

OPTIMAL TRAJECTORIES OF CURVATURE CONSTRAINED MOTION IN THE HAMILTON-JACOBI FORMULATION

RYO TAKEI AND RICHARD TSAI

ABSTRACT. We propose a Hamilton-Jacobi equation approach for computing time-optimal trajectories of a vehicle which travel under curvature constraints. We derive a class of Hamilton-Jacobi equations which models such motions; it unifies two well-known vehicular models, the Dubins' and Reeds-Shepp's cars, and gives further generalizations. We prove that the value function of the control problem solves the Hamilton-Jacobi equation in the viscosity sense. Numerical schemes and results that illustrate the versatility of the Hamilton-Jacobi approach are presented for the two dimensional case.

1. INTRODUCTION

We present a Hamilton-Jacobi formulation that models the time optimal motion of a particle that traces paths with an upper bound on curvature. The problem arises in describing the motion of a *simple car*, illustrated in Figure 1. It can be shown that a simple car with axes that are L length apart traces a path with a curvature bounded from above by $\kappa_{\max} = (\tan \phi_{\max})/L$, where ϕ_{\max} the maximum angle that the (front) wheels can tilt. Equivalently, the simple car has a minimum *turning radius* $\rho = 1/\kappa_{\max}$. There are two well-known prototypical models of a simple car. The first is the *Dubins'*

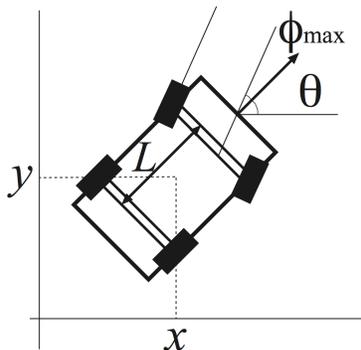


FIGURE 1. A simple car.

car [Dub57], a forward-moving (irreversible) car on a plane traveling at unit speed with a prescribed minimum turning radius. The second is the *Reeds-Shepp's car* [RS90], a reversible version of Dubins' car. At time $t > 0$, if we write $x(t), y(t)$ to be the spatial coordinates in two dimensions representing the position of the car and $\theta(t)$ to be the

direction of motion, the dynamical system that models the Dubins' and Reeds-Shepp's cars is

$$(1) \quad \begin{aligned} \dot{x}(t) &= \sigma(t) \cos \theta(t) \\ \dot{y}(t) &= \sigma(t) \sin \theta(t) \\ \left| \dot{\theta}(t) \right| &\leq \frac{1}{\rho}. \end{aligned}$$

Here, $\rho > 0$ is the (minimum) turning radius of the vehicle and the control $\sigma(\cdot)$ determines whether to move forward or backward. If $\sigma(t) \equiv 1$, Dubins' car is obtained, and if $\sigma(t) \in \{\pm 1\}$, we have the Reeds-Shepp's car. This can be further generalized to allow for different, spatially variable forward and reverse speeds and turning radii. Generalization could be made to higher dimensions, for instance, the problem of modeling an aircraft in three dimensional space (a *Dubins' plane* [CL07], also see Example 5 below). Throughout this article, we shall refer to the motion of such vehicles as *curvature constrained motion*.

In this article, we describe a Hamilton-Jacobi formulation that unifies a wide range of problems of curvature constrained motion in an Eulerian framework. This formulation computes the value function (see (9) below) by solving a first order partial differential equation, called a Hamilton-Jacobi equation, and naturally handles the aforementioned generalizations. There are several advantages to this approach. First, the value function sought is versatile in that it contains all relevant information pertaining to a path planning problem, such as minimum arrival times (cost-to-go), reachable sets (see Definition 24) and optimal paths. Second, the theory of weak solutions to Hamilton-Jacobi equations known as *viscosity solutions* is well understood [CL83, BCD97], see also section 3. Finally, convergent and efficient numerical schemes can be exploited from the well-developed numerical theory for Hamilton-Jacobi equations. These schemes are easy to implement, can be solved on uniform cartesian grids, and can handle obstacles with arbitrary geometries with ease.

The article is organized as follows. We close the introduction by mentioning other works concerning curvature constrained motion and Hamilton-Jacobi equations. In section 2 we formally derive the Hamilton-Jacobi equation for curvature constrained motion in two and three dimensions, and present the equations for some well known models. A recurring theme in this article is the striking difference between Dubins' and Reeds-Shepp's car; the value function for the former is continuous and the latter is not. In section 3, we prove that the value function of the Reeds-Shepp's car is continuous by showing that it is *small-time local controllable*, and therefore fits in to the standard viscosity solution framework. For the Dubins' car, a more general class of weak solutions (called *e-solutions*) are exploited to establish the connection between the (discontinuous) value function and the Hamilton-Jacobi equation. In section 4, approximation schemes for computing the value function and the optimal trajectories are presented. While standard monotone finite difference scheme are sufficient for problems with continuous value functions, the performance deteriorates for solutions with discontinuities (see Remark 18). Thus, we apply a semi-Lagrangian scheme to specifically solve the Dubins' car problem. Finally, numerical results are presented for the two dimensional cases to justify the convergence and versatility of the formulation as well as to point out limitations.

1.1. Previous Work. One of the earliest relevant work in curvature constrained motion is the article of Dubins [Dub57], where it was shown that curvature constrained geodesics

in \mathbb{R}^2 could be classified into just six types of paths. Later, Reeds and Shepp [RS90] generalized Dubins' work to allow for cusps in the curve, and showed that such geodesics can be classified into 68 types. The pioneering works of Dubins, Reeds and Shepp were of theoretical interest, but provided no explicit formulas.

Traditionally, numerical algorithms for curvature constrained path planning have been combinatorial in a Lagrangian framework. Barraquand and Latombe [BL93] constructed a reachability tree assuming that, for example, Dubins' car would locally only move straight, fully left or fully right for a short distance. This assumption is known as the *bang-bang principle* [SVV65]. In the presence of polygonal obstacles, the authors in [WA96, JC92] developed algorithms to compute optimal ϵ -robust paths; informally, a path is ϵ -robust if perturbations by $\epsilon/2$ distance is still feasible. Typically, the complexity of such algorithms are inversely proportional to ϵ . Others have considered building optimal curvature constrained paths among obstacles based on obstacle free optimal paths [RW00], way-points known *a priori* [BK07], or optimal paths with unconstrained curvatures [LJTM94]. Our approach is similar to that of [KL99]: first determine the value function by discretizing the dynamic programming principle, then compute individual trajectories. Their approach (which is Eulerian) involves approximating the value function on tetrahedrons constructed using barycentric coordinates. For a comprehensive treatment of path planning algorithms and relevant bibliography, see [LaV06].

