

Geometric Construction of Time Optimal Trajectories for Differential Drive Robots

Devin J. Balkcom, *Carnegie Mellon University, Pittsburgh PA 15213*
 Matthew T. Mason, *Carnegie Mellon University, Pittsburgh PA 15213*

Abstract

We consider a differential drive mobile robot: two unsteered coaxial wheels are independently actuated. Each wheel has bounded velocity, but no bound on torque or acceleration. Pontryagin's Maximum Principle gives an elegant description of the *extremal* trajectories, which are a superset of the time optimal trajectories. Further analysis gives an enumeration of the time optimal trajectories, and methods for identifying the time optimal trajectories between any two configurations. This paper recapitulates and refines the results of [1] and [2] and presents a simple graphical technique for constructing time optimal trajectories.

1 Introduction

This paper addresses the time optimal paths for differential drive mobile robots with bounded velocity. Differential drive (or diff drive) means there are two unsteered independently actuated coaxial wheels. Bounded velocity means that each wheel is independently bounded in velocity, but acceleration is not bounded. Even discontinuities in wheel velocity are permitted. The environment is planar and free of obstacles.

Under these assumptions, we will see that the time optimal paths are composed of straight lines alternating with turns “in place”, i.e. turns about the center of the robot. Optimal paths contain at most three straights and two turns. There are a number of other restrictions, leading to a set of 40 different combinations arranged in 9 different symmetry classes. The simplest nontrivial motions are turn-straight-turn motions: turn to face the goal (or away from the goal); roll straight forward (or backward) to the goal; turn to the goal orientation. In some instances the optimal path passes through an intermediate “via” point. See Figure 1 for example motions from seven of the nine classes.

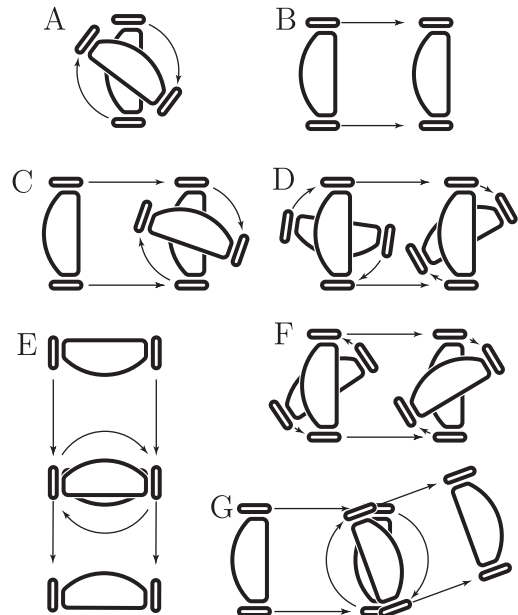


Figure 1: The seven simplest optimal trajectory classes.

To derive the optimal paths we will use Pontryagin's maximum principle to obtain a geometric program for the *extremal* trajectories, which are a superset of the optimal trajectories. We then derive some additional necessary conditions, leading to a complete enumeration of optimal trajectories and a planning algorithm. Finally we reformulate the analysis to give a more intuitive geometric procedure for constructing optimal trajectories.

Previous Work

This paper expands on the results presented in [1] and [2]. Other work on diff drive robots has assumed bounds on acceleration rather than on velocity; for example see papers by Reister and Pin [6] and Renaud and Fourquet [7]. For the bounded acceleration model, the time optimal trajectories have been found numerically, and there is current work to

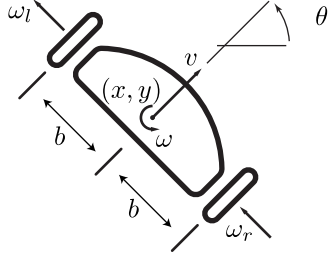


Figure 2: Notation

find a closed form solution. The bounded velocity model is simpler, and the structure and cost of the fastest trajectories can be determined analytically.

Most of the work on time optimal control with bounded velocity models has focused on steered vehicles rather than diff drives, originating with papers by Dubins [3] and by Reeds and Shepp [5]. Many of the techniques employed here are an extension of optimal control techniques developed for steered vehicles in [8, 9, 10].

2 Assumptions, definitions, notation

The state of the robot is $q = (x, y, \theta)$, where the robot reference point (x, y) is centered between the wheels, and the robot direction θ is 0 when the robot is facing along the x -axis (Figure 2). The robot's velocity in the forward direction is v and its angular velocity is ω . The robot's width is $2b$. The wheel angular velocities are ω_l and ω_r . With suitable choices of units we obtain

$$v = \frac{1}{2}(\omega_l + \omega_r) \quad (1)$$

$$\omega = \frac{1}{2b}(\omega_r - \omega_l) \quad (2)$$

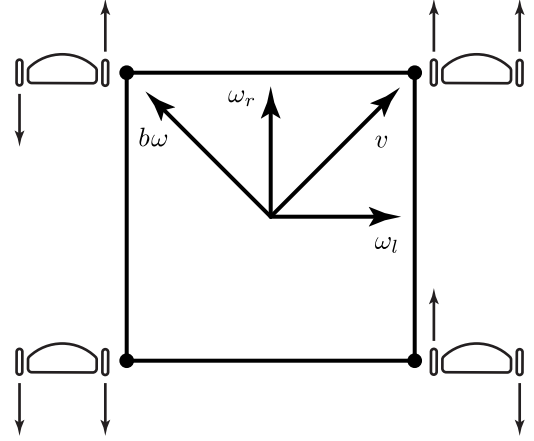
and

$$\omega_l = v - b\omega \quad (3)$$

$$\omega_r = v + b\omega \quad (4)$$

The robot is a system with control input $w(t) = (\omega_l(t), \omega_r(t))$ and output $q(t)$. Admissible controls are bounded Lebesgue measurable functions from time interval $[0, T]$ to the closed box $W = [-1, 1] \times [-1, 1]$ (see Figure 3).

It follows immediately that $v(t)$ and $\omega(t)$ are measurable functions defined on the same interval. Given initial condi-

Figure 3: Bounds on (ω_l, ω_r)

tions $q_s = (x_s, y_s, \theta_s)$ the path of the robot is given by

$$x(t) = x_s + \int_0^t v \cos(\theta) \quad (5)$$

$$y(t) = y_s + \int_0^t v \sin(\theta) \quad (6)$$

$$\theta(t) = \theta_s + \int_0^t \omega \quad (7)$$

We also define rectified path length in E^2 , the plane of robot positions:

$$s(t) = \int_0^t |v| \quad (8)$$

and rectified arc length in S^1 , the circle of robot orientations:

$$\sigma(t) = \int_0^t |\omega| \quad (9)$$

It follows that θ , x , y , s , and σ are continuous, that their time derivatives exist almost everywhere, and that

$$\dot{\theta} = \omega \quad \text{a.e.} \quad (10)$$

$$\dot{x} = v \cos(\theta) \quad \text{a.e.} \quad (11)$$

$$\dot{y} = v \sin(\theta) \quad \text{a.e.} \quad (12)$$

The admissible control region W also provides a convenient comparison with previously studied bounded velocity models. If we plot W in v - ω space, we obtain a diamond shape. Steered vehicles are typically modeled as having a bound on the steering ratio $\omega : v$, and on the velocity v (Figure 4).

