

Time Optimal Trajectories for Bounded Velocity Differential Drive Robots

Devin J. Balkcom

Matthew T. Mason

Robotics Institute and Computer Science Department
Carnegie Mellon University
Pittsburgh PA 15213

Abstract

A differential drive robot is perhaps the simplest type of mobile robot, and the bounded velocity model is perhaps the simplest useful model of the admissible controls. This paper develops the bounded velocity model for diff drive mobile robots, and derives the time-optimal trajectories.

1 Introduction

A differential drive robot has two independently driven coaxial wheels. It is the configuration used by most wheelchairs, and due to its simplicity is commonly used by mobile robots. By *bounded velocity*, we mean that the wheel angular velocities are bounded, but otherwise we allow essentially arbitrary motions of the robot. There are no bounds on wheel angular acceleration. In fact, we do not even require that angular acceleration be defined—discontinuities in wheel angular velocity are admissible.

This paper addresses the question: what are the fastest trajectories for a bounded velocity diff drive robot, in a planar environment free of obstacles? We prove that between given start and goal configurations, the fastest trajectories are composed of at most five segments, where each segment is either a straight line or a rotation about the robot's center. We present an algorithm for computing all optimal trajectories, and show a few plots illustrating the performance limits of bounded velocity diff drive robots.

1.1 Previous Work

Much of the work reported in this paper is a straightforward application of methods developed in the nonholonomic control and motion planning literature. We have found the surveys by Laumond [3] and Wen [10] to be very helpful. Most of the work on time-optimal control with bounded velocity models has focused on steered ve-

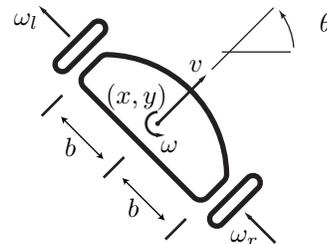


Figure 1: Notation

hicles rather than diff drives, originating with papers by Dubins [2] and Reeds and Shepp [4]. For diff drives, previous work has assumed bounded acceleration rather than bounded velocity. See, for example, papers by Reister and Pin [5] and Renaud and Fourquet [6]. Fortunately, the techniques developed for velocity models of steered cars may readily be applied to differential drives. The present paper follows the techniques developed in the papers by Sussman and Tang [9], by Souères and Boissonnat [7], and by Souères and Laumond [8].

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 parallel to the x -axis, and increases in the counterclockwise direction (Figure 1). 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

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

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

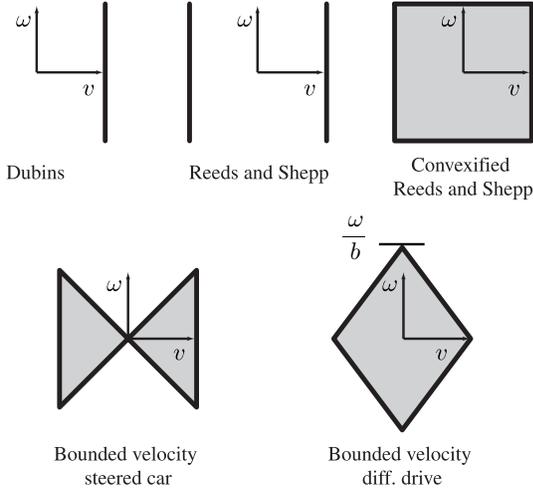


Figure 2: Bounded velocity models of mobile robots

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 $W = [-1, 1] \times [-1, 1]$

The admissible control region W 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 2).

We also need notation for trajectory types. We will use the symbols \uparrow , \downarrow , \curvearrowright , and \curvearrowleft , to denote forwards, backwards, left turns, and right turns. A trajectory of several segments is indicated by a string. Thus, for example, $\uparrow\curvearrowright\downarrow\curvearrowleft$ means a motion of four segments: forward, right turn, backward, left turn.

3 Time cost of saturated trajectories

We define a *saturated* trajectory to be one for which the input $w(t)$ is at the boundary of the box W over the entire trajectory. That is, at almost all times either ω_r or ω_l is at the limit. We define rectified arc length in E^2

$$s(t) = \int_0^t |v| \quad (5)$$

and in S^1

$$\sigma(t) = \int_0^t |\omega| \quad (6)$$

For a saturated trajectory, it is easily shown that

$$|v| + b|\omega| = 1 \quad (7)$$

almost everywhere. Integrating this equation yields

$$s + b\sigma = T \quad (8)$$

Thus the time for a saturated trajectory is just the sum of the arc length in E^2 and the arc length in S^1 scaled by the robot radius b . This suggests that to minimize the time we ought to turn in place or make straight lines. This observation is borne out by Pontryagin's Maximum Principle, which is discussed in the next section.