We also mention a related work [PT09], where reachability sets of Dubins' and Reeds-Shepp's cars (and others) are geometrically investigated. While they have presented interesting computational results of reachability sets of various simple cars, their approach does not yield individual optimal trajectories.

Our approach approximates the value function on a uniform mesh, via numerically solving a Hamilton-Jacobi equation. It is well known [CL84, BS91] that a monotone, consistent scheme for a Hamilton-Jacobi equation converges to the unique viscosity solution; further techniques for constructing monotone finite difference schemes were studied in [Obe06]. Standard monotone, consistent numerical methods for solving discretized static Hamilton-Jacobi equations can be roughly classified into two types: finite difference schemes and discretization of the dynamic programming principle. The former includes the fast marching and sweeping methods; both solve the same system of nonlinear equations derived from discretizing the Hamilton-Jacobi equation. The fast marching method [Tsi95, Set95] is based on the monotonicity of information along characteristics. The solutions are constructed using a variant of Dijkstra algorithm and a heap-sort data structure. The fast sweeping method is based on a Gauss-Seidel iterative strategy. Rouy and Tourin [RT92] showed that iteratively sweeping a monotone discretization of the eikonal equation solved for the unique viscosity solution. Efficient and easy-to-implement Gauss-Seidel type update sweeping schemes for Hamilton-Jacobi equations with convex Hamiltonians were proposed in [TCOZ03, Zha04, KOT04]. Similar methods for more general, non-convex Hamiltonian was proposed by Kao et. al. [KOQ04]. The discretization of the dynamic programming principle is often referred to as semi-Lagrangian schemes [FF02]. The main advantage of the semi-Lagrangian scheme over finite difference schemes is the former's convergence properties for discontinuous solutions, see section 4.2. We also mention that the techniques of discontinuous Galerkin finite element methods have been applied to Hamilton-Jacobi equations [HS, CS07].

This article is a generalization of the results in [TTSL10], where the authors (joint with H. Shen and Y. Landa) studied the Hamilton-Jacobi equation and finite difference numerical scheme for the Dubins' car problem.

2. THE HAMILTON-JACOBI FORMULATION

In this section we formally derive a first order PDE, the Hamilton-Jacobi equation, that models a class of curvature constrained motions in free space. For exposition, we shall formulate the prototypical case of the minimum exit-time problem with spatially constant speeds and turning radii. Modifications to accommodate further generalizations, such as terminal costs, variable running costs and spatially varying speeds and turning radii can be modified accordingly. Readers interested in derivations of Hamilton-Jacobi equations for general optimal control problems are referred to [BCD97].

2.1. Notations and setup. We begin by developing the necessary notions used in the rest of the article. For convenience, we provide a list of commonly used notations in section 2.1.1.

In two dimensions, consider a bounded, connected, open set $\Omega_0 \subset \mathbb{R}^2$ representing the free space. We call a *pose* to be point in Ω_0 coupled with an angle $\theta \in [0, 2\pi)$ representing the direction of forward motion of the car, in radians. Thus we write $\Omega := \Omega_0 \times [0, 2\pi)$ as the set of all allowable poses for the given free space. In three dimensions the free space is $\Omega_0 \subset \mathbb{R}^3$, and using spherical coordinates the set of all allowable poses become $\Omega = \Omega_0 \times [-\pi/2, \pi/2] \times [0, 2\pi)$.

We now derive a more general version of the dynamical system (1) for two dimensional curvature constrained motion. Define a *path* or *trajectory* as a piecewise differentiable curve $\mathbf{z}: t \in [0, \infty) \rightarrow \Omega$,

$$(2) \quad \mathbf{z}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \theta(t) \end{bmatrix}, \quad \mathbf{x}(t) \in \Omega_0, \theta(t) \in [0, 2\pi).$$

The parameter t represents the time elapsed from the start of the motion. We assume the vehicle travels at a given finite positive speed. By introducing a control function $\sigma(t) \in \{\pm 1\}$, the spatial dynamics can be written as

$$(3) \quad \dot{\mathbf{x}}(t) = \sigma(t) v(\mathbf{z}(t), \sigma(t)) \begin{bmatrix} \cos \theta(t) \\ \sin \theta(t) \end{bmatrix}.$$

We shall call $\sigma(t)$ the *switching control*, since its sign “switching” between $+1$ and -1 determines whether to move forward or reverse. The function $v(\mathbf{z}(t), \sigma(t))$ is the speed of motion at the pose $\mathbf{z}(t)$ and choice of forward/reverse motion $\sigma(t)$. Note that (3) implies

$$(4) \quad |\dot{\mathbf{x}}(t)| = v(\mathbf{z}(t), \sigma(t)),$$

which is the speed constraint, as required.

For the phase dynamics, $\dot{\theta}(t)$, the derivation requires the bound on the curvature, modeled by

$$(5) \quad \left| \frac{d\theta}{ds} \right| \leq \frac{1}{\rho},$$

where $\rho > 0$ is the minimum turning radius, and s is the arc length parameter. If a vehicle travels with speed $v(\mathbf{z}(t), \sigma(t)) > 0$,

$$(6) \quad \frac{d\theta}{ds} = \frac{d\theta}{dt} \bigg/ \frac{ds}{dt} = \frac{1}{v(\mathbf{z}(t), \sigma(t))} \dot{\theta}(t),$$

and (5) translates to $|\dot{\theta}(t)| \leq v(\mathbf{z}(t), \sigma(t))/\rho$. Introduce another control $a(t) \in [-1, 1]$ to write the phase dynamics as

$$(7) \quad \dot{\theta}(t) = a(t) \frac{v(\mathbf{z}(t), \sigma(t))}{\rho}, \quad a(t) \in [-1, 1].$$

