besides ruling out chattering. It allows us to accurately model tasks such as pushing a rigid body (e.g. a part on an assembly table, or a piece of furniture in a room), a situation in which changing the pushing direction has costs associated with re-orienting the manipulator towards the new direction. For a very simple example, we may consider the problem of a robot moving a heavy bench by lifting one end at a time and rotating the bench around the other end. A naive kinematic model would yield trajectories that, for instance, achieve a forward translation by chattering the two allowed motions. The augmented model would, in effect, keep track of the robot's position along the bench as well. Such an analysis should quantitatively model the practical trade-off between costly control switches and continuous motion: we may often prefer a longer trajectory, if it has fewer control switches.

Furthermore, a costly switches model allows the analysis of hybrid kinematic-dynamic models, in which we replace a "dynamic" control switch (change the direction of the force, movement slows down in the current direction, speeds up in the new direction) with a corresponding "delayed" switch (keep going at the current speed a little further, stop for a bit, start at full speed in the new direction) that takes the same amount of time. As dynamic physical models have proven difficult to solve with the Maximum Principle in the past, this approach could yield analytical optimal trajectories for vehicles with significant inertia.

In conclusion, we consider a costly switches model of rigid body motion as a significant improvement, and we see the following as the main issues in constructing and analyzing such a model:

- Construct a rigorous mathematical definition of a costly switches model that is suitable to non-trivial application of the Maximum Principle.

- Apply the Maximum Principle to this definition, solve for the adjoint and compare with the current "inertialess" model.
- Find a way to transform any given dynamic model into a time-equivalent costly switches kinematic model, or characterize the limits of such transformation, if these limits exist.

Existence of optimal trajectories for the costly switches model

We wish to present a proof of existence of optimal trajectories for the costly switches model. This proof was developed in collaboration with Vladimir Chernov.

Consider a controllable system with a finite number of controls and costly switches between pairs of these controls; the switch between any two controls costs at least $\epsilon > 0$. We will prove that, under some reasonable assumptions about the cost function, optimal trajectories always exist for such a system. First, we will give a bound on the number of switches for an optimal trajectory; second, we will show that, given this bound, an optimal trajectory can be attained.

The first part is trivial. Assume that the cost of applying a control, after switching to it, is always positive. Then for any pair of configurations there is an upper bound on the number of switches of a minimum cost trajectory connecting the two configurations: c_c/ϵ, where c_c is the cost of a trajectory uniting the two configurations. Coupled with the finite number of controls, this immediately gives us a finite possible number of structures for the optimal trajectory. We will next prove that, for any such trajectory structure, the existence of an admissible trajectory of this structure implies the existence of an admissible optimal trajectory of the same structure.

Let q denote a state of the system. We assume that the system has a closed set Q of *admissible* states (e.g., states that are in obstacle-free space). A trajectory is described by a sequence of controls, and the time for which each control is applied. Assume a finite set of controls. We define each control as a differentiable function $F_i : Q \times \mathbb{R}^+ \mapsto Q$, where $F_i(q_0, t) = q_1$ iff q_1 is the resulting state after applying control i for time t in state q_0.

A *trajectory structure* of length n consists of a finite sequence S of positive integers that represent indices into the control set. Given a structure S, a trajectory is specified by a corresponding equal length sequence T of positive real numbers, containing the times each control is applied.

The *cost functions* $c_i(q_0, t) >= 0$ are real valued differentiable functions such that the pre-image of any interval $[0, b]$ under c_i is a bounded set (i.e., arbitrarily long trajectories have arbitrarily high costs as well).

