

# 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? Our companion paper [1] proves 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. This paper completes the analysis of these trajectories.

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

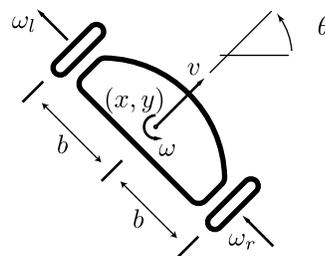


Figure 1: Notation

very helpful. 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 [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 apply readily 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  $\theta$  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  $\omega$ . The robot's width is  $2b$ . The wheel angular velocities are  $\omega_l$  and  $\omega_r$ . With suitable choices of units we obtain

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

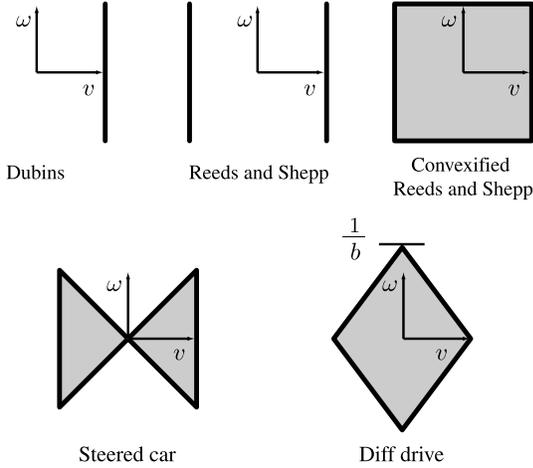


Figure 2: Bounded velocity models of mobile robots

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

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

The admissible control region  $W$  provides a convenient comparison with previously studied bounded velocity models. If we plot  $W$  in  $v$ - $\omega$  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  $\omega_r$  or  $\omega_l$  is at the limit. We define rectified arc length in the plane of robot positions

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

and in the circle of robot orientations

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

For a saturated trajectory, it is easily shown that

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

almost everywhere. Integrating this equation yields

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

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. Our companion paper [1] proves that this is indeed the case using Pontryagin's maximum principle.

### 4 Controllability. Existence of optimal controls. Extremals.

This section summarizes the results of our companion paper [1]. The bounded velocity diff drive is globally controllable, and time optimal controls exist. 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 an enumeration of extremals is obtained.

The extremal trajectories can be expressed as a geometric program, using a construction called the  $\eta$ -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  $\eta$ -line such that the trajectory can be achieved by a control of the form:

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

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

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

- CCW and CW: If the robot is in the left half plane out of reach of the  $\eta$ -line, it turns in the counter-clockwise direction (CCW). CW is similar.

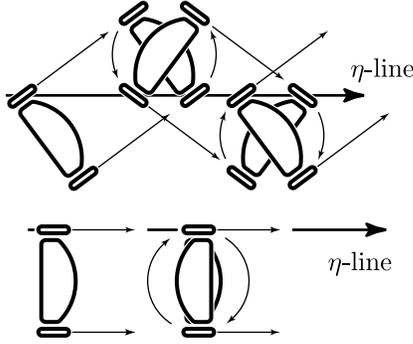


Figure 3: Two extremals: *zigzag right* and *tangent CW*. Other extremal types are *zigzag left*, *tangent CCW*, and turning in place: *CW* and *CCW*. Straight lines are special cases of zigzags or tangents.

- TCCW and TCW: If the robot is in the left half plane, but close enough that a circumscribed circle is tangent to the  $\eta$ -line, then the robot may either roll straight along the line, or it may turn through any positive multiple of  $\pi$ . TCW is similar.
- ZR and ZL: If the circumscribed circle crosses the  $\eta$ -line, then a zigzag behavior occurs. The robot rolls straight in the  $\eta$ -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:  $\dots \uparrow\curvearrowright\downarrow\curvearrowleft\uparrow \dots$  called *zigzag right* ZR, and  $\dots \uparrow\curvearrowleft\downarrow\curvearrowright\uparrow \dots$  called *zigzag left* ZL.