(We shall see later that for trajectories in free space, the control $a(t)$ is invoked only at +1 and -1, see equation (15).) We combine (3) and (7) to obtain the *kinetic equation* and boundary condition:

$$(8) \quad \begin{aligned} \dot{\mathbf{z}}(t) &= f(\mathbf{z}(t), \sigma(t), a(t)) := v(\mathbf{z}(t), \sigma(t)) \left[\sigma(t) \cos \theta(t), \sigma(t) \sin \theta(t), a(t) \frac{1}{\rho} \right]^T, \\ \mathbf{z}(0) &= \mathbf{y}_0. \end{aligned}$$

The function f is called the *dynamics* associated with the kinetic equation (8). We say that a path is *feasible* if it is contained in Ω for all $t \geq 0$. We call the *admissible paths from the pose* \mathbf{y}_0 , denoted by $\mathcal{A}_{\mathbf{y}_0}$, the set of feasible paths $\mathbf{z}(\cdot)$ satisfying (8).

Finally, given a *target set* $\mathcal{T} \subset \Omega$, we define the *value function* $u: \Omega \rightarrow \mathbb{R}$ as

$$(9) \quad u(\mathbf{y}) = \inf \{ t \mid \mathbf{z}(\cdot) \in \mathcal{A}_{\mathbf{y}}, \mathbf{z}(t) \in \mathcal{T} \}.$$

In other words, $u(\mathbf{y})$ is the minimum arrival time to \mathcal{T} from \mathbf{y} under the speed constraint (4) and the curvature constraint (5). By definition, the value function is non-negative, and $u(\mathbf{y}_0) = 0$ if and only if $\mathbf{y}_0 \in \mathcal{T}$.

2.1.1. *Summaries of notation.* As a reference, we list notations used frequently throughout this article. For concreteness, the notation is for the two dimensional case; the three dimensional case, using spherical coordinates, are analogous.

- $\Omega_0 \subset \mathbb{R}^2$ is the spatial domain.
- $\Omega = \Omega_0 \times [0, 2\pi)$ is the set of all poses.
- $\mathbf{x} = (x, y) \in \Omega_0$ is a generic point in space.
- $\mathbf{y} \in \Omega$ is a generic pose.
- $\mathbf{z}(\cdot) = (\mathbf{x}(\cdot), \theta(\cdot))$ is a generic (feasible) path in Ω , parametrized by time t .
- $\sigma(\cdot), a(\cdot)$ are generic controls (parametrized by time t) associated with the car maneuverability properties. In particular, $\sigma(\cdot)$ is referred to as the switching control.
- \mathcal{T} denotes the target set, typically a final location or a final pose.
- f denotes the dynamics of the control problem.
- $\mathcal{A}_{\mathbf{y}}$ represent the set of feasible paths from the pose \mathbf{y} associated with given dynamics.
- u denotes the value function associated with given dynamics and target set.
- $B_r(\mathbf{y})$ is a ball of radius r centered at \mathbf{y} .
- $Dw(\mathbf{y})$ denotes the gradient of a function $w: \Omega \rightarrow \mathbb{R}$ at the point $\mathbf{y} \in \Omega$.
- w_x denotes the partial derivative of a function w in the variable x .

- The forward and reverse speeds and turning radii in (8) as a function of the pose \mathbf{y} are denoted as

$$(10) \quad \begin{aligned} v(\mathbf{y}, \sigma) &= \begin{cases} v_1(\mathbf{y}) & = \text{forward speed } (\sigma = 1), \\ v_{-1}(\mathbf{y}) & = \text{reverse speed } (\sigma = -1), \end{cases} \\ \rho(\mathbf{y}, \sigma) &= \begin{cases} \rho_1(\mathbf{y}) & = \text{forward turning radius } (\sigma = 1), \\ \rho_{-1}(\mathbf{y}) & = \text{reverse turning radius } (\sigma = -1). \end{cases} \end{aligned}$$

2.2. Formal derivations. Here, we derive a class of Hamilton-Jacobi equations (16) and (18) that describes curvature constrained motions in two and three dimensions, respectively. For brevity we suppress the dependence of speeds and turning radii on the pose, i.e. $v_1(\mathbf{y}) = v_1$, $\rho_1(\mathbf{y}) = \rho_1$, etc.

2.2.1. Two dimensions. Recall the notation for a path (2). In two dimensions, the kinetic equations for the forward motion is

$$(11) \quad \dot{\mathbf{z}}(t) = f_1(\mathbf{z}(t), a(t)) := \left[v_1 \cos \theta(t), v_1 \sin \theta(t), a(t) \frac{v_1}{\rho_1} \right]^T, \quad a(t) \in [-1, 1],$$

and the reverse motion is

$$(12) \quad \dot{\mathbf{z}}(t) = f_{-1}(\mathbf{z}(t), a(t)) := \left[-v_{-1} \cos \theta(t), -v_{-1} \sin \theta(t), a(t) \frac{v_{-1}}{\rho_{-1}} \right]^T, \quad a(t) \in [-1, 1].$$

Thus, the set of admissible paths $\mathcal{A}_{\mathbf{y}_0}$ contains feasible paths defined by the kinetic equation with dynamics

$$f(\mathbf{z}(t), \sigma(t), a(t)) := f_{\sigma(t)}(\mathbf{z}(t), a(t)), \quad \sigma(t) \in \{\pm 1\}, a(t) \in [-1, 1],$$

with boundary conditions $\mathbf{z}(0) = \mathbf{y}_0$. We emphasize that, $\sigma(t)$ is the switching control that identifies forward (+1) and reverse (-1) motion.