4 Controllability. Existence of optimal controls. Extremals.

This section summarizes results presented in [1]. The following key properties of the bounded velocity diff drive are presented:

- The bounded velocity diff drive robot is globally controllable, *i.e.* admissible trajectories exist for every pair of start and goal configurations.
- Time optimal controls exist. That is, given any pair of start and goal configurations, the set of admissible controls from the start to the goal will include at least one that minimizes the time.
- Pontryagin's Maximum Principle yields necessary conditions for time optimal controls. The trajectories satisfying these conditions are thus a superset of the time optimal trajectories, and are called the *extremal* trajectories.
- Using additional necessary conditions and identifying symmetries in the optimal trajectories, an enumeration of extremal is obtained.

The extremal trajectories can be expressed as a geometric program, using a construction called the η -line. It is a directed line in the plane, which divides the plane into a left half plane and a right half plane. Pontryagin's Maximum Principle implies that for any optimal trajectory there is an η -line such that the trajectory can be achieved by a control of the form:

$$\omega_l \begin{cases} = 1 & \text{if right wheel} \in \text{right half plane} \\ \in [-1, 1] & \text{if right wheel is on the line} \\ = -1 & \text{if right wheel} \in \text{left half plane} \end{cases} \quad (9)$$

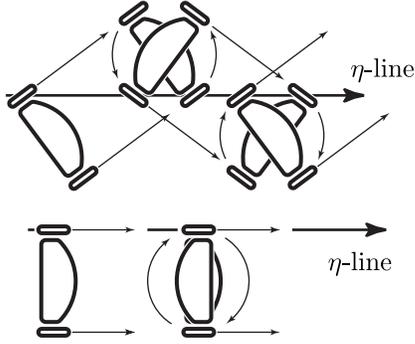


Figure 3: Two extremals: *zigzag right* and *tangent CW*. Other extremal types are *zigzag left*, *tangent CCW*, and turning in place: *CW* and *CCW*. Straight lines are special cases of zigzags or tangents.

$$\omega_r \begin{cases} = 1 & \text{if left wheel } \in \text{ left half plane} \\ \in [-1, 1] & \text{if the left wheel is on the line (10)} \\ = -1 & \text{if left wheel } \in \text{ right half plane} \end{cases}$$

The behavior of the robot falls into one of the following cases (see Figure 3):

- **CCW and CW:** If the robot is well in the left half plane, it turns in the counter-clockwise direction (CCW). CW is similar.
- **TCCW and TCW:** 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.

Since each extremal falls in one of the above classes, it follows that each time optimal trajectory does as well.

We note above that not all extremals are optimal. For example, a robot turning in place for several revolutions is not time optimal. Further, a zigzag of several segments is not optimal. In fact, it can be shown that optimal trajectories of class TCW or TCCW have at most three segments, and that optimal trajectories of class ZR and ZL have at most 5 segments. These restrictions give us a finite enumeration that includes all optimal trajectories.

The final result of [1] is to reduce the complexity of enumerating and analyzing these classes by employing symmetries developed by Souères and Boissonnat [7] and Souères and Laumond [8]. The result is that all optimal trajectories fall into one of nine symmetry classes.

The symmetry classes are expressed in terms of three transformations between symmetric robot start configurations, T_1 , T_2 , and T_3 , and three corresponding transformations between symmetric path structures.

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

The transformations are then applied to present an enumeration of trajectories classified by symmetry class:

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

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

5 Time optimal trajectories.

The previous section summarized the results of [1]. In this section we complete the analysis of extremal trajectories to identify the time optimal trajectories.