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

A robot turning in place for several revolutions is not time optimal. Further, a zigzag of several segments is not optimal. In fact, our companion paper shows that optimal trajectories must be subsections of the following extremals:

Zigzag	$\uparrow\curvearrowright\downarrow\curvearrowleft\uparrow$	$\downarrow\curvearrowleft\uparrow\curvearrowright\downarrow$	$\uparrow\curvearrowleft\downarrow\curvearrowright\uparrow$	$\downarrow\curvearrowright\uparrow\curvearrowleft\downarrow$
Tangent	$\curvearrowright\uparrow\curvearrowleft$	$\curvearrowleft\downarrow\curvearrowright$	$\curvearrowleft\uparrow\curvearrowright$	$\curvearrowright\downarrow\curvearrowleft$
Tangent	$\uparrow\curvearrow\pi\downarrow$	$\downarrow\curvearrow\pi\uparrow$	$\uparrow\curvearrow\pi\downarrow$	$\downarrow\curvearrow\pi\uparrow$

## 5 Symmetries

Symmetries developed by Souères and Boissonnat [7] and Souères and Laumond [8] reduce the complexity of analyzing the above enumeration of optimal trajectories.

The symmetries are summarized in Figure 4. Let “base” be an extremal trajectory from  $q = (x, y, \theta)$  to the origin. Then there are seven other trajectories, obtained by applying one or more of three transformations defined below.

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

The three transformations are:

$\tau_1$ : Swap $\uparrow$ and $\downarrow$	$T_1$ : $q = (-x, -y, \theta)$
$\tau_2$ : Reverse order	$T_2$ : $(x, y) = \text{Rot}(\theta)(x, -y)$
$\tau_3$ : 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.

	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.	$\curvearrowright\downarrow\curvearrowleft\uparrow$	$\curvearrowleft\uparrow\curvearrowright\downarrow$	$\uparrow\curvearrowright\downarrow\curvearrowleft$	$\downarrow\curvearrowleft\uparrow\curvearrowright$
C.	$\downarrow\curvearrow\uparrow$	$\uparrow\curvearrow\downarrow$	$\uparrow\curvearrow\downarrow$	$\downarrow\curvearrow\uparrow$
D.	$\uparrow\curvearrow\pi\downarrow$	$\downarrow\curvearrow\pi\uparrow$	$\downarrow\curvearrow\pi\uparrow$	$\uparrow\curvearrow\pi\downarrow$
E.	$\curvearrow\downarrow\curvearrow$	$\curvearrow\uparrow\curvearrow$	$\curvearrow\downarrow\curvearrow$	$\curvearrow\uparrow\curvearrow$
F.	$\curvearrow\downarrow\curvearrow$	$\curvearrow\uparrow\curvearrow$	$\curvearrow\downarrow\curvearrow$	$\curvearrow\uparrow\curvearrow$
G.	$\downarrow\curvearrow$	$\uparrow\curvearrow$	$\curvearrow\downarrow$	$\curvearrow\uparrow$
H.	$\downarrow$	$\uparrow$	$\downarrow$	$\uparrow$
I.	$\curvearrow$	$\curvearrow$	$\curvearrow$	$\curvearrow$

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

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. The number of cases can be further reduced by noticing that classes D, G, H, and I can be treated as degenerate or limiting cases of classes B, C, E, and F. Class A, consisting of five-segment trajectories that are optimal only when the robot start and goal headings are parallel, is also easily analyzed as it only occurs when two different members of class B are valid. Thus we have obtained a reduced set of extremal trajectories, which still includes all optimal trajectories, which can be analyzed by considering just four cases.

## 6 Time optimal trajectories.

In this section we discuss the analysis of extremal trajectories to identify the time optimal trajectories. We choose the origin coincident with the goal position, and assume a goal heading of zero. Naturally, the process is eased by employing the symmetries. 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  $(x, y, \theta)$ . This defines a map from trajectory type to a region of configuration space.
2. Now we consider a point in configuration space  $(x, y, \theta)$ . 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.