We start the derivation with the *dynamic programming principle* [Bel10] for path planning problems with unit running cost (or equivalently, the minimum arrival time formulation):

$$(13) \quad u(\mathbf{y}) = \inf_{\mathbf{z}(\cdot) \in \mathcal{A}_{\mathbf{y}}} \{u(\mathbf{z}(\Delta t)) + \Delta t\}.$$

If u is smooth, a Taylor series expansion and a division by Δt formally implies that, in the limit $\Delta t \rightarrow 0$,

$$\begin{aligned} 0 &= \sup_{\mathbf{z}(\cdot) \in \mathcal{A}_{\mathbf{y}}} \{-Du(\mathbf{y}) \cdot \dot{\mathbf{z}}(0)\} - 1, \\ 0 &= \sup_{\hat{\sigma} \in \{\pm 1\}, \hat{a} \in [-1, 1]} \{-Du(\mathbf{y}) \cdot f(\mathbf{y}, \hat{\sigma}, \hat{a})\} - 1. \end{aligned}$$

This PDE is known as the *Hamilton-Jacobi-Bellman equation* associated with the dynamics f . The supremum is actually a maximum since the controls are invoked over compact sets:

$$(14) \quad \begin{aligned} 0 &= \max_{\hat{\sigma} \in \{\pm 1\}, \hat{a} \in [-1, 1]} \left\{ -v_{\hat{\sigma}} \left[\hat{\sigma} (\cos \theta u_x + \sin \theta u_y) + \hat{a} \frac{u_\theta}{\rho_{\hat{\sigma}}} \right] \right\} - 1 \\ 0 &= \max_{\hat{\sigma} \in \{\pm 1\}} \left\{ -v_{\hat{\sigma}} \hat{\sigma} (\cos \theta u_x + \sin \theta u_y) + v_{\hat{\sigma}} \max_{\hat{a} \in [-1, 1]} \left\{ -\hat{a} \frac{u_\theta}{\rho_{\hat{\sigma}}} \right\} \right\} - 1. \end{aligned}$$

The inner maximum can be evaluated by choosing \hat{a} to be

$$(15) \quad \hat{a} \in \partial[-1, 1] = \{\pm 1\}.$$

Thus, we arrive at the Hamilton-Jacobi equation for two dimensional curvature constrained motion, with constant speeds and turning radii (10):

$$(16) \quad 0 = \max_{\hat{\sigma} \in \{\pm 1\}} \left\{ -v_{\hat{\sigma}} \hat{\sigma} (\cos \theta u_x + \sin \theta u_y) + \frac{v_{\hat{\sigma}}}{\rho_{\hat{\sigma}}} |u_{\theta}| \right\} - 1.$$

2.2.2. *Three dimensions: spherical coordinates.* We derive the three dimensional analogue to (16) in spherical coordinates. For a path $\mathbf{z}(t) = (\mathbf{x}(t), \theta(t), \phi(t))$, where $\theta(t) \in [-\pi/2, \pi/2]$ and $\phi(t) \in [0, 2\pi)$ are the inclination and azimuth angles, respectively, the velocity is

$$\dot{\mathbf{x}}(t) = \sigma(t) v_{\sigma(t)} [\sin \theta(t) \cos \phi(t), \sin \theta(t) \sin \phi(t), \cos \theta(t)]^T,$$

and the curvature is

$$\begin{aligned} \left| \frac{d}{ds} \frac{\dot{\mathbf{x}}(t)}{|\dot{\mathbf{x}}(t)|} \right| &= \frac{1}{v_{\sigma(t)}} \sqrt{\dot{\theta}^2(t) + \sin^2 \theta(t) \dot{\phi}^2(t)} \\ &=: \kappa_{\theta(t), \sigma(t)} (\dot{\theta}(t), \dot{\phi}(t)), \quad \sigma(t) = \pm 1. \end{aligned}$$

Thus, the kinetic equations for forward and reverse motions become

$$\begin{aligned} \dot{\mathbf{z}}(t) &= f_{\sigma(t)}(\mathbf{z}(t), \hat{a}(t), \hat{b}(t)) \\ &:= \left[v_{\sigma(t)} \sin \theta(t) \cos \phi(t), v_{\sigma(t)} \sin \theta(t) \sin \phi(t), v_{\sigma(t)} \cos \theta(t), \hat{a}(t), \hat{b}(t) \right]^T, \end{aligned}$$

with controls $\hat{a}(\cdot), \hat{b}(\cdot)$ restricted under the curvature constraint

$$(17) \quad \kappa_{\theta(t), \sigma(t)}(\hat{a}(t), \hat{b}(t)) = \frac{1}{v_{\sigma(t)}} \sqrt{\hat{a}^2(t) + \sin^2 \theta(t) \hat{b}^2(t)} \leq \frac{1}{\rho_{\sigma(t)}}$$

for $\sigma(t) \in \{\pm 1\}$. Note that if $\sin \theta \neq 0$, then $\kappa_{\theta, \sigma}(\cdot, \cdot)$ is a norm in \mathbb{R}^2 , with a dual norm

$$\kappa_{\theta, \hat{\sigma}}^*(x, y) := \max_{\kappa_{\theta, \hat{\sigma}}(\hat{a}, \hat{b}) \leq 1} \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = v_{\hat{\sigma}} \sqrt{x^2 + \frac{y^2}{\sin^2 \theta}}$$

The derivation of the Hamilton-Jacobi equation from the dynamic programming principle is nearly identical to the two dimensional case. In three dimensions, (14) becomes

$$\begin{aligned} 0 &= \max_{\hat{\sigma} \in \{\pm 1\}, \kappa_{\theta, \hat{\sigma}}(\hat{a}, \hat{b}) \leq 1/\rho_{\hat{\sigma}}} \left\{ -\hat{\sigma} v_{\hat{\sigma}} (\sin \theta \cos \phi u_x + \sin \theta \sin \phi u_y + \cos \theta u_z) - \hat{a} u_{\phi} - \hat{b} u_{\theta} \right\} - 1 \\ &= \max_{\hat{\sigma} \in \{\pm 1\}} \left\{ -\hat{\sigma} v_{\hat{\sigma}} (\sin \theta \cos \phi u_x + \sin \theta \sin \phi u_y + \cos \theta u_z) + \max_{\kappa_{\theta, \hat{\sigma}}(\hat{a}, \hat{b}) \leq 1/\rho_{\hat{\sigma}}} \begin{bmatrix} -\hat{a} \\ -\hat{b} \end{bmatrix} \cdot \begin{bmatrix} u_{\theta} \\ u_{\phi} \end{bmatrix} \right\} - 1. \end{aligned}$$

