2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Generalizing the Dubins and Reeds-Shepp cars: fastest paths for bounded-velocity mobile robots

Andrei A. Furtuna, Devin J. Balkcom
Hamidreza Chitsaz, Paritosh Kavathekar

Abstract— **What is the shortest or fastest path a mobile robot can follow between two configurations in the unobstructed plane? The answer to this fundamental question is only known analytically for a few planar mobile robots: the Dubins and Reeds-Shepp steered cars, the differential drive, and a particular omnidirectional robot. This paper explores the optimal trajectories for a general parameterized model of a mobile robot that includes each previously-studied vehicle as a special case. The model also allows characterization of the optimal trajectories for several other mobile robot designs for which the optimal trajectories have not been previously explored.**

The paper applies Pontryagin's Maximum Principle to the generalized robot to find necessary conditions that optimal trajectories must satisfy, and gives geometric interpretations of the conditions. We also present an algorithm that generates and classifies all optimal trajectories for a given design.

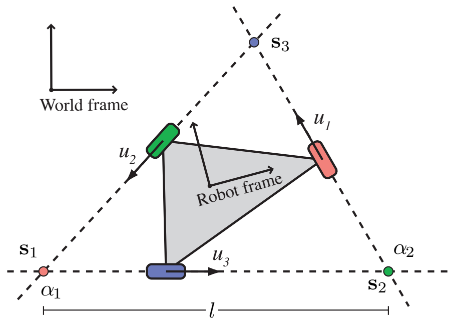


Fig. 1: A 'generalized' wheeled vehicle. With appropriate constraints on the controls, this vehicle can simulate any bounded-velocity rigid body in the plane.

I. Introduction

There may be an infinite set of paths that connect a start and goal configuration for a mobile robot. Some of the paths will be faster or shorter than the others. The optimal paths are only known analytically for four ground vehicles: Dubins and Reeds-Shepp steered cars, the differential-drive, and the symmetric three-wheeled omnidirectional vehicle. This paper develops a single, more-general model that explains the optimal trajectories for these and similar but unstudied systems.

Understanding optimality is both basic and useful. A fork-lift robot operating in a constrained factory environment should have a different design than a race car. Knowledge of the analytical optimal trajectories may allow alternate mechanical designs to be considered rapidly, independently of the particular planning or control system implemented on the robot. Optimal trajectories may also be used as a set of primitives for more complete planning systems that take obstacles and vehicle dynamics into account.

Figure 1 shows the model. There are three *omniwheels*, like those used to build both omnidirectional robots and certain types of conveyer belts. Omniwheels slip sideways freely, but like standard wheels, slip only minimally in the direction they are facing. While the practical construction of an omniwheel may be intricate, omniwheels provide a very convenient model: each provides control over one degree of freedom, without limiting the others.

The state of the robot is its position and orientation in the plane, given by the coordinates (x, y, θ). There are three wheels; the speeds of the wheels in the facing direction, (u_1, u_2, u_3), determine the velocity and angular velocity of the robot.

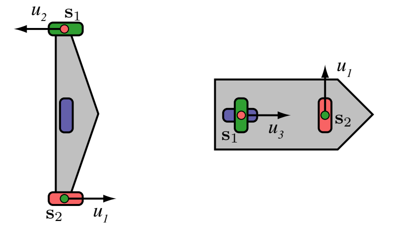
Different placements of the wheels lead to different kinematics for the system. Figure 2 shows how a differential-drive and the Dubins and Reeds-Shepp steered cars can each be built using omniwheels. For example, a differential-drive can be built by placing omniwheels at the same place on the robot as rubber wheels would be placed; However, these two omniwheels are not enough – the robot can slip sideways freely. We therefore place a third wheel on the line between the first two wheels, perpendicular to the first two wheels, and set the corresponding control to be zero.

This paper three main results about the time-optimal trajectories of the omniwheeled robot with independent bounds on the controls.

1) Every optimal trajectory is composed of arcs of circles and straight lines.
2) For every optimal trajectory there is a line in the plane (the *control line*). There are three switching points attached to the robot, at the intersections of the lines through the wheels. If a switching point falls below the control line, the corresponding wheel spins at maximum speed in the forward direction; above the line, and the wheel spins at maximum speed in the reverse direction. If a switching point falls on the line, the robot may translate along the line.
3) We present a geometric algorithm that enumerates all optimal trajectory classes for a given vehicle design.