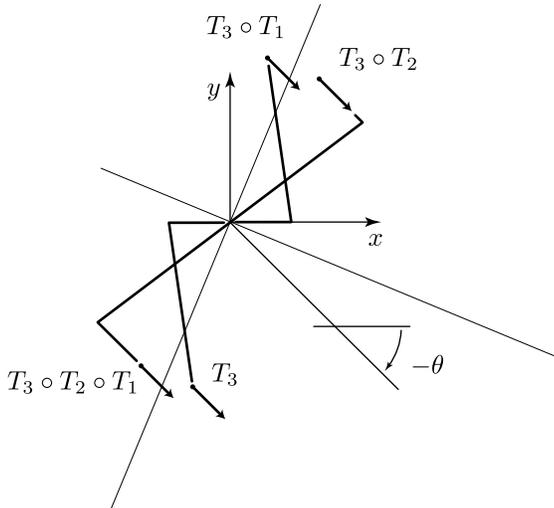
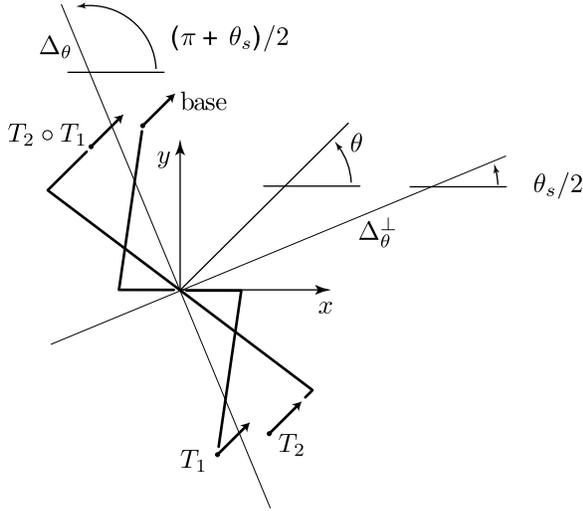


Figure 4: Given an optimal trajectory from “base” with heading  $\theta_s$  to the origin with heading  $\theta_g = 0$ , transformations  $T_1, T_2$ , and  $T_3$  yield up to seven other optimal trajectories symmetric to the original.

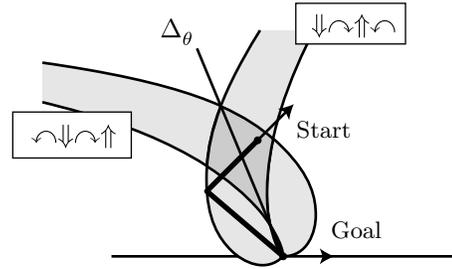


Figure 5: An example of overlapping regions. The path shown is extremal, but not optimal.

To illustrate this procedure, we present the following example. The feasible regions for  $\downarrow\rightarrow\uparrow\leftarrow$  and  $\leftarrow\downarrow\rightarrow\uparrow$  overlap. For almost all  $q_s$  in the overlap, there are two possible extremals but only one true optimal path. Figure 5 illustrates the proof that the  $\Delta_\theta$  line is a decision boundary. First we observe that the alternatives give equal time on the  $\Delta_\theta$  line, because that line is the axis of reflection for the  $T_1 \circ T_2$  isometry. So *both* paths are optimal on  $\Delta_\theta$ .

Now, the argument proceeds by contradiction. Suppose a  $\downarrow\rightarrow\uparrow\leftarrow$  path is optimal from the start pose shown. When the path crosses  $\Delta_\theta$ , the remaining cost is unchanged if it switches to  $\leftarrow\downarrow\rightarrow\uparrow$ . But then the total path would not be a legitimate zigzag. We conclude that for  $q_s$  to the right of  $\Delta_\theta$  the optimum is  $\leftarrow\downarrow\rightarrow\uparrow$ . Similarly, to the left of  $\Delta_\theta$  the optimum is  $\downarrow\rightarrow\uparrow\leftarrow$ .

Similar techniques can be applied to the other regions. The end result is a mapping that defines for each point in