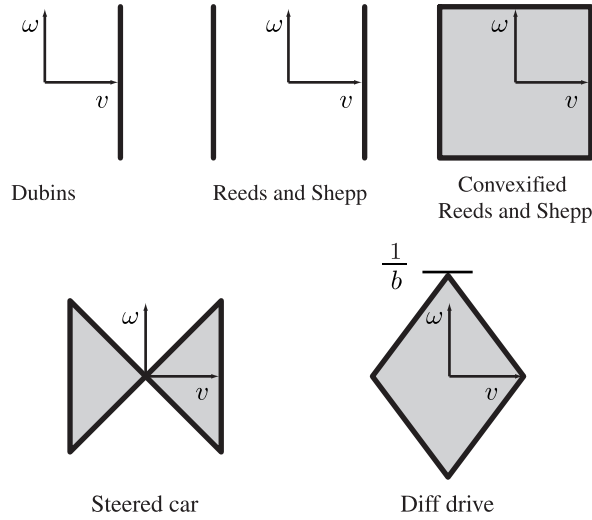


Figure 4: Bounded velocity models of mobile robots

We also need a notation for trajectories. Later sections show that extremal trajectories are composed of straight lines and turns about the robot's center. We will represent *forward* by \uparrow , *backward* by \downarrow , *left turn* by \curvearrowleft , and *right turn* by \curvearrowright . Thus the trajectory $\curvearrowleft\uparrow\curvearrowright$ can be read "left forward right". When necessary, a subscript will indicate the distance or angle traveled. We will use \mathfrak{t} and \mathfrak{s} to represent turns and straights of indeterminate direction.

3 Pontryagin's Maximum Principle. Extremal controls.

The existence of a time optimal trajectory for every pair of start and goal configurations is proven in [1]. This section uses Pontryagin's Maximum Principle [4] to derive necessary conditions for time optimal trajectories of the bounded velocity diff drive robot.

The robot system is described by

$$\dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\omega_l + \omega_r) \cos(\theta) \\ \frac{1}{2}(\omega_l + \omega_r) \sin(\theta) \\ \frac{1}{2b}(\omega_r - \omega_l) \end{pmatrix} \quad (13)$$

where our input is

$$w = \begin{pmatrix} \omega_l \\ \omega_r \end{pmatrix} \in W$$

Equation 13 can be rewritten:

$$\dot{q} = \omega_l f_l + \omega_r f_r \quad (14)$$

where f_l and f_r are the vector fields corresponding to the left and right wheels:

$$f_l = \begin{pmatrix} \frac{1}{2} \cos \theta \\ \frac{1}{2} \sin \theta \\ -\frac{1}{2b} \end{pmatrix} \quad (15)$$

$$f_r = \begin{pmatrix} \frac{1}{2} \cos \theta \\ \frac{1}{2} \sin \theta \\ \frac{1}{2b} \end{pmatrix} \quad (16)$$

Vector field f_l corresponds to turning about a center located under the right wheel, and f_r corresponds to turning about a center located under the left wheel.

Define λ to be an R^3 -valued function of time called the *adjoint vector*:

$$\lambda(t) = \begin{pmatrix} \lambda_1(t) \\ \lambda_2(t) \\ \lambda_3(t) \end{pmatrix}$$

Let $H : R^3 \times SE^2 \times W \rightarrow R$ be the *Hamiltonian*:

$$H(\lambda, q, w) = \langle \lambda, \omega_l f_l + \omega_r f_r \rangle$$

The maximum principle states that for a control $w(t)$ to be optimal, it is *necessary* that there exist a nontrivial (not identically zero) adjoint vector $\lambda(t)$ satisfying the *adjoint equation*:

$$\dot{\lambda} = -\frac{\partial}{\partial q} H \quad (17)$$

while the control $w(t)$ minimizes the Hamiltonian at every t :

$$H(\lambda, q, w) = \min_{z \in W} H(\lambda, q, z) = \lambda_0. \quad (18)$$

with $\lambda_0 \geq 0$. Equation 17 is called the *adjoint equation* and Equation 18 is called the *minimization equation*.

For the bounded velocity diff drive, the adjoint equation gives

$$\dot{\lambda} = -\frac{\partial}{\partial q} \langle \lambda, \omega_l f_l + \omega_r f_r \rangle \quad (19)$$

$$= \frac{\omega_l + \omega_r}{2} \begin{pmatrix} 0 \\ 0 \\ \lambda_1 \sin \theta - \lambda_2 \cos \theta \end{pmatrix} \quad (20)$$

Fortunately these equations can be integrated to obtain an expression for the adjoint vector. First we observe that λ_1

and λ_2 are constant and define c_1 and c_2 accordingly

$$\lambda_1(t) = c_1 \quad (21)$$

$$\lambda_2(t) = c_2 \quad (22)$$

For λ_3 we have the equation

$$\dot{\lambda}_3 = \frac{\omega_l + \omega_r}{2}(\lambda_1 \sin \theta - \lambda_2 \cos \theta) \quad (23)$$

But we can substitute from Equations 1, 11, and 12 to obtain

$$\dot{\lambda}_3 = c_1 \dot{y} - c_2 \dot{x} \quad (24)$$

which is integrated to obtain the solution for λ_3 :

$$\lambda_3 = c_1 y - c_2 x + c_3 \quad (25)$$

where c_3 is our third and final integration constant. It will be convenient in the rest of the paper to define a function η of x and y :

$$\eta(x, y) = c_1 y - c_2 x + c_3 \quad (26)$$

So then the adjoint equation is satisfied by

$$\lambda = \begin{pmatrix} c_1 \\ c_2 \\ \eta(x, y) \end{pmatrix} \quad (27)$$

for any c_1, c_2, c_3 not all equal to zero.

Let the η -line be the line of points (x, y) satisfying $\eta(x, y) = 0$, and note that $\eta(x, y)$ gives a scaled directed distance of a point (x, y) from the η -line. Let the *right half plane* be the points satisfying

$$\eta(x, y) > 0 \quad (28)$$

and let the *left half plane* be the points satisfying

$$\eta(x, y) < 0 \quad (29)$$

We also define a direction for the η -line consistent with the choice of “left” and “right” for the half planes.

The minimization equation 18 can be rewritten

$$\omega_l \phi_l + \omega_r \phi_r = \min_{z_l, z_r} z_l \phi_l + z_r \phi_r \quad (30)$$

where ϕ_l and ϕ_r are defined to be the two *switching functions*:

$$\phi_l = \langle \lambda, f_l \rangle \quad (31)$$

$$= \begin{pmatrix} c_1 \\ c_2 \\ \eta(x, y, \theta) \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{2} \cos \theta \\ \frac{1}{2} \sin \theta \\ -\frac{1}{2b} \end{pmatrix} \quad (32)$$

$$= -\frac{1}{2b} \eta(x + b \sin \theta, y - b \cos \theta) \quad (33)$$

$$\phi_r = \langle \lambda, f_r \rangle \quad (34)$$

$$= \begin{pmatrix} c_1 \\ c_2 \\ \eta(x, y, \theta) \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{2} \cos \theta \\ \frac{1}{2} \sin \theta \\ \frac{1}{2b} \end{pmatrix} \quad (35)$$

$$= \frac{1}{2b} \eta(x - b \sin \theta, y + b \cos \theta) \quad (36)$$

Note that the wheels' coordinates can be written

$$\begin{pmatrix} x_l \\ y_l \end{pmatrix} = \begin{pmatrix} x - b \sin \theta \\ y + b \cos \theta \end{pmatrix} \quad (37)$$