Although we do not discuss the details in this paper, we have also shown that the problem of finding the optimal trajectory between a particular start and goal can be reduced to one-dimensional root-finding.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.



a) Differential drive b) Dubins/ Reeds-Shepp

Fig. 2: Models of differential drives, Dubins cars, and Reeds-Shepp cars built using omniwheels.

A. Background and related work

The structures of the optimal trajectories have been previously found for kinematic models of four vehicles: a steered car that can only go forwards (Dubins [3]), a car that can also go backwards (Reeds and Shepp [6]), the common differential-drive robot (Balkcom and Mason [2]) and the symmetric three-wheeled omnidirectional robot (Balkcom, Kavathekar and Mason [1]).

Reeds and Shepp's results were further refined by several authors. Sussmann and Tang [12] reduced the number of families of trajectories thought to be optimal by two, and Souères and Boissonnat [9], and Souères and Laumond [10] discovered the mapping from pairs of configurations to optimal trajectories.

All of these reults were achieved by using bounded-velocity, kinematic models. Although ignoring accelerations is not completely satisfying, the optimal-control problem for dynamic models of ground vehicles appears to be very difficult; the differential equations describing the trajectories do not have recognizable analytical solutions, and the optimal trajectories may involve chattering [11]. Tight optimality bounds have proven difficult to determine, as has complete characterization of the geometric structure of trajectories, even though papers by Reister and Pin [7], and Renaud and Fourquet [8] present numerical and partial geometric results for steered cars, and Kalmár-Nagy *et al.* [4] present algorithms for numerically computing optimal trajectories for a bounded-acceleration model of the symmetric omnidirectional robot.

II. Model and kinematics

The robot is a rigid body, with state $q = (x, y, \theta)$, the location and orientation of a frame attached to the robot with respect to a fixed reference frame. Let $\dot{\hat{q}} = (\dot{\hat{x}}, \dot{\hat{y}}, \dot{\hat{\theta}})$ be the generalized velocity of the robot in its own frame of reference. The relationship between the generalized velocities in the two frames of reference is given by

$$\dot{q} = R(\theta)\dot{\hat{q}}, \tag{1}$$

where

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

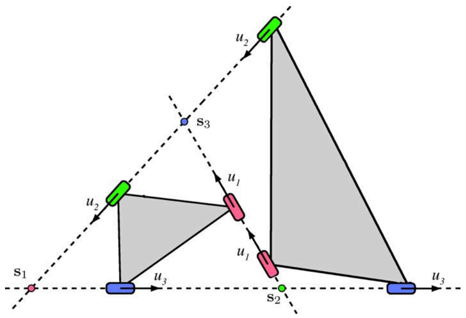


Fig. 3: Two robots sharing the same design class. Their wheel lines and switching points are identical.

We define a bounded-velocity rigid body to be any rigid body in the unobstructed plane with $\dot{\hat{q}} \in V$, where V is a set with boundary. In this paper we will only consider cases where V is invariant to changes in q.

Let u_i be the controlled component of the velocity of the ith wheel, as shown in figure 1. We place bounds on the controls: $u_i \in [l_i, h_i]$, for $l_i, h_i \in \mathbf{R}$ with $l_i < h_i$.

We would like to find a mapping from the controls u to the generalized velocity $\dot{q}$, and we expect it to be linear:

$$\dot{q} = Bu. \tag{3}$$

The mechanism is parallel, so to find B, we find the mapping from robot velocities to controls, and invert that mapping.

Since translating one of the wheels along its wheel line will not change the mapping between controls and velocities, the design of the robot can be given by any three numbers that describe the (possibly open) triangle formed by the intersections of the wheel lines, with vertices $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$ (see figure 1).We call this triangle the *switching triangle*. All robots that share the same switching triangle are equivalent, forming a single design class (see figure 3). We choose to use a side length, l, and two angles, α_1 and α_2, to describe the switching triangle, as shown in figure 1.