If $\sin \theta \neq 0$, the last term is equivalent to the (scaled) dual norm expression of $\kappa_{\theta, \hat{\sigma}}$:

$$(18) \quad \begin{aligned} 0 &= \max_{\hat{\sigma} \in \{\pm 1\}} \left\{ -\hat{\sigma} v_{\hat{\sigma}} (\sin \theta \cos \phi u_x + \sin \theta \sin \phi u_y + \cos \theta u_z) + \frac{1}{\rho_{\hat{\sigma}}} \kappa_{\theta, \hat{\sigma}}^*(u_{\theta}, u_{\phi}) \right\} - 1 \\ &= \max_{\hat{\sigma} \in \{\pm 1\}} \left\{ -\hat{\sigma} v_{\hat{\sigma}} (\sin \theta \cos \phi u_x + \sin \theta \sin \phi u_y + \cos \theta u_z) + \frac{v_{\hat{\sigma}}}{\rho_{\hat{\sigma}}} \sqrt{u_{\theta}^2 + \frac{u_{\phi}^2}{\sin^2 \theta}} \right\} - 1. \end{aligned}$$

Remark 1. In general, if the Hamilton-Jacobi-Bellman equation is linear in a control, the maximizing controls should be evaluated at the extremal points. This is known as the *bang-bang principle* [SVV65]. For example, (14) is linear in \hat{a} , so \hat{a} should be evaluated as per (15). Intuitively, choosing \hat{a} to be -1 or $+1$ is equivalent to making a “full right” or a “full left” turn, respectively; in this sense, the Hamilton-Jacobi formulation described above can be seen as an Eulerian counterpart to the Lagrangian scheme proposed in [BL93] (see section 1.1).

Remark 2. While the solution sought by invoking the bang-bang principle is optimal in free space [Dub57, RS90], it may be suboptimal near the obstacle boundaries. For example, consider the two dimensional case in the presence of an obstacle with a boundary of curvature less than $1/\rho_{\pm 1}$. If the true optimal path traces along this boundary, the control must satisfy $|\hat{a}| < 1$, hence does not obey the bang-bang principle. Indeed, the correct solution near such obstacle boundaries would be characterized by the Hamilton-Jacobi-Bellman equation, which does not assume the bang-bang principle. See section 4.5.5 for numerical implications of this issue.

2.3. Examples. In light of the equations (16) and (18) we present the Hamilton-Jacobi equations for several well known models of curvature constrained motion, in two and three dimensions.

Example 3. Dubins’ Car [Dub57, TTSL10]. *The parameters are $v_1 = 1$, $v_{-1} = 0$, $\rho_1 = \rho > 0$ (p_{-1} is irrelevant). By virtue of the equation, the maximum in (16) must be evaluated at $\sigma = 1$ (if $\sigma = -1$, then the equation becomes $0 = -1$). Thus*

$$(19) \quad 0 = -\cos(\theta)u_x - \sin(\theta)u_y + \frac{1}{\rho}|u_\theta| - 1.$$

Example 4. Reeds-Shepp’s Car [RS90]. *The parameters are $v_1 = v_{-1} = 1$, $\rho_1 = \rho_{-1} = \rho > 0$. Thus (16) becomes,*

$$(20) \quad \begin{aligned} 0 &= \max_{\sigma=\pm 1} \{-\sigma(\cos \theta u_x + \sin \theta u_y)\} + \frac{1}{\rho}|u_\theta| - 1 \\ &= |\cos \theta u_x + \sin \theta u_y| + \frac{1}{\rho}|u_\theta| - 1. \end{aligned}$$

For the Reeds-Shepp’s car (20), since

$$|\cos \theta u_x + \sin \theta u_y| = |\cos(\theta + \pi) u_x + \sin(\theta + \pi) u_y|,$$

the value function $u(x, y, \theta)$ is π -periodic. Indeed, a vehicle with equal forward and reverse turning radii and speeds, will have the same optimal arrival time starting at the poses (x, y, θ) and $(x, y, \theta + \pi)$. From the point of view of numerics, one needs only to compute the value function for $[0, \pi)$ in the θ domain, with periodic boundary conditions.

Example 5. Dubins’ Plane [CL07]. *The turning radius constraint is enforced as in the Dubins car, but has an additional independent bounded constraint over speed in the z direction: $|\dot{z}| \leq c$ for some $c > 0$. This is a simple generalization to (19), with an additional state variable z :*

$$0 = -\cos \theta u_x - \sin \theta u_y + c|u_z| + \frac{1}{\rho}|u_\theta| - 1.$$

Example 6. Dubins' Missile. We can model a vehicle that moves forward in space at unit speed, with a turning radius constraint in all directions. This is another three dimensional counterpart to Dubins' car. Thus, for $\sin \theta \neq 0$,

$$0 = -\sin \theta \cos \phi u_x - \sin \theta \sin \phi u_y - \cos \theta u_z + \frac{1}{\rho} \sqrt{u_\theta^2 + \frac{u_\phi^2}{\sin^2 \theta}} - 1.$$

As pointed out in [RS90], this model can also be useful for plumbers who wish to connect two existing fixed pipe ends by pipes that can be bent, but not too quickly so that the curvature is bounded.

3. VISCOSITY SOLUTIONS

In this section, we present a theoretical treatment of the value function as a weak solution to the Hamilton-Jacobi equations. The solution is in general a non-smooth function, so typically a unique weak solution, called a *viscosity solution* [CL83] is sought. See section 3.1 for the definition of a viscosity solution. We discuss the solution for the Hamilton-Jacobi equations in two dimensions (16) in the viscosity solution framework. We show that the value function to the Reeds-Shepp's car problem is continuous, while the value function of Dubins' car is not. Consequently, we take different routes for describing the solutions, see sections 3.2 and 3.3. For a more rigorous and comprehensive treatment of Hamilton-Jacobi equations for control problems, see [BCD97].