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} x + b \sin \theta \\ y - b \cos \theta \end{pmatrix} \quad (38)$$

so the switching functions can be written

$$\phi_l = -\frac{1}{2b} \eta(x_r, y_r) \quad (39)$$

$$\phi_r = \frac{1}{2b} \eta(x_l, y_l) \quad (40)$$

Now the minimization equation says that if the controls ω_l, ω_r are optimal then they minimize the Hamiltonian $H = \omega_l \phi_l + \omega_r \phi_r$. This implies the optimal controls can be expressed

$$\omega_l \begin{cases} = 1 & \text{if right wheel} \in \text{right half plane} \\ \in [-1, 1] & \text{if right wheel} \in \eta\text{-line} \\ = -1 & \text{if right wheel} \in \text{left half plane} \end{cases} \quad (41)$$

$$\omega_r \begin{cases} = 1 & \text{if left wheel} \in \text{left half plane} \\ \in [-1, 1] & \text{if left wheel} \in \eta\text{-line} \\ = -1 & \text{if left wheel} \in \text{right half plane} \end{cases} \quad (42)$$

If $c_1 = c_2 = 0$, then the the entire plane is the left half plane or the right half plane, depending on the sign of c_3 . (Recall that all three integration constants cannot be simultaneously zero.)

The location of the η -line depends on the apparently arbitrary integration constants. The maximum principle does not give the location of the line; it merely says that if we have an optimal control then the line exists and the optimal control must conform to the equations above. The question that naturally arises is how to locate the line properly, given the start and goal configurations of the robot. There seems

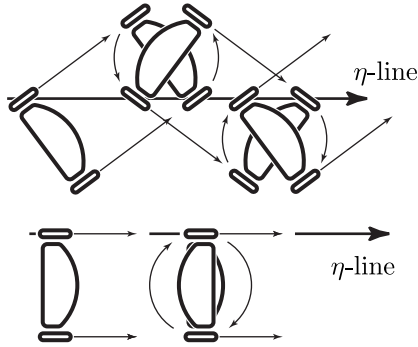


Figure 5: Two extremals: zigzag right and tangent CW. Other extremal types are zigzag left, tangent CCW, and turning in place: CW and CCW. Straight lines are special cases of zigzags or tangents.

to be no direct way of doing so. Rather, we must use other means to identify the extremal trajectory.

The robot switches only when a wheel touches the η -line, so the possibilities can be enumerated by constructing a circle whose diameter is the robot wheelbase, and considering the different possible relations of this circle to the η -line:

- CCW and CW: If the robot is in the left half plane and out of reach of the η -line, it turns in the counter-clockwise direction (CCW). CW is similar.
- TCCW and TCW (Tangent CCW and Tangent CW). If the robot is in the left half plane, but close enough that a circumscribed circle is tangent to the η -line, then the robot may either roll straight along the line, or it may turn through any positive multiple of π . TCW is similar.
- ZR and ZL: If the circumscribed circle crosses the η -line, then a zigzag behavior occurs. The robot rolls straight in the η -line's direction until one wheel crosses. It then turns until the other wheel crosses, and then goes straight again. There are two non-degenerate patterns: $\dots \uparrow \curvearrowright \downarrow \curvearrowleft \dots$ called *zigzag right* ZR, and $\dots \uparrow \curvearrowleft \downarrow \curvearrowright \dots$ called *zigzag left* ZL. (Recall that *forward* and *backwards* actions are denoted by \uparrow and \downarrow , and spins in place are denoted by \curvearrowright and \curvearrowleft .)

Figure 5 shows examples of ZR and TCW.

Examining these classes, we see that

Theorem 1 *For an optimal trajectory,*

$$t = s(t) + b\sigma(t) \quad (43)$$

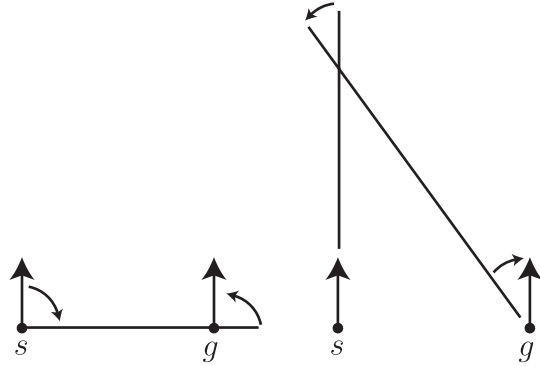


Figure 6: Turn-straight-turn is not always optimal.

Proof: Extremal trajectories are composed only of turns and straight lines. \square

Note that in [2], we demonstrate that equation 43 actually holds for any trajectory such that $\max(|\omega_l|, |\omega_r|) = 1$ for almost all t ; i.e., for trajectories in which one control is always saturated. This may provide some intuition for why turns and straights are faster than curves.

The fastest trajectories for a Reeds and Shepp car with zero turning radius are of the form tst . It might at first seem that Equation 43 implies the same is true for the diff drive. Trajectories of type tst certainly minimize translation time; however, they do not necessarily minimize rotation time. Figure 3 shows an example: the robot is at the origin facing north, and the goal is some distance to the east, facing north. The robot could follow a $\curvearrowright \uparrow \curvearrowleft$ trajectory, with a total turning angle of π . However, there is a $\uparrow \curvearrowleft \downarrow \curvearrowright$ trajectory that turns through a total angle of less than π . For a wide enough robot (b large enough), the second trajectory will be optimal.

4 Further conditions for optimality

The previous section showed that every time optimal path must be of type CCW, CW, TCCW, TCW, ZR, or ZL. However, the converse is definitely not true—not every such trajectory is optimal. For example, a robot turning in place for several revolutions is not time optimal. To keep the distinction clear, we refer to trajectories satisfying Pontryagin's Maximum Principle as *extremal*, and we note that the time-optimal trajectories are a subset of the extremal trajectories. In this section we find additional necessary conditions, ultimately finding that no time optimal path can have more than three straights and two turns.

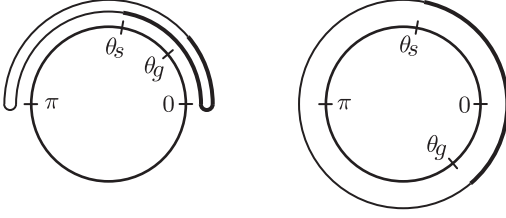


Figure 7: There is always a roundtrip of length 2π visiting θ_s , θ_g , 0, and π .

Necessary conditions on TCCW and TCW trajectories

Theorem 2 For every time-optimal trajectory $\sigma(T) \leq \pi$.

Proof: Consider the fastest tst trajectory between a given start and goal. Obviously the straight action connects the start to the goal. Suppose we orient our coordinates so the angle from start to goal is 0. The robot's heading during the straight is either 0 or π . To plan the turns, we must consider different paths on the circle from θ_s to θ_g , passing through either 0 or π along the way. Note that in every instance there is a round trip from θ_s to 0, π , θ_g , and back to θ_s , of length 2π (Figure 7). Hence there is a one-way trip of length π or less. \square

Theorem 3 Tangent trajectories containing more than three actions are not optimal.