The plane velocity of a point at the intersection of two wheel lines depends only on the speeds of the two wheels that lie along those lines. It is therefore convenient to fix the robot frame to a vertex of the switching triangle, and align the x axis of the robot frame with one of its sides.

We collect all the terms dealing with the robot's design into a single matrix A, mapping between velocities and controls in the robot's frame. With this choice of reference frame,

$$A = \begin{bmatrix} \cos\alpha_2 & \sin\alpha_2 & l\sin\alpha_2 \\ \cos\alpha_1 & -\sin\alpha_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \tag{4}$$

and

$$A^{-1} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -\csc\alpha_1 & \cot\alpha_1 \\ \frac{\csc\alpha_2}{l} & \frac{\csc\alpha_1}{l} & -\frac{\cot\alpha_1+\cot\alpha_2}{l} \end{bmatrix}. \tag{5}$$

We will assume for the remainder of the paper that the placement of the wheels is such that the three lines through the wheels in the controlled directions are unique, not all parallel, and do do not intersect at a single point. In this case, A is non-singular, B exists, and

$$\dot{q} = Bu = RA^{-1}u. \tag{6}$$

Theorem 1: The vehicle follows either an arc of a circle or a straight line over any constant controls interval.

Proof: Direct from integration of equation 6. ■

In the robot reference frame, the center of rotation O is located at

$$\left(-\frac{\dot{\hat{y}}}{\omega}, \frac{\dot{\hat{x}}}{\omega}\right). \tag{7}$$

III. The Maximum Principle

This section uses Pontryagin's Maximum Principle [5] to derive necessary conditions for time-optimal trajectories. The Maximum Principle states that for any time-optimal trajectory $q(t)$ with corresponding control $u(t)$:

1) There exists a non-trivial (not identically zero) *adjoint function*: an absolutely continuous $\mathbf{R}^3$-valued function of time, $\lambda(t)$,

$$\lambda(t) = \left(\begin{array}{c} \lambda_1(t) \\ \lambda_2(t) \\ \lambda_3(t) \end{array}\right),$$

defined by a differential equation, the *adjoint equation*, in the configuration and its time-derivatives:

$$\dot{\lambda} = -\frac{\partial}{\partial q}\langle \lambda, \dot{q}(q,u)\rangle \quad \text{a.e.} \tag{8}$$

We call the above inner product the *Hamiltonian*:

$$H(\lambda, q, u) = \langle \lambda, \dot{q}(q,u)\rangle. \tag{9}$$

2) The control $u(t)$ minimizes the Hamiltonian:

$$H(\lambda(t), q(t), u(t)) = \min_{z \in U} H(\lambda(t), q(t), z) \quad \text{a.e.} \tag{10}$$

3) The Hamiltonian is constant and non-positive over the trajectory. We define λ_0 as the negative of the value of the Hamiltonian.

A. Application of the Maximum Principle to time-optimal bounded-velocity rigid bodies in the plane

Theorem 2: The adjoint function corresponding to the time-optimal trajectories of a rigid body in the unobstructed plane, with configuration $q = (x, y, \theta)$, is

$$\lambda = \left(\begin{array}{c} k_1 \\ k_2 \\ k_1 y - k_2 x + k_3 \end{array}\right), \tag{11}$$

regardless of the controls for the system, so long as the controls are invariant under $SO(2)$ action.

Proof: For any rigid body, the configuration space velocity of the body in the universal (fixed) coordinate frame is given by

$$\dot{q} = R\dot{\hat{q}}. \tag{12}$$

The adjoint equation is

$$\dot{\lambda} = -\frac{\partial}{\partial q}\langle \lambda, \dot{q}(q,u)\rangle \tag{13}$$

$$= -\left(\begin{array}{c} 0 \\ 0 \\ \lambda^T(\frac{\partial}{\partial \theta}R)\dot{\hat{q}} \end{array}\right). \tag{14}$$

By direct integration, $\lambda_1 = k_1$ and $\lambda_2 = k_2$. Substitute these values back into the definition for $\dot{\lambda}_3$:

$$\dot{\lambda}_3 = k_1(s\dot{\hat{x}} + c\dot{\hat{y}}) - k_2(c\dot{\hat{x}} - s\dot{\hat{y}}). \tag{15}$$