In principle, the analysis is completed by the following steps:

1. We choose the origin coincident with the goal posi-

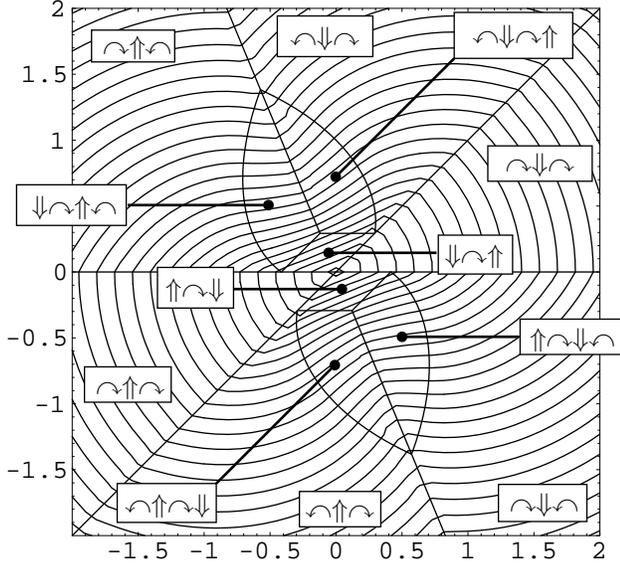


Figure 4: Optimal control for start configuration $q_s = (x, y, \frac{\pi}{4})$ and goal configuration $q_g = (0, 0, 0)$. Coordinates are normalized by division by b .

tion, and assume a goal heading of zero.

2. 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.
3. 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.
4. When regions overlap, we calculate the actual times for each trajectory type to disambiguate.

Naturally, the process is eased by employing the symmetries. The result is a mapping that defines for each point in configuration space the set of optimal trajectories from that point to the origin. This mapping is illustrated by showing two slices at $\theta = \pi/4$ (Figure 4) and at $\theta = 0$ (Figure 5). 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.

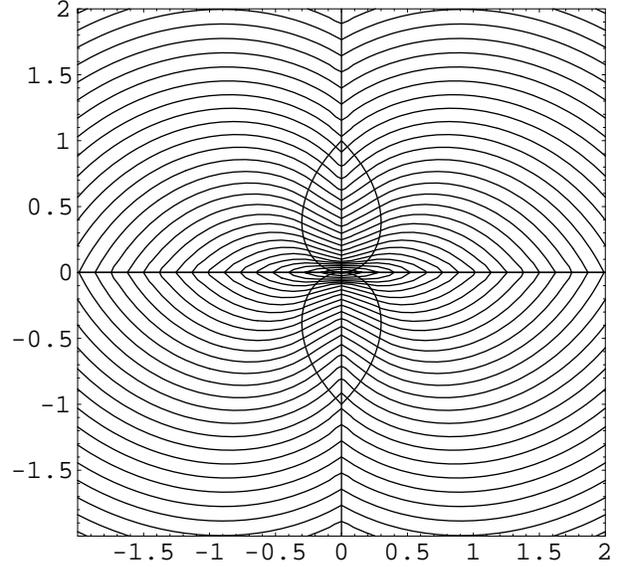


Figure 5: Optimal control for start configuration $q_s = (x, y, 0)$ and goal configuration $q_g = (0, 0, 0)$.

6 Algorithm for optimal control and value function. Balls.

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 type of optimal path, the necessary conditions yield a region as shown in Figures 4 and 5. The algorithm uses the start configuration (x, y, θ) to identify the correct region(s) and then calculates the value function for one of the optimal path structures. The algorithm employs the three symmetries $T_1, T_2,$ and T_3 defined earlier to reduce the number of cases.

First we define functions to calculate the cost of the fastest trajectory for the base trajectory of each symmetry class. For example, the function ValueBaseTSTS below calculates the cost of the fastest trajectory with a structure of $\curvearrowright\downarrow\curvearrowleft\uparrow$.

```

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

```

```

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

```

```

Procedure ValueBaseTST( $(r, \phi, \theta)$ )
   $r + \min(|\phi| + |\phi - \theta|, 2\pi - (|\phi| + |\phi - \theta|))$ 

```

End ValueBaseTST

We now can define OptBVDD. The function recursively applies symmetry transforms until the configuration is in a region for which one of the base trajectories for the symmetry classes is optimal. 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, θ))
  if θ ∈ (π, 2π) then τ3(OptBVDD(T3(q)))

  r = || x y ||
  φ = arctan(y, x)
  if φ ∈ (θ/2, π) then τ2(OptBVDD(T2(q)))
  if y < 0 then τ1(OptBVDD(T1(q)))