Proof: An extremal of type TCW or TCCW alternates turns and straights. Any full untruncated turn must be a multiple of π . If there are four actions, there is at least one untruncated turn of length at least π , and a second turn of nonzero length. The rectified arc length σ would be more than π , contradicting Theorem 2. \square

Necessary conditions on ZR and ZL trajectories

Zigzag trajectories are composed of alternating turn or straight line actions. Successive turns or straights must be in opposite directions, but have the same magnitude if untruncated. Simple geometry also gives a relationship between ϕ , the angle of each turn, and d , the length of each straight. We have:

$$d = 2b \tan\left(\frac{\phi}{2}\right) \quad (44)$$

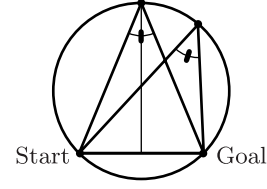


Figure 8: Zigzags of three turns are not optimal

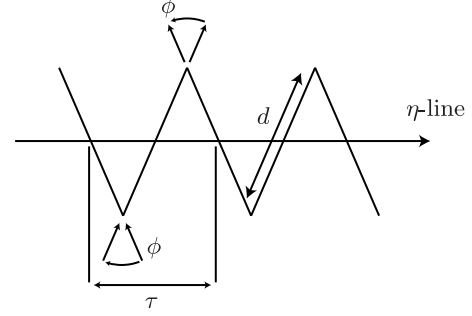


Figure 9: Periodicity of a zigzag

Theorem 4 Zigzag subsections containing three turns are not optimal.

Proof: Consider a zigzag subsection with three turns, and two straights. The straights are the same length, so the path comprises two legs of an isosceles triangle. Construct the circle containing the start, the goal, and the via point as in Figure 8. If we perturb the via point to a nearby point on the same circle the turning time is unchanged, and the translation is decreased. \square

Zigzags can also be said to be periodic. Let τ be the smallest positive time such that:

$$\begin{aligned} \theta(t) &= \theta(t + \tau) \\ \eta(x(t), y(t)) &= \eta(x(t + \tau), y(t + \tau)) \end{aligned}$$

Theorem 5 A zigzag trajectory of more than one period is not optimal.

Proof: Consider a zigzag of more than one period, beginning at time 0 and ending at time $T > \tau$. By theorem 4, the zigzag is not optimal if $\sigma(T) > 2\phi$. If $s(T) > 2d$, then there are three straights. The first and last straights are parallel. If we reorder the actions to perform these consecutively, we get a straight with length greater than d . Thus we

have a path which costs no more than the original but which is no longer a legitimate zigzag. Since it is not extremal, neither it nor the original path can be optimal. \square

Enumeration

Theorems 3, 4, and 5 allow a finite enumeration of the structure of optimal trajectories. The structure must be one of the following, or a subsection of one of the following:

Tangent	$\curvearrowright \uparrow \curvearrowright$	$\curvearrowright \downarrow \curvearrowright$	$\curvearrowleft \uparrow \curvearrowleft$	$\curvearrowleft \downarrow \curvearrowleft$
Tangent $_{\pi}$	$\uparrow \curvearrowleft \pi \downarrow$	$\downarrow \curvearrowright \pi \uparrow$	$\uparrow \curvearrowright \pi \downarrow$	$\downarrow \curvearrowleft \pi \uparrow$
Zigzag	$\uparrow \curvearrowright \downarrow \curvearrowright \uparrow$	$\downarrow \curvearrowleft \uparrow \curvearrowleft \downarrow$	$\uparrow \curvearrowleft \downarrow \curvearrowleft \uparrow$	$\downarrow \curvearrowright \uparrow \curvearrowright \downarrow$

5 Symmetries

Further analysis of the time optimal trajectories is difficult because of the large number of cases. This complexity is reduced using symmetries developed by Souères and Boissonnat [8] and Souères and Laumond [9] for steered cars.

The symmetries are summarized in Figure 10. Let “base” be any trajectory from $q = (x, y, \theta)$ to the origin. Then there are seven symmetric trajectories obtained by applying one or more of three transformations defined below. These transforms are isometries; if the base trajectory is optimal from the base configuration, the transformed trajectories are optimal from the transformed configurations.

Geometrically, the transformations reflect the plane across the origin or across one of three other lines: the x -axis, a line Δ_{θ} at angle $(\pi + \theta_s)/2$, or the line Δ_{θ}^{\perp} at angle $\theta_s/2$.

The three transformations are:

τ_1 : Swap \uparrow and \downarrow	T_1 : $q = (-x, -y, \theta)$
τ_2 : Reverse order	T_2 : $(x, y) = \text{Rot}(\theta)(x, -y)$
τ_3 : Swap \curvearrowright and \curvearrowleft	T_3 : $q = (x, -y, -\theta)$

Each transformation is its own inverse, and the three transformations commute. For any given base trajectory, the transformations yield up to seven different symmetric trajectories. The result is that all optimal trajectories fall in one of nine symmetry classes.

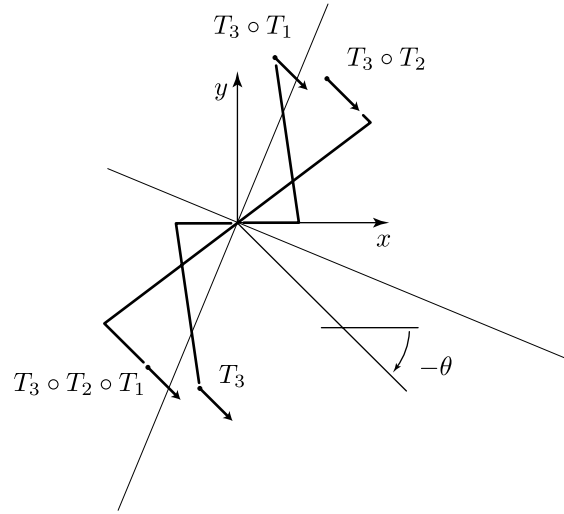
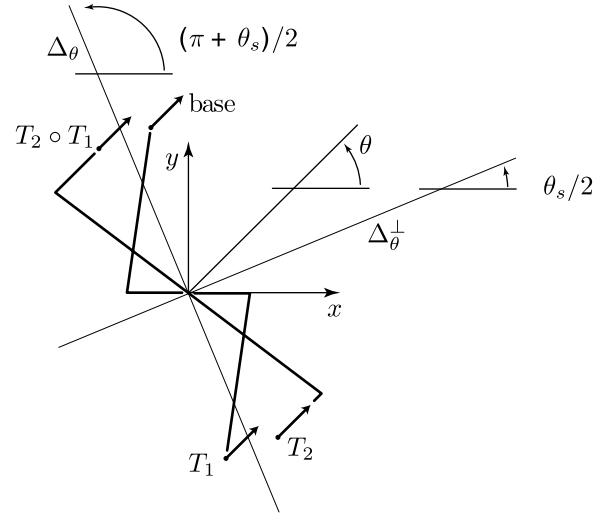


Figure 10: Given an optimal trajectory from “base” with heading θ_s to the origin with heading $\theta_g = 0$, transformations T_1, T_2 , and T_3 yield up to seven other optimal trajectories symmetric to the original.