From equation 12,

$$\dot{x} = c\dot{\hat{x}} - s\dot{\hat{y}} \tag{16}$$

$$\dot{y} = s\dot{\hat{x}} + c\dot{\hat{y}}. \tag{17}$$

Substitute into equation 15,

$$\dot{\lambda}_3 = k_1\dot{y} - k_2\dot{x}, \tag{18}$$

and integrate:

$$\lambda_3 = k_1 y - k_2 x + k_3. \tag{19}$$

■

B. Geometric interpretation of the adjoint

We can write out the Hamiltonian for rigid bodies as a function of the state and the time derivative of the state:

$$H = k_1\dot{x} + k_2\dot{y} + \dot{\theta}(k_1 y - k_2 x + k_3). \tag{20}$$

We will assume for the moment that $k_1^2 + k_2^2 > 0$. Since the Maximum Principle requires us to choose minimizing controls for the Hamiltonian, uniform positive scaling of the constants does not affect the choice of controls; without loss of generality we assume that $k_1^2 + k_2^2 = 1$.

Define the *control line* to be

$$\mathcal{L} = \{(a,b) \in \mathbf{R}^2 : k_1 b - k_2 a + k_3 = 0\},$$

The locus of points (a,b) that satisfy $k_1 b - k_2 a + k_3 = 0$ is a line. The term $(k_1\dot{x} + k_2\dot{y})$ in the Hamiltonian is the component of the velocity vector that lies along this line, and the term $k_1 y - k_2 x + k_3$ is the distance of the point (x,y) from this line.

If $k_1^2 + k_2^2 = 0$, the line can be considered to be 'at infinity', and the robot simply chooses controls to maximize or minimize angular velocity.

The state of the robot and the values of k_1, k_2, and k_3 determine the controls. Therefore, the problem of finding the optimal trajectory from start state q_0 to goal state q_1 optimally is reduced to finding values for k_1, k_2, and k_3 that take the robot to the goal optimally.

Computing the values of the constants directly (or equivalently, finding the location of the control line) for a given start and goal is difficult. First, however, we will consider the nature of trajectories assuming an arbitrary placement of the control line. In this case it is convenient to attach the world coordinate frame to the control line. In this case, the expression for the Hamiltonian simplifies to

$$H = \dot{x} + y\dot{\theta}. \tag{21}$$

C. Switching functions

Since there are independent bounds on the speeds of the wheels, the Hamiltonian can usually be maximized by driving the wheels at full speed either in the forwards or backwards direction.

Theorem 3: For any time-optimal trajectory, there exist constants k_1, k_2, and k_3, with $k_1^2+k_2^2+k_3^2 \neq 0$, and functions φ_1, φ_2, and φ_3 defined by

$$\begin{pmatrix}\varphi_1 & \varphi_2 & \varphi_3\end{pmatrix} = \lambda^T R_\theta A^{-1}. \tag{22}$$

such that the vehicle follows the control law

$$u_i(t) \in \begin{cases} \{h_i\} & \text{if} \quad \varphi_i(t) < 0 \\ [l_i, h_i] & \text{if} \quad \varphi_i(t) = 0 \\ \{l_i\} & \text{if} \quad \varphi_i(t) > 0. \end{cases} \tag{23}$$

Furthermore, the quantity λ_0 defined by

$$\lambda_0 = -\begin{pmatrix}\varphi_1 & \varphi_2 & \varphi_3\end{pmatrix} u \tag{24}$$

is constant along the trajectory.

Proof: We can rewrite the Hamiltonian

$$H = \lambda^T R_\theta A^{-1} u. \tag{25}$$

Application of the Maximum Principle completes the proof. ■

We call the functions φ_1, φ_2, and φ_3 *switching functions*, since the controls switch when these functions change sign. Explicitly, the switching functions may be written

$$\begin{aligned}\varphi_j = \;& k_1(a_{3j}y + a_{1j}\cos\theta - a_{2j}\sin\theta) \\ & - k_2(a_{3j}x - a_{1j}\sin\theta - a_{2j}\cos\theta) + k_3 a_{3j}.\end{aligned} \tag{26}$$

where a_{ij} is the (i, j) element of the design matrix A^{-1}.