3.1. Definition of a viscosity solution. In this section, we keep the equation in the form prior to invoking the bang-bang control:

$$(21) \quad 0 = H(\mathbf{y}, Dw(\mathbf{y})) = \max_{\sigma \in \{\pm 1\}, a \in [-1, 1]} \{-f(\mathbf{y}, \sigma, a) \cdot Dw(\mathbf{y})\} - 1.$$

While the proofs work for a general class of dynamics, for presentation purposes we assume the dynamics in the context of the two dimensional curvature constrained motion:

$$(22) \quad f(\mathbf{y}, \sigma, a) = v_\sigma(\mathbf{y}) \left[\sigma \cos \theta, \sigma \sin \theta, \frac{a}{\rho_\sigma(\mathbf{y})} \right]^T, \quad \mathbf{y} = (x, y, \theta).$$

The boundary condition coupling the PDE (21) is

$$(23) \quad w(\mathbf{y}) = 0 \quad \text{for } \mathbf{y} \in \mathcal{T}.$$

We now define viscosity solutions to the Hamilton-Jacobi equation (21).

Definition 7. We say that a continuous function w is a *viscosity subsolution* of (21) if for any $\phi \in C^1(\Omega)$ such that \mathbf{y} is a local maximum of $w - \phi$, then

$$H(\mathbf{y}, D\phi(\mathbf{y})) \leq 0.$$

We say that a function w is a *viscosity supersolution* of (21) if for any $\phi \in C^1(\Omega)$ such that \mathbf{y} is a local minimum of $w - \phi$, then

$$H(\mathbf{y}, D\phi(\mathbf{y})) \geq 0.$$

If w is both a viscosity subsolution and a viscosity supersolution of (21) then we say that w is a *viscosity solution* of (21).

3.2. Small time local controllability. The key difference between a Dubins' car and a Reeds-Shepp's car is the former lacks the maneuverability of the latter. This intuition will be made precise, by means of *reachability sets* and the concept of *small-time local controllability*. Define the reachability set from $\mathbf{y} \in \Omega$ within time $t > 0$ as

$$(24) \quad \mathcal{R}(\mathbf{y}; t) := \{\mathbf{z}(\tau) \mid \mathbf{z}(\cdot) \in \mathcal{A}_{\mathbf{y}}, 0 \leq \tau < t\} \quad t > 0.$$

Definition 8. The dynamics f is *small-time local controllable*, or STLC, at the point \mathbf{y} if

$$\mathbf{y} \in \text{int } \mathcal{R}(\mathbf{y}; t) \quad \text{for all } t > 0.$$

In other words, the given dynamics is STLC if the vehicle can be steered to anywhere in an small enough neighborhood about the starting pose. The main result for the STLC dynamics problem is that the value function u solves (21) in the viscosity sense, provided STLC holds at each point in the domain:

Theorem 9. *If the dynamics f is STLC in Ω , then the value function u is the viscosity solution to (21) in Ω .*

Reeds-Shepp's car is STLC (see discussion later), therefore its value function is the unique, continuous viscosity solution to (20). Furthermore, the same argument can be applied to a curvature constrained motions with positive forward and reverse speeds. We will see in the next section that characterizing a unique solution to (21) and (23) becomes subtle if STLC fails and u is discontinuous.

To prove Theorem 9, we begin by showing that STLC implies the continuity of the value function, which is of independent interest.

Lemma 10. *If the dynamics f is STLC, then the value function u is continuous.*

Proof. Fix $\mathbf{y} \in \Omega$. By the definition of STLC, for each $t > 0$, there exists $r(t) > 0$ such that

$$B_{r(t)}(\mathbf{y}) \subset \mathcal{R}(\mathbf{y}; t), \quad \text{for all } t > 0.$$

Choose an arbitrary $\mathbf{y}_0 \in \Omega$ so that $|\mathbf{y} - \mathbf{y}_0| < r(t)$. Then \mathbf{y}_0 is reachable from \mathbf{y} within time t . Thus,

$$u(\mathbf{y}_0) \leq u(\mathbf{y}) + t.$$

Swapping the roles of \mathbf{y} and \mathbf{y}_0 , we have $|u(\mathbf{y}_0) - u(\mathbf{y})| < t$. □

Next we show that STLC is a sufficient condition to admit the value function as a viscosity solution of the Hamilton-Jacobi equation.

Proof of Theorem 9. We first show that u is a subsolution. Take $\phi \in C^1(\Omega)$ and let \mathbf{y} be a local maximum point of $u - \phi$. Let $\mathbf{z}(\cdot) \in \mathcal{A}_{\mathbf{y}}$ be a path with arbitrary constant controls $\sigma^* \in \{\pm 1\}$ and $a^* \in [-1, 1]$. Then, for $t > 0$ small enough, since u is continuous by Lemma 10, we have $u(\mathbf{y}) - \phi(\mathbf{y}) \geq u(\mathbf{z}(t)) - \phi(\mathbf{z}(t))$ so

$$u(\mathbf{y}) - u(\mathbf{z}(t)) \geq \phi(\mathbf{y}) - \phi(\mathbf{z}(t)).$$

Since $u(\mathbf{y}) \leq u(\mathbf{z}(t)) + t$ by the dynamic programming principle, we have,

$$\frac{\phi(\mathbf{y}) - \phi(\mathbf{z}(t))}{t} \leq \frac{u(\mathbf{y}) - u(\mathbf{z}(t))}{t} \leq 1.$$

Taking $t \searrow 0$, we have

$$-D\phi(\mathbf{y}) \cdot f(\mathbf{y}, \sigma^*, a^*) \leq 1;$$

since σ^*, a^* were arbitrary, taking the maximum over all controls implies that $H(\mathbf{y}, D\phi(\mathbf{y})) \leq 0$.