In the following, we will keep the trajectory structure fixed, and vary the control times to obtain different trajectories. Given an initial state q_0, we will define $q_i(p)$ to be the state obtained after applying control i for time $T[i]$ along the trajectory T. q_i is given by the recurrence relation

$$q_i(T) = F_i(q_{i-1}(T), T[i]). \tag{9.1}$$

At time t along trajectory p, the system is in state

$$q(T, t) = F_h(q_{m-1}(T), t - \sum_{i=1}^{h} T[i]), \tag{9.2}$$

where h is the highest index such that $t \geq \sum_{i=1}^{h} T[i]$. We will call a trajectory p *admissible* if, for all $t \leq \sum_{i=1}^{n} T[i]$, the state $q(p, t)$ is admissible. We will call a state q_f *reachable* from q_0 if there exists an admissible trajectory p_e such that $q_n(p_e) = q_f$.

The cost of trajectory p is

$$c(p) = C + \sum_{i=1}^{n} c_i(T[i], q_i), \tag{9.3}$$

where C is an overhead cost associated with a trajectory of the given structure (in our case, this contains the cost of switching controls). Since we are only interested in finding the minimum cost trajectory associated with the given trajectory structure, we can safely disregard C in the following analysis.

We will prove the following lemma:

Lemma 18 *Let function sequences F_i, c_i, $i = 1 \ldots n$, define a trajectory structure. Let q_0 be an initial configuration for the system and let q_f be a configuration that is reachable from q_0. If the set of all configurations reachable from q_0 is a closed set, then there exists a minimum cost trajectory from q_0 to q_f.*

Proof: Consider the set of all costs of admissible trajectories from q_0 to q_f. The set is nonempty, and has a lower bound of zero. Therefore, the set has an infimum m. Therefore, there exists a sequence of trajectories T_i such that $c(T_i) \leq m + 2^{-i}$. For this sequence,

$$\lim_{i \to \infty} c(T_i) = m. \tag{9.4}$$

Consider the sequence formed by the time of control 1 for all trajectories in T_i, $T_i[1]$. Since $0 \leq c(T_i[1]) < m + 1$, the assumption we have made about the cost function indicates that the $T_i[1]$ sequence is bounded. By the Bolzano-Weierstrass theorem, this sequence must contain a convergent subsequence t_{1j} that has a limit of t_{1l}. We construct the corresponding subsequence T_j from the initial sequence of trajectories, and repeat the procedure for the times of controls $2, \ldots n$. We thus obtain a sequence of limits $t_{1l}, t_{2l}, \ldots t_{nl}$. This is an admissible trajectory, since the reachable space is closed, and each subsection of this trajectory is the limit of a convergent sequence of admissible trajectories. The constructed trajectory is also the limit of the T_i sequence, and therefore its cost is m. ■

9.1.2 Obstacles

Pontryagin's book ([18]) has all of chapter five dedicated to "Optimal Processes with Restricted Phase Coordinates". The authors consider an optimal trajectory lying entirely in a closed region G, with piecewise smooth boundaries described by a curve $g(x) = 0$, and obtain the following strong results (p. 311):

- Each section of the trajectory that lies in the interior of G satisfies the Maximum Principle.
- Each section of the trajectory that lies on a smooth piece of the boundary of G satisfies Theorem 22 (p. 267).
- At each junction point, the jump condition (p. 302) is satisfied.

The implications of the first result have already been analyzed. The second result is more difficult to interpret; it seems to imply that the boundary of G will only be followed when it has a certain shape; what this means in the context of our problem remains to be seen. The jump condition appears to be relatively more straightforward; in our notation, we can express it as follows. Consider an optimal trajectory, and let t_j be a junction time during it. Let $\lambda^-(t)$ be the adjoint on the time interval preceding t_j, and $\lambda^+(t)$ the

adjoint on the time interval succeeding t_j. Then there exists a real number μ such that one of the following two mutually exclusive conditions holds:

$$\lambda^-(t_j) + \mu \text{ grad } g(x(t_j)) = 0, \tag{9.5}$$

or

$$\lambda^-(t_j) + \mu \text{ grad } g(x(t_j)) = \lambda^+(t_j). \tag{9.6}$$

The implications for our problem seem to be quite geometric. Recall that the (x, y) component of the adjoint, in our case, is a constant direction in the plane (k_1, k_2). Therefore, the above conditions mean that, at a junction point, either the "incoming" control line is parallel to the tangent to the surface; or otherwise, the "outgoing" control line is the reflection of the "incoming" line around the normal to the surface. This seems to indicate an intriguing scenario, in which the control line reflects itself around a maze of obstacles.