D. Geometric interpretation

Theorem 4: There exist three 'switching points', $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$, rigidly attached to the robot, such that the signed distance of $\mathbf{s}_i$ from the control line $\mathcal{L}$ is equal to the switching function φ_i, up to scaling by the constant a_{3i}. The switching points fall at the intersection of the wheel lines, but may be 'at infinity' if the lines do not intersect.

Proof: Take $a_{3i} \neq 0$. Assume the coordinates of the point $\mathbf{s}_i$ in the robot frame are given by

$$\mathbf{s}_i = \left(-\frac{a_{2i}}{a_{3i}}, \frac{a_{1i}}{a_{3i}}\right). \tag{27}$$

In the world coordinate frame, the location of $\mathbf{s}_i$ is given by

$${}^w\mathbf{s}_i = \begin{pmatrix} x \\ y \end{pmatrix} + R(\theta)\mathbf{s}_i. \tag{28}$$

Define

$$\eta(x, y) = k_1 y - k_2 x + k_3 \tag{29}$$

to be the signed, scaled distance of the point (x, y) from the line $\mathcal{L}$. Direct computation shows that

$$a_{3i}\eta({}^w\mathbf{s}_i) = \varphi_i. \tag{30}$$

To show that the switching points lie at the intersection of the wheel lines, without loss of generality we place the robot frame at one of the intersections, as described in section II. We then use the expression for A^{-1} given by equation 5 to compute the locations for the switching points $\mathbf{s}$.

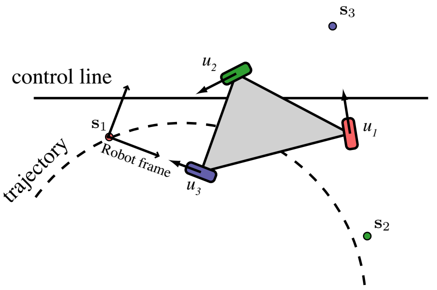


Fig. 4: The switching points' position relative to the control line determine the optimal control and the robot's trajectory.

If $a_{3i} = 0$, then switching point i is 'at infinity'. To determine when the control corresponding to switching function φ_i switches, we consider the angle that the vector (a_{1i}, a_{2i}) makes with the control line; details are left to the reader. ■

Figure 4 shows the geometric interpretation of the optimal control law. In this example, the third switching point is above the control line, causing the third wheel to turn in reverse. The other two switching points are below, so u_1 and u_2 are maximal. This causes the robot to go on a circle. When this motion causes a switching point to cross the control line, the corresponding control will change.

Notice that we can choose a convention for the signs on the controls so that a_{3i} is non-negative for all i. We therefore can drop the constant a_{3i} from the computation of the switching function, as long as we are careful about what happens when $a_{3i} = 0$.

If we align the world coordinate frame with the control line, the form of equation 28 is further simplified. Let (r_i, α_i) be the polar coordinates of switching point $\mathbf{s}_i$ in the robot frame. Then the switching function φ_i is

$$\varphi_i = y + r_i \sin(\theta + \alpha_i). \tag{31}$$

E. Optimal segments

The previous section gave a control law for optimal trajectories, as well as a geometric interpretation. We now turn to examine the implications of such a control law for the shape of optimal trajectories.

Theorem 5: If the trajectory of a bounded-velocity rigid body in the plane, with independent control constraints, is singular on a non-null time interval, then the rigid body is translating in the control line's direction.

Proof: We will prove that θ is constant on such an interval. Assume that switching function φ_j is null at some moment. Then equation 31 indicates that y is a linear combination of $\sin\theta$ and $\cos\theta$, thus all three switching functions are linear combinations of these as well. Consider the time sub-interval before any other switching function becomes 0. Since the controls corresponding to the non-null switching functions are constant on this sub-interval, equation 24 indicates that the Hamiltonian becomes a linear combination of $\sin\theta$ and $\cos\theta$ as well:

$$H = k_1 \sin\theta + k_2 \cos\theta \tag{32}$$

where k_1 and k_2 are constants. Since the Hamiltonian cannot be zero, at least one of these constants is not null. Then let α be the direction of the (k_1, k_2) vector:

$$\frac{H}{||(k_1, k_2)||} = \sin(\theta + \alpha) \tag{33}$$

This equation has distinct roots, while θ is a continuous function of time. Therefore θ must be constant on the sub-interval considered. Furthermore, equation 31 implies that y is also constant, thus keeping all the switching functions constant on the sub-interval, so none of them can switch from a non-null to a null value. Therefore the sub-interval considered is identical to the whole singular interval. ■

Theorem 6: The time-optimal trajectories for any wheeled mobile robot in the unobstructed plane, with independent bounds on the speeds of the wheels, are composed only of arcs of circles and straight lines.