	base	T_1	T_2	$T_2 \circ T_1$
A.	\curvearrowright	\curvearrowright	\curvearrowright	\curvearrowright
B.	\downarrow	\uparrow	\downarrow	\uparrow
C.	$\downarrow \curvearrowright$	$\uparrow \curvearrowright$	$\curvearrowright \downarrow$	$\curvearrowright \uparrow$
D.	$\curvearrowright \downarrow \curvearrowright$	$\curvearrowright \uparrow \curvearrowright$	$\curvearrowright \downarrow \curvearrowright$	$\curvearrowright \uparrow \curvearrowright$
E.	$\uparrow \curvearrowleft \pi \downarrow$	$\downarrow \curvearrowright \pi \uparrow$	$\downarrow \curvearrowright \pi \uparrow$	$\uparrow \curvearrowleft \pi \downarrow$
F.	$\curvearrowleft \downarrow \curvearrowleft$	$\curvearrowleft \uparrow \curvearrowleft$	$\curvearrowleft \downarrow \curvearrowleft$	$\curvearrowleft \uparrow \curvearrowleft$
G.	$\downarrow \curvearrowleft \uparrow$	$\uparrow \curvearrowright \downarrow$	$\uparrow \curvearrowright \downarrow$	$\downarrow \curvearrowleft \uparrow$
H.	$\curvearrowleft \downarrow \curvearrowright \uparrow$	$\curvearrowleft \uparrow \curvearrowright \downarrow$	$\uparrow \curvearrowright \downarrow \curvearrowright$	$\downarrow \curvearrowleft \uparrow \curvearrowright$
I.	$\uparrow \curvearrowright \downarrow \curvearrowright \uparrow$	$\downarrow \curvearrowleft \uparrow \curvearrowleft \downarrow$	$\uparrow \curvearrowright \downarrow \curvearrowright \uparrow$	$\downarrow \curvearrowleft \uparrow \curvearrowleft \downarrow$

	T_3	$T_3 \circ T_1$	$T_3 \circ T_2$	$T_3 \circ T_2 \circ T_1$
A.	\curvearrowright	\curvearrowright	\curvearrowright	\curvearrowright
B.	\downarrow	\uparrow	\downarrow	\uparrow
C.	$\downarrow\curvearrowright$	$\uparrow\curvearrowright$	$\downarrow\curvearrowleft$	$\uparrow\curvearrowleft$
D.	$\curvearrowleft\downarrow\curvearrowright$	$\curvearrowleft\uparrow\curvearrowright$	$\curvearrowleft\downarrow\curvearrowleft$	$\curvearrowleft\uparrow\curvearrowleft$
E.	$\uparrow\curvearrow\pi\downarrow$	$\downarrow\curvearrow\pi\uparrow$	$\downarrow\curvearrow\pi\uparrow$	$\uparrow\curvearrow\pi\downarrow$
F.	$\curvearrow\downarrow\curvearrow$	$\curvearrow\uparrow\curvearrow$	$\curvearrow\downarrow\curvearrow$	$\curvearrow\uparrow\curvearrow$
G.	$\downarrow\curvearrow\uparrow$	$\uparrow\curvearrow\downarrow$	$\uparrow\curvearrow\downarrow$	$\downarrow\curvearrow\uparrow$
H.	$\curvearrow\downarrow\curvearrow\uparrow$	$\curvearrow\uparrow\curvearrow\downarrow$	$\uparrow\curvearrow\downarrow\curvearrow$	$\downarrow\curvearrow\uparrow\curvearrow$
I.	$\uparrow\curvearrow\downarrow\curvearrow\uparrow$	$\downarrow\curvearrow\uparrow\curvearrow\downarrow$	$\uparrow\curvearrow\downarrow\curvearrow\uparrow$	$\downarrow\curvearrow\uparrow\curvearrow\downarrow$

We can analyze all types of trajectories by analyzing just one type from each of the nine classes, and then applying the transformations T_1, T_2, T_3 to obtain the other members of the class.

6 Time optimal trajectories.

In this section we identify the time optimal trajectories between any given start and goal configuration. We introduce a “goal-centric” coordinate system, with the origin coincident with the goal position, and the x axis aligned with the goal heading.

The symmetries of the previous section greatly simplify our analysis. We only need to consider a “base” region; the results then apply to symmetric regions. In principle, the analysis is completed by the following steps:

1. For each trajectory type, we identify every feasible choice of start configuration (x, y, θ) . This defines a map from trajectory type to a region of configuration space.
2. Now we consider a point in configuration space (x, y, θ) . If it is in only one region, then the corresponding trajectory type is optimal from that point.
3. When regions overlap, we derive additional necessary conditions for optimality or calculate the actual times for each trajectory type to disambiguate.

To illustrate this procedure, we present the following example (Figure 11). The feasible regions for $\downarrow\curvearrow\uparrow\curvearrow$ and $\curvearrow\downarrow\curvearrow\uparrow$ overlap. For almost all q_s in the overlap, there are two possible extremals but only one true optimal path. The Δ_θ line is a decision boundary: for q_s to the right of Δ_θ the optimum is $\curvearrow\downarrow\curvearrow\uparrow$, and to the left of Δ_θ the optimum is $\downarrow\curvearrow\uparrow\curvearrow$. Figure 11 illustrates the proof. First we observe that the alternatives give equal time on the Δ_θ line, because

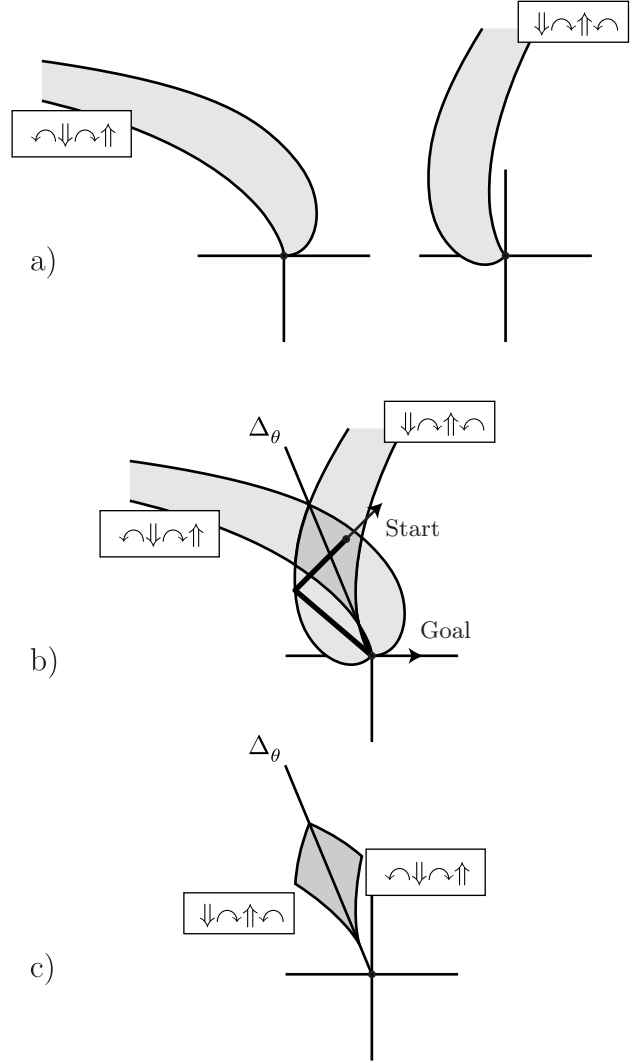


Figure 11: An example of overlapping regions. The path shown in b) is extremal, but not optimal.

that line is the axis of reflection for the $T_1 \circ T_2$ isometry. So both paths are optimal on Δ_θ .