These are, in conclusion, the main issues in studying a phase-restricted model within the context of our problem:

- Determine the implications of Pontryagin's Theorem 22: what kind of obstacle boundaries can be followed, and what kind can not?
- Determine the implications of the jump condition in the θ coordinate as well: can we say anything about the location of the control line at the jump point, as opposed to just its direction?
- Develop the two points above into a global theory of rigid body optimal motion in the obstacle-obstructed plane.

9.2 Lessons learned

The essential underpinning of our work is the Pontryagin Principle, but this principle by itself is nowhere near sufficient for solving the problem at hand. The approach we have followed does suggest a general strategy for approaching a class of optimal control problems. Consider a system that has an n dimensional state, a polyhedral control space, and a linear mapping from control space to velocity space. Any optimal trajectory for this system has an adjoint $\lambda(t)$. We do not know the initial value of the adjoint, $\lambda(0)$; but, once this value is set, there will usually only exist one corner of the polyhedral control space that maximizes the projection of the system's velocity onto the adjoint, i.e. the Hamiltonian. Applying this control will result in changes in both the system's state, and in the adjoint as characterized by the Pontryagin differential equation. Once a multimax point is reached, sustainability analysis is necessary in order to indicate which control can be chosen. In order for the trajectory to be singular, it is necessary for the adjoint to stay in the same orientation relative to velocity space; this seems a strong condition, which only allows few controls to be applied in order to obtain such a situation.

This is obviously just a very general approach, and a lot depends on the specifics of the system studied. An analytic integration of the adjoint function is very useful, but seems difficult to achieve for most systems; in this respect, rigid bodies in the plane are a fortunate case. Sustainability analysis and a characterization of singular segments are also very important, and seem to depend a lot on the specifics of the system studied.

For our particular case, geometric considerations are very important. On the one hand, this is helpful as it makes everything easier to visualize. On the other hand, an algebraic study of the problems involved, in a form suitable for algorithmic implementation, is hampered by the scarcity of all-purpose geometric

computational tools. Procedures as simple as computations with angles, or an easy way to compute the time to collision of a constant-velocity point with a line, are not readily available and need to be implemented from scratch. We hope that our results in this direction will also prove useful for other geometric computational efforts, not necessarily related to optimal control.

Overall, the problem is surprisingly challenging, and leads to a number of very distinct cases that need to be considered. The practical implications and the theoretical simplicity of the mathematical question are, however, a source of distinct satisfaction.

Bibliography

[1] Pankaj K. Agarwal, Therese Biedl, Sylvain Lazard, Steve Robbins, Subhash Suri, and Sue Whitesides. Curvature-constrained shortest paths in a convex polygon. In *SIAM J. Comput*, pages 392–401. ACM Press, 1998.

[2] Devin J. Balkcom, Paritosh A. Kavathekar, and Matthew T. Mason. Time-optimal trajectories for an omni-directional vehicle. *International Journal of Robotics Research*, 25(10):985–999, 2006.

[3] Devin J. Balkcom and Matthew T. Mason. Time optimal trajectories for differential drive vehicles. *International Journal of Robotics Research*, 21(3):199–217, 2002.

[4] R. Bellman and R. Kalaba. *Dynamic Programming and Modern Control Theory*. Academic Press, 1965.

[5] Jean-Daniel Boissonnat, André Cérézo, and Juliette Leblond. Shortest paths of bounded curvature in the plane. In *Proceedings of the 1992 International Conference on Robotics and Automation*, pages 2315–2320, 1992.