Proof: This follows directly from theorems 1 and 5 and equation 23. ■

IV. Optimal trajectory classes

The previous section has identified the kinds of segments that optimal trajectories are composed of. We now proceed to a closer examination of the control switches, and the possible sequences in which they can occur. We will see that optimal trajectories can be grouped into well separated classes, according to these sequences, and we develop a method of enumerating these classes.

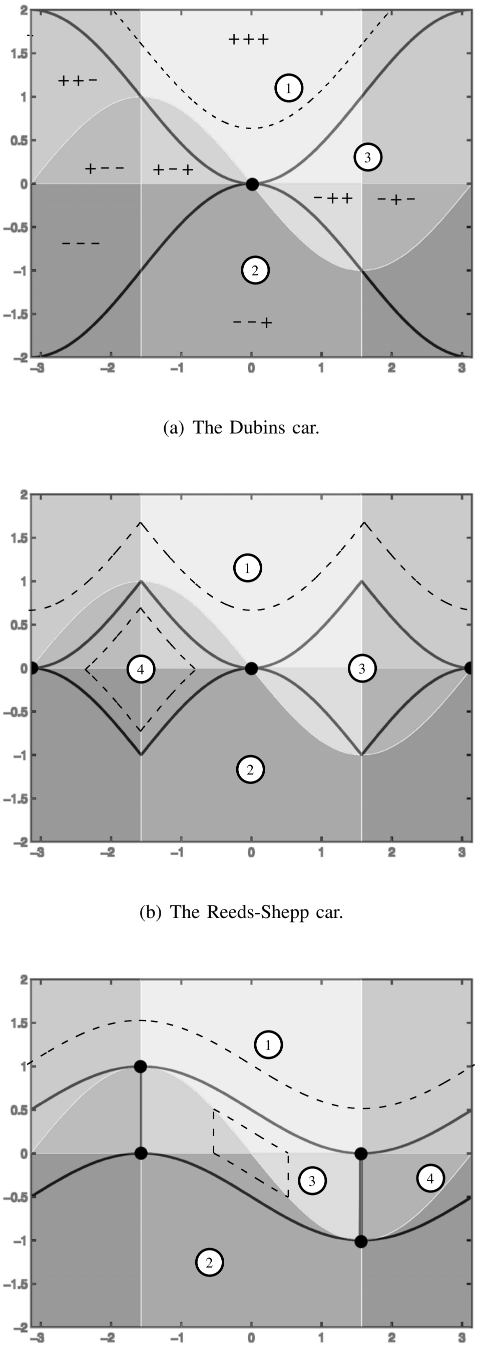
A. Switching space

We can enumerate all optimal trajectories for a given design by looking at all the possible configurations that the robot can have with respect to the control line. Only the robot's distance and orientation relative to the control line determine its optimal controls. (See equation 31.) It makes sense, therefore, to project these trajectories onto their y and θ dimensions, which we will call the *switching space*.

Given a configuration in switching space, we can determine the optimal controls by referring to the three curves described by setting the switching functions to equal 0.

Together, these sinusoidal curves divide switching space into 8 control regions, each characterized by its own set of extremal controls (see fig. 5(a) for an example). For example, the so-called - + + region is composed of all the points which are below the first switching curve, but above the other two. All robot configurations that fall into this region will share optimal controls that set u_1 to its maximum bound, and u_2 and u_3 to their minimum bounds.

Each optimal trajectory then appears as a continuous curve in switching space, a curve that changes its controls (and general shape) when it crosses a boundary between control regions. The crossings are usually instantaneous, with only singular trajectories being able to dwell on the boundaries.



(a) The Dubins car.

(b) The Reeds-Shepp car.

(c) The differential drive.