Next, to show that u is a supersolution, assume that \mathbf{y} is a local minimum point of $u - \phi$. Then as before, for any path $\mathbf{z}(\cdot) \in \mathcal{A}_{\mathbf{y}}$, and small enough $t > 0$,

$$u(\mathbf{y}) - u(\mathbf{z}(t)) \leq \phi(\mathbf{y}) - \phi(\mathbf{z}(t)).$$

Then for any $\epsilon > 0$, there exists (non-constant) controls $\tilde{\sigma}(\cdot) \in \{\pm 1\}, \tilde{a}(\cdot) \in [-1, 1]$, such that the corresponding path $\tilde{\mathbf{y}}(\cdot)$ satisfies

$$u(\mathbf{x}) > t + u(\tilde{\mathbf{y}}(t)) - \epsilon,$$

Therefore,

$$\frac{\phi(\mathbf{y}) - \phi(\tilde{\mathbf{y}}(t))}{t} \geq \frac{u(\mathbf{y}) - u(\tilde{\mathbf{y}}(t))}{t} > 1 - \epsilon.$$

Taking $t \searrow 0$, we have

$$-D\phi(\mathbf{y}) \cdot f(\mathbf{y}, \tilde{\sigma}(0), \tilde{a}(0)) > 1 - \epsilon.$$

By taking $\epsilon \searrow 0$, we conclude that $H(\mathbf{y}, D\phi(\mathbf{x})) \geq 0$, as desired. \square

It is straightforward to see that Reeds-Shepp's car is STLC. While this was proved using Lie algebra in [ST91], perhaps a more intuitive argument is that a Reeds-Shepp's car can change its angle of direction by $\Delta\theta$ in time $\mathcal{O}(\Delta\theta)$ while staying within an arbitrarily small neighborhood in Ω_0 . Such a feat is not possible for a Dubins' car. For the Reeds-Shepp's car, this is achieved by setting the controls $a(t) = \text{sgn}(\Delta\theta)$ and $\sigma(t)$ to alternately 'switch' between ± 1 in contiguous arbitrarily small intervals of t . Thus, a Reeds-Shepp's car can maneuver like a car with zero turning radius (i.e. an omnidirectional car, which is clearly STLC) modulo $\mathcal{O}(\Delta\theta)$ extra time, to change its direction by $\Delta\theta$.

Furthermore, the preceding argument can be easily generalized for cars with $v_1, v_{-1} > 0$. For velocities that vary in Ω , as long as $\inf_{\mathbf{y}} v_1(\mathbf{y}), \inf_{\mathbf{y}} v_{-1}(\mathbf{y}) > \delta > 0$, STLC still holds.

3.3. Discontinuous value function: Dubins' car. If one of v_1 or v_{-1} is zero, STLC no longer holds. For instance, for $\epsilon > 0$ small, a Dubins' car with minimum turning radius ρ starting at pose $(0, \epsilon, 0)$ is required to make a full turn to arrive at $\mathcal{T} = \{(0, 0, 0)\}$, so $u(0, \epsilon, 0) > 2\rho\pi$ for all $\epsilon > 0$. But since $u(0, 0, 0) = 0$, the value function u for the Dubins' car is discontinuous at \mathcal{T} and the dynamics is not STLC. Without continuity, u cannot be a viscosity solution as per Definition 7.

While there are several notions for discontinuous viscosity solutions [GS01, TGO03, CS02, Ish87], we chose to exploit *e-solutions* [BCD97], which are particularly suited for control problems and enjoys existence and uniqueness properties. Let us begin by defining subsolutions and supersolutions of a boundary value problem.

Definition 11. We say that a bounded upper semicontinuous (resp. lower semicontinuous) function w is a *subsolution* (resp. *supersolution*) of the boundary value problem (21) and (23) if w is a viscosity subsolution (resp. viscosity supersolution) and $w \leq 0$ (resp. $w \geq 0$) at \mathcal{T} .

Let \mathcal{S} and \mathcal{Z} be the set of all subsolutions and supersolutions, respectively, to (21) and (23). Since a continuous function is both lower and upper semicontinuous, a viscosity solution w of (21) that satisfies the boundary condition (23) is a subsolution and a supersolution. Thus, in this case, $w \in \mathcal{S} \cap \mathcal{Z}$. However, for discontinuous value functions u

(e.g. for dynamics that are not STLC), it is too strong to demand that u_* to be a supersolution and u^* to be a subsolution; for example, for the Dubins' car with $\mathcal{T} = \{(0, 0, 0)\}$, $u^* \geq 2\rho\pi \not\leq 0$ at \mathcal{T} .

To achieve both existence and uniqueness of a weak solution for such a boundary value problems, a particular viscosity sub- and supersolution must be sought.

Definition 12. We say w is an *e-subsolution* of (21) and (23) if there exists $\emptyset \neq \mathcal{S}(w) \subset \mathcal{S}$ such that

$$(25) \quad w(\mathbf{y}) = \sup_{g \in \mathcal{S}(w)} g(\mathbf{y}) \quad \mathbf{y} \in \Omega.$$

We say w is an *e-supersolution* of (21) and (23) if there exists $\emptyset \neq \mathcal{Z}(w) \subset \mathcal{Z}$ such that

$$(26) \quad w(\mathbf{y}) = \inf_{g \in \mathcal{Z}(w)} g(\mathbf{y}) \quad \mathbf{y} \in \Omega.$$

Furthermore, if w is an e-subsolution and an e-supersolution of (21) and (23), we say it is an *e-solution* of (21) and (23).

The main claim in this section is that the value function u associated with the dynamics (22) is the unique e-solution to the boundary value problem (21) and (23).

Theorem 13. *The value function u associated with the dynamics (22) is the e-solution of the problem (21) and (23).*

To prove the preceding theorem, we proceed with a sequence of lemmas. Assume that u is the value function associated with the dynamics (22). We adopt the strategy of [BCD97] by showing that u_* is both an e-subsolution and an e-supersolution of (21) and (23). Then the main claim follows by observing that $u = u_*$, since u is lower semi-continuous.

Lemma 14. *u_* is a supersolution of the problem (21) and (23).*

Proof. Since $u_* = u = 0 \geq 0$ at \mathcal{T} , we are left to prove that u_* is a viscosity supersolution to (21). Let $\mathbf{y} \in \Omega$ be a local minimum of $u_* - \phi$, for some $\phi \in C^1(\Omega)$; without the loss of generality, assume $u_*(\mathbf{y}) = \phi(\mathbf{y})$. Towards a contradiction, suppose

$$H(\mathbf{y}, D\phi(\mathbf{y})) < 0.$$

Then by the continuity of f and $D\phi$, there exists $\epsilon > 0$ such that

$$H(\tilde{\mathbf{y}}, D\phi(\tilde{\mathbf{y}})) \leq -\epsilon < 0, \quad \text{for all } \tilde{\mathbf{y}} \in B_\epsilon(\mathbf{y}).$$