  if φ ≤ θ or r ≥ tan(φ/2)
    ValueBaseTST(r, φ, θ)
  else if y ≤ 1 - cos(θ)
    ValueBaseSTS(q)
  else
    ValueBaseTSTS(q)
End OptBVDD

```

For the sake of brevity, certain special cases which should be considered have been omitted from the pseudocode presented. Although the value function remains the same, there are cases where more than one path structure may be optimal. Whenever two 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 $\curvearrowleft\uparrow\curvearrowleft$ are optimal. The other special case occurs when $\theta = 0$. In this case, there are a number of five-segment paths that are also optimal whenever a four-segment path is optimal. In the complete algorithm, these cases need to be treated specially and return multiple path structures.

The value function which the algorithm expresses provides a nonholonomic metric in the configuration space [3]. The level sets of the value function show the reachable configurations of the robot for some given amount of time. Figure 7 shows the shape of this region using normalized x and y and normalized time 2. (Space and time are normalized by dividing by the robot radius b .) Slices of this value function allow the regions in which various extremal paths are optimal to be seen more clearly. For example, figure 4 shows a slice where the angle between the

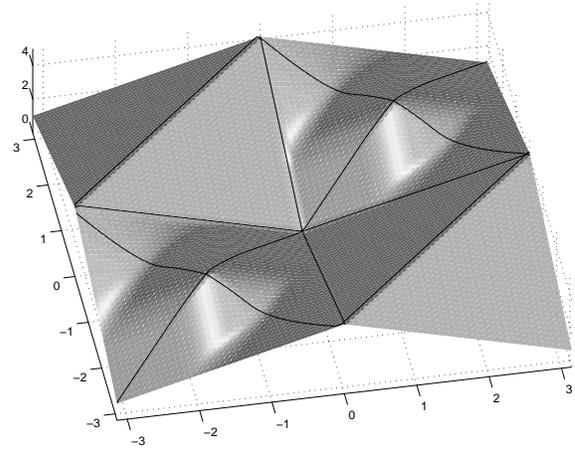


Figure 6: Time cost for $(x_s, y_s) = (1, 0)$

start and goal robot is fixed at $\frac{\pi}{4}$. Other slices of the ball may also be informative. For example, Figure 6 shows the value function as a function of the start and goal headings, where the goal position is $(0, 0)$ and the start position is $(1, 0)$. Each trajectory type is represented by some region of the figure.

7 Summary and Conclusion.

The bounded velocity model of the diff drive vehicle does not take into account some important factors such as the momentum of the robot. For this reason, bounded acceleration models more accurately describe the behaviour of most real differential drive robots. However, the bounded velocity model is simple enough that the set of time-optimal trajectories between any two robot configurations may be found. Pontryagin's Maximum Principle suggests a set of extremal trajectories, and a combination of calculus and geometry eliminate suboptimal trajectories from this set. Once the optimal trajectories have been found, we can calculate a value function between any start and goal configuration.

The time optimal trajectories for the bounded velocity diff drive robot are surprisingly simple, and are composed only of turns in place and straight lines. The structure of these trajectories may justify the simple technique of programming diff drive robots to move only in turns and straight lines. The value function is useful because it provides an easily calculated lower bound on the time cost between two configurations of a real diff drive robot with bounds on the velocity of its wheels.

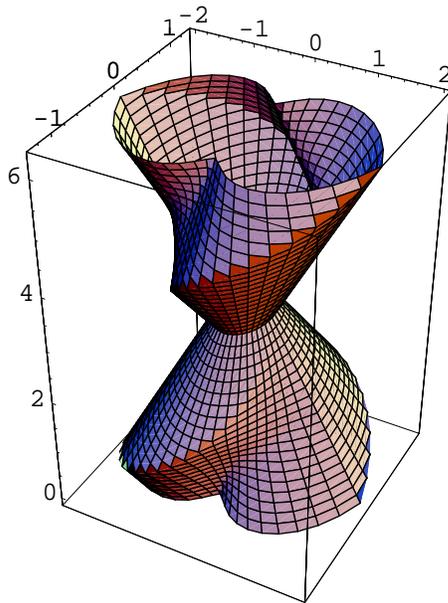


Figure 7: Reachable configurations in normalized time 2.

Acknowledgments

We would like to thank Jean Paul Laumond for guidance. We would also like to thank Al Rizzi, Howie Choset, Erkan 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. submitted.
- [2] 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.
- [3] J. P. Laumond. Nonholonomic motion planning for mobile robots. Technical report, LAAS, 1998.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. T. Wen. *Control of Nonholonomic Systems*, chapter 76.3, pages 1359–1368. CRC Press; IEEE Press, 1996.