Fig. 5: Separating the optimal trajectories for non-holonomic orthogonal vehicles in switching space. The shaded areas are control regions. The continous lines are tangential trajectories, with the dots representing the points where translation along the control line is possible. The dotted lines show examples of non-tangential trajectories, with the numbers representing the non-tangential trajectory classes.

As shown in theorem 6 above, optimal trajectories are composed of segments of lines and circles. Translation will show up in switching space as either a point (if it is parallel to the control line) or a vertical line. Rotation projects to a sinusoid. Each of these curves also needs to be intersected with the control region that makes it possible.

B. λ_0 *level curves*

Theorem 3 indicates that each optimal trajectory is characterized by its λ_0 quantity. Given λ_0, it is possible to use the above method and identify all 8 optimal curve segments which correspond to it. By examining how they connect (through visual inspection, or a simple graph search), we determine the structure of all optimal trajectories corresponding to this λ_0 value. Together, the obtained optimal segments form a λ_0 *level curve*.

Each λ_0 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 continously, with the possibility of periodicity as well; see figure 5(b) for two examples. We can distinguish the *optimal classes* appearing on a level curve by first separating the unconnected regions, and then picking start and end regions on each. For instance, suppose a level curve contains two continuous components. One goes through the - - - and - - + control regions, while the other alternates between + + + and + + -. This gives us two optimal classes, each of which contains many kinds of trajectory structures (such as partial segment - - -, full segment - - +, full segment - - -, partial segment - - +).

C. *Tangential level curves*

Small changes in the robot's orientation and distance to the control line will not change the control order (and thus the trajectory class), unless one of the circles that switching points follow around the rotation centers is tangent to the control line; we call trajectories where such a situation occurs *tangentials*. Drawing out the tangential level curves therefore separates the switching space into regions where all the level curves belong to the same class.

A particular category of tangentials are *singular trajectories*, where it is possible to keep translating along the control line at the tangency point.In contrast to the region-bound classes, those containing singular trajectories are made up of parts of a single tangential level curve. Theorem 5, which indicates that translations have to be parallel to the control line, makes it possible to determine the points along a tangential curve (if any) at which singular segments of arbitrary length can occur.

The λ_0 values for the tangentials can be calculated for each (switching point, circle center) pair from equation 21, which becomes

$$\lambda_0 = d_{SP,IC}\ \omega_{IC} \tag{34}$$

where $d_{SP,IC}$ is the distance between the switching point and the circle center, while ω_{IC} is the angular velocity around this center. Using the robot design and control bounds, these can be easily calculated from formula 7 and equation 5.

D. *Separation algorithm*

The process of finding the optimal trajectory classes can be automated. We have developed and implemented an algorithm that, given a robot design, produces a graph of all its tangential level curves, thus allowing us to visually determine the structure of all trajectory classes for the robot. While the space limitations constrain us to skip a detailed exposition, we present below a top-level sketch of its main procedures. All the iterations are over constant quantities, such as the switching points or the tangential λ_0 values; the algorithm runs in $O(1)$ time.

```
Separation-Algorithm($design$)
1) Calculate list of tangential $\lambda_0$ values
2) For each value: Plot-level-curve($\lambda_0, design$)

Plot-level-curve($\lambda_0, design$)
1) Plot-level-segment($\lambda_0, design, -, -, -$)
2) Plot-level-segment($\lambda_0, design, -, -, +$)
3) ...

Plot-level-segment($\lambda_0, design, controls$)
1) If Angular-Velocity($design, controls$) $= 0$, then
   Plot-Translation($\lambda_0, design, controls$)
2) Else Plot-Rotation($\lambda_0, design, controls$)

Plot-Rotation($\lambda_0, design, controls$)
1) For each switching point, calculate the angular intervals
   that keep it in a position, relative to the control line, that
   is consistent with its corresponding control
2) Intersect these intervals among switching points
3) Plot the rotation in switching space, as a sinusoid
   restricted to the intervals obtained above

Plot-Translation($\lambda_0, design, controls$)
1) $\theta \leftarrow \mathrm{acos}(\frac{\lambda_0}{\text{Plane-Velocity}(design, controls)})$
2) For each switching point, calculate the $y$ intervals that
   keep it in a position, relative to the control line, that is
   consistent with its corresponding control and the robot's
   orientation
3) Intersect these intervals among switching points
4) Plot the rotation in switching space, as two vertical lines
   restricted to the intervals obtained above, one at $\theta$ and
   the other at $2\pi - \theta$
```