Consider a $\downarrow\curvearrow\uparrow\curvearrow$ path from the start pose shown. When the path crosses Δ_θ during the \downarrow action, the remaining cost is unchanged if it switches to $\curvearrow\downarrow\curvearrow\uparrow$. But then the total path would have a structure of $\downarrow\curvearrow\downarrow\curvearrow\uparrow$, and would not be a legitimate extremal. \square

Similar techniques can be applied to the other regions. The end result is a mapping that defines for each point in configuration space the set of optimal trajectories from that

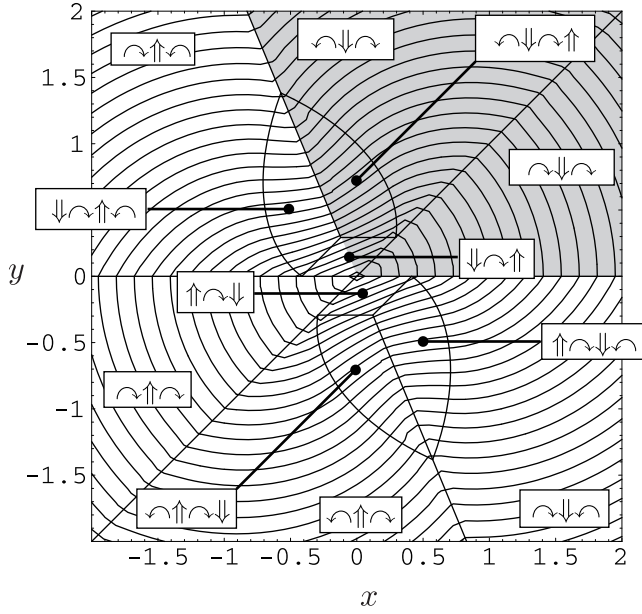


Figure 12: Optimal control for start configuration $q_s = (x, y, \frac{\pi}{4})$ and goal configuration $q_g = (0, 0, 0)$, with isocost lines. Coordinates are measured in units of b .

point to the origin. This mapping is illustrated by showing a slice at $\theta = \pi/4$ (Figure 12). The mapping from start configuration to optimal trajectory is usually, but not always, unique. At some boundaries in the figures there are two distinct trajectories that give the same time cost. More interesting is the case at $\theta = 0$ where a continuum of different trajectories of type A are all optimal, bounded by optimal trajectories of type B.

7 Algorithm for optimal control and value function.

We now present an algorithm to determine the optimal paths between a given start and goal position, and the time cost of those paths. For each optimal path structure, the necessary conditions yield a region. (Twelve such regions are shown in Figure 12.) The determines which region(s) the start configuration (x, y, θ) falls in, and then calculates the value function for one of the optimal path structures. For example, the function ValueBaseTSTS below calculates the cost of the fastest trajectory with a structure of $\curvearrowright\downarrow\curvearrowright\uparrow$.

```

Procedure ValueBaseTSTS( $q = (x, y, \theta)$ )
   $\arccos(1 - y) - \theta/2 - x + \sqrt{y(2 - y)}$ 
End ValueBaseTSTS

```

```

Procedure ValueBaseSTS( $q = (x, y, \theta)$ )
  If  $y = 0$  then  $|x| + \theta/2$ 
  else  $y(1 + \cos(\theta))/\sin(\theta) - x + \theta/2$ 
End ValueBaseSTS

```

```

Procedure ValueBaseTST( $q = (x, y, \theta)$ )
   $r = \|(x, y)\|$ 
   $\zeta = \arctan(y, x)$ 
   $r + \min(|\zeta| + |\zeta - \theta|, 2\pi + |\zeta| - |\zeta + \theta|)$ 
End ValueBaseTST

```

We now can define OptBVDD (optimal bounded velocity diff drive). The function recursively applies symmetry transforms until the configuration is in the base region.

The optimal path structure can then be determined based on the necessary conditions for extremal paths to be optimal. The value for that path structure is calculated. The recursion applies the appropriate combination of τ_1 , τ_2 , and τ_3 transforms to the base path structure to determine the actual optimal path structure.

```

Procedure OptBVDD( $q = (x, y, \theta)$ )
  if  $\theta \in (\pi, 2\pi)$  then  $\tau_3(\text{OptBVDD}(T_3(q)))$ 

   $r = \|(x, y)\|$ 
   $\zeta = \arctan(y, x)$ 
  if  $\zeta \in ((\theta + \pi)/2, \pi) \cup ((\theta - \pi)/2, 0)$ 
    then  $\tau_2(\text{OptBVDD}(T_2(q)))$ 
  if  $y < 0$  then  $\tau_1(\text{OptBVDD}(T_1(q)))$ 

  if  $\zeta \leq \theta$ 
    return( $\curvearrowright\downarrow\curvearrowright$ , ValueBaseTST( $q$ ))
  else if  $y \leq 1 - \cos(\theta)$ 
    return( $\downarrow\curvearrowright\uparrow$ , ValueBaseSTS( $q$ ))
  else if  $r \geq \tan(\zeta/2)$ 
    return( $\curvearrowright\downarrow\curvearrowright$ , ValueBaseTST( $q$ ))
  else
    return( $\curvearrowright\downarrow\curvearrowright\uparrow$ , ValueBaseTSTS( $q$ ))
End OptBVDD

```

For the sake of brevity, certain special cases have been omitted from the pseudocode presented. Whenever two

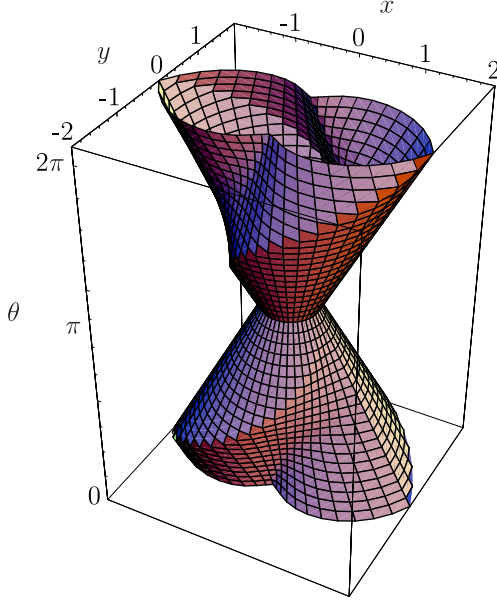


Figure 13: Reachable configurations in normalized time 2.

symmetric regions are adjacent, the fastest paths for both regions are optimal. For example, if the robot starts at $(0, 1, \pi)$, then both the paths $\curvearrowright\downarrow\curvearrowright$ and $\curvearrowright\uparrow\curvearrowright$ are optimal.

There are two other cases where multiple paths will be optimal. When $\theta_s = 0$, there may be a continuum of optimal five action paths, bounded by two different four-action paths. When $\theta_s = \pi$, there will be a continuum of optimal τ_π s paths (Class E), bounded by two-action paths of class C.

In all cases, the above algorithm will return a single optimal trajectory. Some additional bookkeeping would allow all of the optimal trajectories to be returned.