[6] H. R. Chitsaz. *Geodesic problems for mobile robots*. PhD thesis, University of Illinois at Urbana-Champaign, 2008.

[7] Hamidreza Chitsaz, Steven M. LaValle, Devin J. Balkcom, and Matthew T. Mason. Minimum wheel-rotation paths for differential-drive mobile robots. In *IEEE International Conference on Robotics and Automation*, 2006. To appear.

[8] M. Chyba and T. Haberkorn. Designing efficient trajectories for underwater vehicles using geometric control theory. In *24rd International Conference on Offshore Mechanics and Artic Engineering*, Halkidiki, Greece, 2005.

[9] E. J. Cockayne and G. W. C. Hall. Plane motion of a particle subject to curvature constraints. *SIAM Journal on Control*, 13(1):197–220, 1975.

[10] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.

[11] Guy Desaulniers. On shortest paths for a car-like robot maneuvering around obstacles. *Robotics and Autonomous Systems*, 17:139–148, 1996.

[12] 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.

[13] Andrei A. Furtuna and Devin J. Balkcom. Generalizing dubins curves: Minimum-time sequences of body-fixed rotations and translations in the plane. *Int. J. Rob. Res.*, 29:703–726, May 2010.

[14] Andrei A. Furtuna, Devin J. Balkcom, Hamidreza Chitsaz, and Paritosh Kavathekar. Generalizing the Dubins and Reeds-Shepp cars: Fastest paths for bounded-velocity mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2533–2539, 2008.

[15] Andrey Andreyevich Markov. Some examples of the solution of a special kind of problem in greatest and least quantities. In *Soobshch. Karkovsk. Mat. Obshch.*, pages 250–276, 1887.

[16] H.J Oberle and W. Grimm. Bndsco. a program for the numerical solution of optimal control problems. 1989.

[17] A. Rege P. Jacobs and J-P. Laumond. Non-holonomic motion planning for hilare-like mobile robots. In *Proceedings of the International Symposium on Intelligent Robotics*, 1991.

[18] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. John Wiley, 1962.

[19] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[20] David B. Reister and Francois G. Pin. Time-optimal trajectories for mobile robots with two independently driven wheels. *International Journal of Robotics Research*, 13(1):38–54, February 1994.

[21] Marc Renaud and Jean-Yves Fourquet. Minimum time motion of a mobile robot with two independent acceleration-driven wheels. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 2608–2613, 1997.

[22] Paolo Salaris, Felipe A. W. Belo, Daniele Fontanelli, Luca Greco, and Antonio Bicchi. Optimal paths in a constarained image plane for purely image-based parking. In *International Conference on Intelligent Robots and Systems*, pages 1673–1680, 2008.

[23] J.M. Selig. *Geometrical Methods in Robotics*. Springer-Verlag, 1996.

[24] P. Souères and J.-D. Boissonnat. Optimal trajectories for nonholonomic mobile robots. In J.-P. Laumond, editor, *Robot Motion Planning and Control*, pages 93–170. Springer, 1998.

[25] Philippe Souères and Jean-Paul Laumond. Shortest paths synthesis for a car-like robot. *IEEE Transactions on Automatic Control*, 41(5):672–688, May 1996.

[26] Héctor Sussmann. The Markov-Dubins problem with angular acceleration control. In *IEEE International Conference on Decision and Control*, pages 2639–2643, 1997.

[27] Héctor Sussmann. The brachistochrone problem and modern control theory. In *Contemporary trends in non-linear geometric control theory and its applications*, pages 113–165, 2002.

[28] Héctor Sussmann and Guoqing Tang. Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control. SYCON 91-10, Department of Mathematics, Rutgers University, New Brunswick, NJ 08903, 1991.

[29] M. Vendittelli, J.P. Laumond, and C. Nissoux. Obstacle distance for car-like robots. *IEEE Transactions on Robotics and Automation*, 15(4):678–691, 1999.