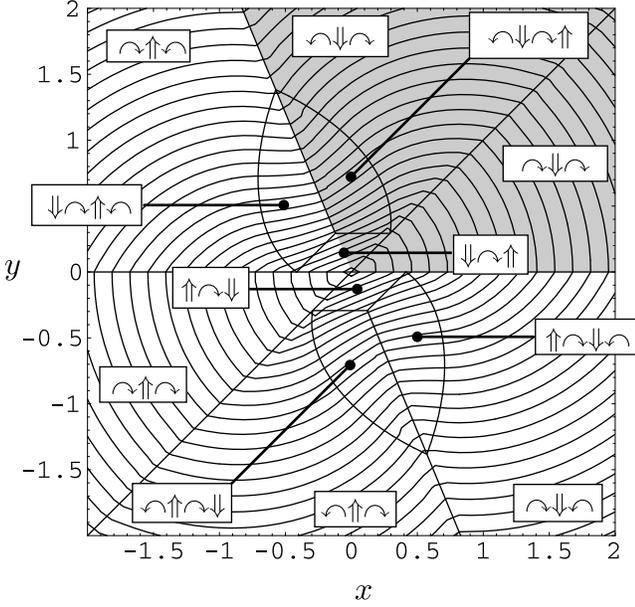


Figure 6: Optimal control for start configuration  $q_s = (x, y, \frac{\pi}{4})$  and goal configuration  $q_g = (0, 0, 0)$ . Coordinates are normalized by division by  $b$ .

configuration space the set of optimal trajectories from that point to the origin. This mapping is illustrated by showing a slice at  $\theta = \pi/4$  (Figure 6). 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. 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 6. The algorithm uses the start configuration  $(x, y, \theta)$  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  $\curvearrowright\downarrow\curvearrowleft\uparrow$ . Let  $(r, \zeta)$  be the polar coordinates of the start configuration.

```

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( $r, \zeta, \theta$ )
   $r + \min(|\zeta| + |\zeta - \theta|, 2\pi - (|\zeta| + |\zeta - \theta|))$ 
End ValueBaseTST

```

We now can define OptBVDD (optimal bounded velocity diff drive). The function recursively applies symmetry transforms until the configuration is in 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  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$  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  $\zeta \in (\theta + \pi/2, \pi)$  then  $\tau_2(\text{OptBVDD}(T_2(q)))$ 
  if  $y < 0$  then  $\tau_1(\text{OptBVDD}(T_1(q)))$ 

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

```

For the sake of brevity, certain special cases have been omitted from the pseudocode presented. Whenever two symmetric regions are adjacent, the fastest paths for both

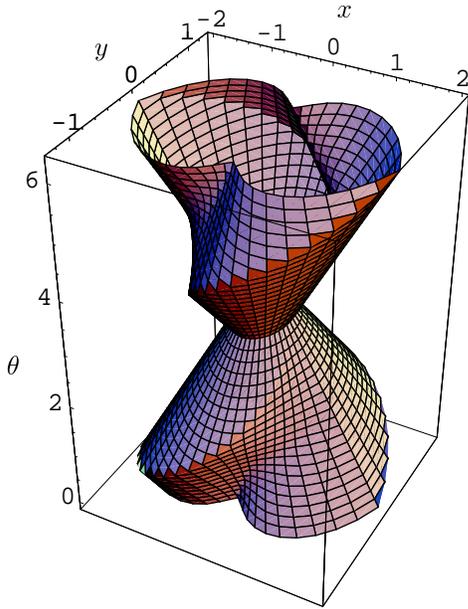


Figure 7: Reachable configurations in normalized time 2.

regions are optimal. For example, if the robot starts at  $(0, 1, \pi)$ , then both the paths  $\curvearrowright\downarrow\curvearrowleft$  and  $\curvearrowleft\uparrow\curvearrowright$  are optimal. There are two other cases where multiple path trajectories will be optimal. When  $\theta_s = 0$ , there may be a continuum of optimal five segment paths, bounded by two different four segment paths. When  $\theta_s = \pi$ , straight-turn $_{\pi}$ -straight paths (Class D) may be optimal; in this case there will also be a continuum of optimal paths of this form bounded by three segment trajectories of different classes. In each of these cases, the above algorithm will return an optimal trajectory. Some additional bookkeeping would allow all of the optimal trajectories to be returned in these cases.

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 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 6 shows a slice where the angle between the start and goal robot is fixed at  $\frac{\pi}{4}$ .

## 8 Summary and Conclusion.

The time optimal trajectories for the bounded velocity diff drive robot are simple, and are composed only of turns in place and straight lines. We have presented the value function, and an algorithm to determine the optimal trajectories between any start and goal configuration of the robot.

## 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] 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 98211, LAAS, May 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.