The level sets of the value function show the reachable configurations of the robot for some given amount of time. Figure 13 shows the shape of this region for time 2. (x , y , and time are normalized by b , the width of the robot.) Slices of this value function allow the regions in which various extremal paths are optimal to be seen more clearly. For example, figure 12 shows a slice where the angle between the start and goal robot is fixed at $\frac{\pi}{4}$.

8 Graphical method

Graphical construction of the time optimal trajectories is usually straightforward. Figure 14 shows examples for each

of the nine symmetry classes. The first five classes are really quite obvious. The last four classes are more challenging. In this section we will see a graphical way to identify which class gives the time-optimal trajectory, and to find the location of a via point if one exists.

The main idea is that the primary decision boundaries in the algorithm OptBVDD can be stated as conditions comparing the start heading θ_s , the goal heading θ_g , and the direction from start to goal. Based on this observation we can translate OptBVDD to a graphical procedure.

It is convenient to choose a different coordinate system, which we will call *bisector-centric* coordinates. We place the origin at the midpoint between the start and goal, with the positive x axis directed toward the goal. The y axis is then the perpendicular bisector of the segment between x_s and x_g , oriented in the usual way. For convenience, we define l to be the distance between the start and goal; i.e., $l = 2x_g$. We define the range of θ_s and θ_g to be $(-\pi, \pi]$.

It is also convenient to define the *startline* and *goalline* to be the lines aligned with the robot heading at the start and goal, respectively. We define n to be the intersection of the startline and goalline, if they intersect.

For the rest of this section, we first walk through the cases from simplest to most complex. For the zigzags the most interesting part is finding the location of a via point, if any, and addressing some of the special cases.

Tangent trajectories. Although the time optimal trajectories for classes A, B, and C are obvious, we list them here for completeness:

- A If s coincides with g , then turn in place.
- B If $\theta_s = \theta_g = 0$ or $\theta_s = \theta_g = \pi$, then roll straight to the goal.
- C If θ_s is 0 or π , then roll to the goal and turn to the goal heading. If θ_g is 0 or π then turn to the goal heading and then roll straight to the goal.

Next are the τ_{st} tangent and τ_{ts} tangent trajectories.

- D If θ_s and θ_g are neither 0 nor π , and have different signs, a turn from θ_s to θ_g by the shortest arc will either pass through 0 or π . So turn to 0 (π), roll forward (backward) to the goal, then turn to θ_g . The cost of this trajectory is

$$t_{\text{tangent}} = b \min(|\theta_s| + |\theta_g|, 2\pi - |\theta_s| - |\theta_g|) + l \quad (45)$$

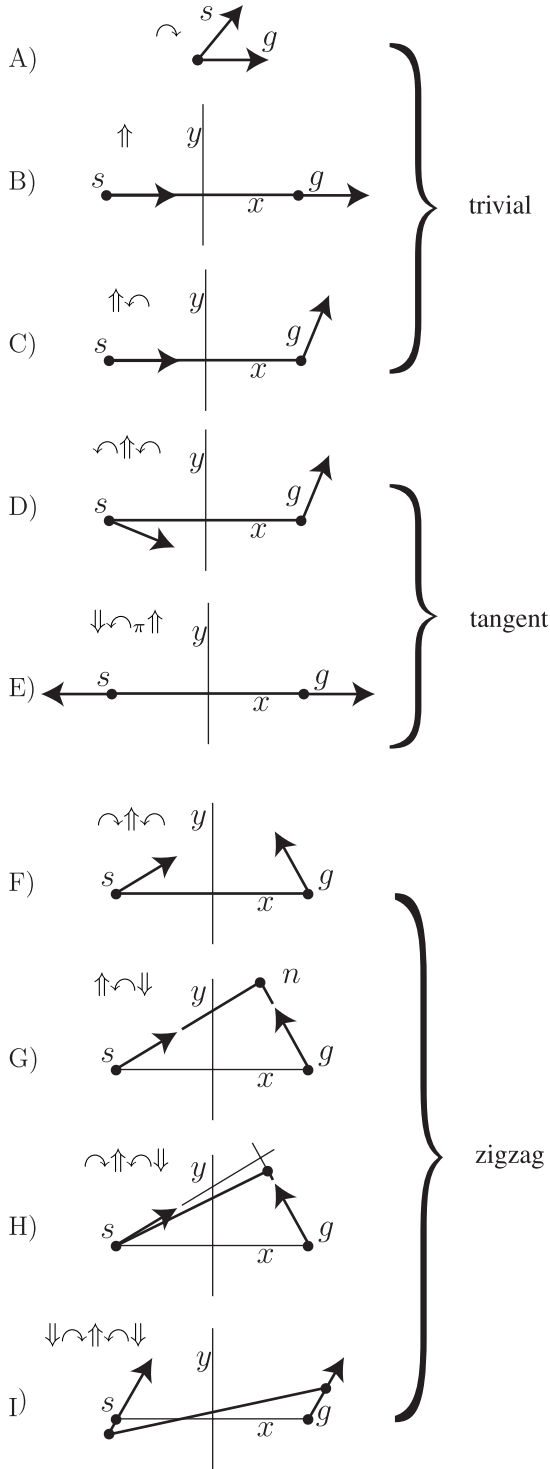


Figure 14: Examples for each of the nine classes.

E If $(\theta_s, \theta_g) = (0, \pi)$ or $(\pi, 0)$, then roll partway toward the goal, turn through π or $-\pi$, then roll the rest of the way to the goal. This yields a continuum of optimal trajectories of class E, and four optimal trajectories of class C. The cost of trajectories in this region is also given by equation 45.

Zigzag trajectories. If the signs of θ_s and θ_g are the same, but neither is equal to 0 or π , then the optimal trajectory will be a zigzag. The easiest way to approach zigzags is in terms of their via points. If there is *no* via, then we have a simple *tst* trajectory. If there are one or two via points, then we have to look at special cases to determine the number of vias and where they can occur.

- Every via point is on the startline or on the goalline;
- If the startline and the goalline intersect to the right of the y axis, there is at most one via, and it is on the goalline between g and n , or at n .
- If the startline and the goalline intersect to the left of the y axis, there is at most one via, and it is on the startline between s and n , or at n .
- If the startline and the goalline intersect on the y axis, then there may be two optimal trajectories of class H. For one, the via point is on the startline between s and n . For the other, the via point is on the goalline between g and n .
- If the startline and the goalline are parallel to the y axis, there may be up to four optimal trajectories of class H, and there may be a continuum of optimal trajectories of class I.
- If the startline and the goalline are parallel to each other but not the y axis, there may be up to two via-points. One is on the goalline, above the x axis iff the goalline does not intersect y above the x axis. The other is on the startline, with the same constraint. This gives up to two optimal trajectories of class H and a continuum of optimal trajectories of class I.

The above case analysis tells where to look for via points. Next we determine how to find the via points.