V. Some examples

Three well studied vehicles are the Dubins and Reeds-Shepp cars and the differential drive. Figure 2 has shown how to model these with omniwheels, and also how to obtain their switching points by drawing out and intersecting the wheel lines. Since the "middle" wheel can be moved along its line (as discussed in II), these three vehicles share the same design class, and only differ in their control bounds. The common switching triangle is degenerate (two wheel lines being parallel), with two switching points falling on the wheels. The third switching point (only relevant for the Reeds-Shepp car) is "at infinity".

The eight control regions determined by the switching curves are indicated as shadings in figure 5. This figure shows the output of the algorithm described above, thus indicating all the optimal trajectory classes. For the Dubins car, the tangential level curves divide switching space into three regions, each corresponding to a class. One is on top; the optimal trajectories are similar in shape to the region's boundary, can be periodic, and cross three control regions: + + - , + - + and + + + respectively. Since only the u_1 control can switch in the Dubins car, this corresponds to a constantly right-turning trajectory, where u_1 is set to its lowest bound. The analysis of the bottom region is similar, and results in a constantly left-turning trajectory. The middle region (which wraps around in the cylindrical space) has level curves which are similar in shape to its boundary, and go in succession through all the control regions: + + -, + - -, + - +, - - +, - - -, - + -, - + -, + + + . This effectively results in a succession of left- and right-turns (*scrL r*, in Reeds and Shepp's notation).

Given the vehicle's design, singular trajectories are only possible by setting u_1 to 0 at the point where y and θ are 0 as well. The tangential level curve indicated as a continuous line on the figure constitutes by itself a fourth, singular, class, containing, in addition to the left- and right-turns generated by the control regions which it passes through, line segments of arbitrary length at the (0, 0) point in the switching space. Optimal trajectories belonging to this class can therefore consist of any succession of maximum left- and right-turns and straight driving (e.g. *scrL s r*), with all the straight segments colinear.

Let us now move to a somewhat more complicated example, the Reeds-Shepp car (see figure 5(b)). The switching space is divided into 4 areas by the single tangential level curve. With u_2 fixed, the upper region corresponds to keeping u_1 minimized and alternating u_3 between its bounds. This results in a (forward-right, reverse-right) (*r^+ r^-*)) class. Analogously, the lower region corresponds to a (forward-left, reverse-left) (*scrL$^+$ scrL$^-$*) class. The right middle area indicates the succession + + +, + + -, - + -, - + +, corresponding to a (reverse-right, forward-right, forward-left, reverse-left) class, while the left middle area gives us (forward-left, forward-right, reverse-right, reverse-left).

In addition to the 4 non-singular classes, there also exists one singular class, containing the tangential curve itself. The design of the robot makes possible translations along the control line at two points, (0, 0) and (π, 0) (identical to ($-\pi$, 0). One example of such a singular trajectory is (forward-left, forward-straight, forward-left, reverse-left, reverse-straight, reverse-right).

The 5 optimal classes for the differential drive (fig. 5(c)) are left as an exercise to the reader.

We have also analyzed some non-orthogonal designs, such as a hypothetical robot with a switching triangle that has $\frac{\pi}{3}$ and $\frac{\pi}{4}$ angles. Its switching space features 6 tangential level curves separating the switching space into 5 singular and 18 non-singular classes. Allowing asymmetric control bounds increases complexity even further.

VI. Open problems and future work

While we are able to determine the optimal trajectory classes for any kinematic model of an omniwheel-specified design, the problem of determining the optimal trajectory given specific start and end configurations remains open. We have made some progress towards determining the location of the control line; specifically we can find a small finite set of points which contains a point on this line, and we are able to determine its exact location for tangential trajectories. We are working towards determining its location in the general case.

References

[1] 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.

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

[3] 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.

[4] Tamás Kalmár-Nagy, Raffaello D'Andrea, and Pritam Ganguly. Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, 46:47–64, 2004.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.