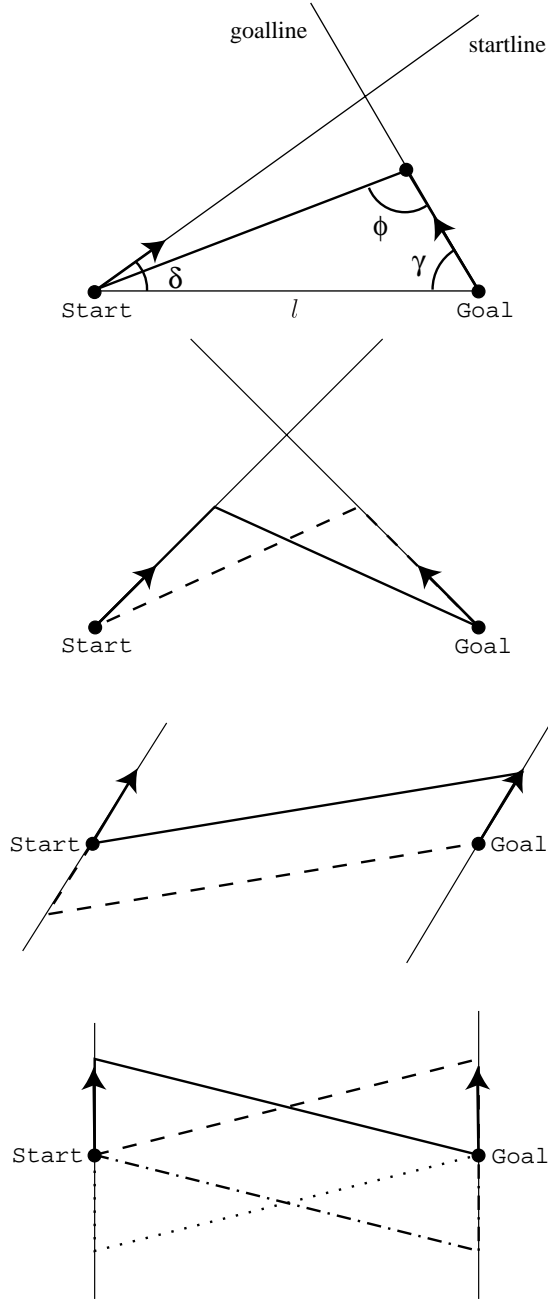


Figure 15: Examples of optimal zigzags. If startline and goalline are symmetric across the y axis, there are at least two zigzags of equal cost. If startline and goalline are parallel, there are two zigzags of equal cost, and zigzags of class I may also be optimal.

8.1 Graphical construction of Class H trajectories

For any trajectory with a via point we have a simple equation for the cost of the straight line actions. The above enumeration show that there are four line segments where the via point may fall. Let γ be the magnitude of the internal angle between any line segment possibly containing a via point and the x axis, and let δ be the internal angle of the other line segment on the same side of the x axis. (See figure 15.) Recall that ϕ is the angle of the turns of the zigzag extremal; we wish to determine the optimal value for ϕ . Some simple geometry gives us:

$$t_{zz} = b(2\phi + \delta + \gamma - \pi) + \frac{l(\sin(\phi + \gamma) + \sin \gamma)}{\sin \phi} \quad (46)$$

We take the derivative with respect to ϕ and set to zero. After some simplification, we get the following condition on ϕ :

$$\cos(\phi) = 1 - \frac{l \sin(\gamma)}{2b} \quad (47)$$

ϕ must be no greater than the turning angle of the fastest tst trajectory, and cannot be less than the angle of the turn if the via point were at n . (Recall that the via does not fall past this intersection.)

$$\phi \leq \min(|\theta_s| + |\theta_g|, 2\pi - |\theta_s| - |\theta_g|) \quad (48)$$

$$\phi \geq \pi - \gamma - \delta \quad (49)$$

If equation 47 have no solution, or requires that ϕ be larger than the turning angle of the fastest tst trajectory, the cost function is monotonically increasing with ϕ . In this case it is impossible to save turning time by using a four action trajectory; a trajectory of no more than three actions will be optimal. If equation 47 requires that $\phi < (\pi - \gamma - \delta)$, the cost function is monotonically decreasing with ϕ and the optimal trajectory will have a via at n .

There is a graphical interpretation of equation 47. For simplicity, we consider the case where n is above the x axis and to the left of the y axis, which means we should look for a via on the startline between s and g .

Put the robot at the origin, with angle θ_s . Roll it forward in a straight line. The right wheel rolls on a straight line, call it l_R . Now put the robot at the goal, with the right wheel on l_R . That is the configuration at which the robot should arrive at the goal. (See figure 16.) We can determine

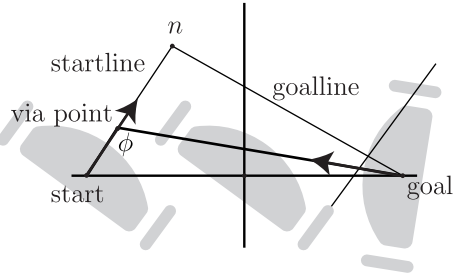


Figure 16: Graphical construction of optimal path

the location of the via point by rolling the robot from this configuration in a straight line until it intersects the startline.

It may be impossible to place the right wheel on l_R , or the goal configuration of the robot may seem to imply that the via point is not between s and n . This occurs when equation 47 has no solution satisfying equations 48 and 49, implying that either a trajectory of less than four actions is optimal, or the via point of the four action trajectory will be at n .

8.2 Construction of Class I trajectories

Equation 47 allows the optimal value of ϕ to be calculated, and this determines the structure of the optimal zigzag extremal. The above graphical algorithm allows this extremal to be constructed. However, if the startline and the goalline are parallel, there is more than one subsection of this extremal that connects the start and goal. In this case, if trajectories of class H are optimal, there will therefore also be a continuum of optimal class I trajectories.

9 Summary and Conclusion.

The bounded velocity model of diff drive robots is simple enough that the set of time optimal trajectories between any two robot configurations may be found. Pontryagin's Maximum Principle provides an elegant geometric program describing extremal trajectories. These extremal trajectories are a superset of the optimal trajectories.

We derived conditions necessary for extremal trajectories to be optimal. These conditions require that optimal trajectories fall in one of 40 extremal trajectory classes. We then applied symmetries developed by Souères and Boissonnat, reducing the set of trajectory classes to be analysed to 9. We analysed each of the 9 classes to determine the start and goal configurations for which it was optimal. This yields a

simple algorithm to determine the optimal trajectory structure and cost between any two configurations. The analysis also yields a simple geometric method of determining the optimal path structure.

Acknowledgments

We would like to thank Jean Paul Laumond for guidance. We would also like to thank Al Rizzi, Howie Choset, Ercan Acar, and the members of the Manipulation Lab for helpful comments.

References

- [1] D. J. Balkcom and M. T. Mason. Extremal trajectories for bounded velocity differential drive robots. In *IEEE International Conference on Robotics and Automation*, 2000.
- [2] D. J. Balkcom and M. T. Mason. Time optimal trajectories for bounded velocity differential drive robots. In *IEEE International Conference on Robotics and Automation*, 2000.
- [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] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. John Wiley, 1962.
- [5] 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.
- [6] D. B. Reister and F. G. Pin. Time-optimal trajectories for mobile robots with two independently driven wheels. *International Journal of Robotics Research*, 13(1):38–54, February 1994.
- [7] M. Renaud and J.-Y. 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.
- [8] 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.
- [9] P. Souères and J.-P. Laumond. Shortest paths synthesis for a car-like robot. *IEEE Transactions on Automatic Control*, 41(5):672–688, May 1996.
- [10] H. Sussmann and G. 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.