Furthermore, there exists $t > 0$ such that for any $\tilde{\mathbf{y}} \in B_{\epsilon/2}(\mathbf{y})$ and any path $\mathbf{z}(\cdot) \in \mathcal{A}_{\tilde{\mathbf{y}}}$,

$$\mathbf{z}(s) \in B_\epsilon(\mathbf{y}) \quad \forall s \in [0, t].$$

Fix such at t and let $\delta = \epsilon t/2$. From the dynamic programming principle,

$$u(\mathbf{y}) > u(\mathbf{z}(t)) + t - \delta.$$

Since $\frac{d}{ds}\phi(\mathbf{z}(s)) = D\phi(\mathbf{z}(s)) \cdot f(\mathbf{z}(s), a(s))$, we have

$$\begin{aligned} \phi(\mathbf{z}(s)) &= \int_0^t D\phi(\mathbf{z}(s)) \cdot f(\mathbf{z}(s), a(s)) ds + \phi(\tilde{\mathbf{y}}) \\ &\geq - \int_0^t H(\mathbf{z}(s), D\phi(\mathbf{z}(s))) + 1 ds + \phi(\tilde{\mathbf{y}}) \\ &\geq \epsilon t - t + \phi(\tilde{\mathbf{y}}). \end{aligned}$$

Then

$$\begin{aligned}
 u(\tilde{\mathbf{y}}) - \phi(\tilde{\mathbf{y}}) &> u(\mathbf{z}(t)) + t - \delta + \epsilon t - t - \phi(\mathbf{z}(t)) \\
 &\geq u_*(\mathbf{z}(t)) - \phi(\mathbf{z}(t)) - \delta + \epsilon t \\
 &\geq -\delta + \epsilon t \\
 &= \delta.
 \end{aligned}$$

Thus,

$$u_*(\mathbf{y}) - \phi(\mathbf{y}) = \liminf_{\tilde{\mathbf{y}} \rightarrow \mathbf{y}} (u(\tilde{\mathbf{y}}) - \phi(\tilde{\mathbf{y}})) \geq \delta,$$

which contradicts the assumption that $u_*(\mathbf{y}) = \phi(\mathbf{y})$. \square

Lemma 15. u_* is an ϵ -supersolution of the problem (21) and (23).

Proof. This follows immediately from the conclusion of Lemma 14; let $\mathcal{Z}(u_*) = \{u_*\}$, then (26) trivially holds. \square

Lemma 16. u_* is an ϵ -subsolution of the problem (21) and (23).

Sketch of proof. The proof makes use of the ϵ -perturbed problem [Sor93], which has a value function u_ϵ that is a subsolution. Then we set $\mathcal{S}(u_*) = \{u_\epsilon\}_{\epsilon > 0}$ to be the class of subsolutions such that (25) holds.

Begin by introducing the dynamics \tilde{f} defined by

$$\tilde{f}(\mathbf{y}, \sigma, a) = \begin{cases} f(\mathbf{y}, \sigma, a) & \sigma \in \{\pm 1\}, |a| \leq 1/\rho \\ 0 & \sigma = 0. \end{cases}$$

By extending f to include the control $\sigma = 0$ of null dynamics yields a modified Hamiltonian

$$\begin{aligned}
 \tilde{H}(\mathbf{y}, \mathbf{p}) &:= \max_{\sigma \in \{0, \pm 1\}, a \in [-1, 1]} \left\{ -\tilde{f}(\mathbf{y}, \sigma, a) \cdot \mathbf{p} \right\} - 1 \\
 &= \max \left\{ \max_{\sigma \in \{\pm 1\}, a \in [-1, 1]} \left\{ -f(\mathbf{y}, \sigma, a) \cdot \mathbf{p} \right\}, 0 \right\} - 1.
 \end{aligned}$$

The value function for the dynamics \tilde{f} coincides with the original value function with the dynamics f , since the null dynamics control $\sigma = 0$ would never be invoked for an optimal control. Given $\epsilon > 0$, consider the optimal control problem, called the ϵ -perturbed problem, with the kinetic equation

$$\dot{\mathbf{y}}_\epsilon(t) = f_\epsilon(\mathbf{y}(t), \sigma(t), a(t), \beta) := \tilde{f}(\mathbf{y}(t), \sigma(t), a(t)) + \epsilon\beta(t)$$

for $\beta(t) \in \{(x, y, 0) \mid (x, y) \in B_1(0, 0)\}$. Then the resulting Hamilton-Jacobi equation is

$$\begin{aligned}
 (27) \quad H_\epsilon(\mathbf{y}, Dw(\mathbf{y})) &:= \tilde{H}(\mathbf{y}, Dw(\mathbf{y})) + \epsilon|D_{\mathbf{x}}w| = 0 \quad \text{note: } \mathbf{y} = (\mathbf{x}, \theta) \\
 w(\mathbf{y}) &= 0 \quad \text{for } \mathbf{y} \in \mathcal{T}.
 \end{aligned}$$

Denote by u_ϵ the value function associated with the dynamics f_ϵ . Since f_ϵ is STLC, u_ϵ is the unique (continuous) viscosity solution for (27). Then,

$$\begin{aligned} 0 &= \tilde{H}(\mathbf{y}, Du_\epsilon(\mathbf{y})) + \epsilon |D_{\mathbf{x}} u_\epsilon(\mathbf{y})| \\ &\geq \tilde{H}(\mathbf{y}, Du_\epsilon(\mathbf{y})) \\ &\geq \max_{\sigma \in \{\pm 1\}, a \in [-1, 1]} \{-f(\mathbf{y}, \sigma, a) \cdot Du_\epsilon(\mathbf{y})\} - 1 \\ &= H(\mathbf{y}, Du_\epsilon(\mathbf{y})), \end{aligned}$$

so u_ϵ is a viscosity subsolution of (21). Furthermore, since $u_\epsilon = 0 \leq 0$ at \mathcal{T} , it is a subsolution.

What is remaining to show is the property (25) for $\mathcal{S}(u_*) = \{u_\epsilon\}_{\epsilon > 0}$. Note that $u_\epsilon(\mathbf{y}) \leq u(\mathbf{y})$ by virtue of ϵ perturbed problem having more control (and thus, more admissible paths). The idea is, as $\epsilon \rightarrow 0$, the speed associated with the omnidirectional control $\beta(t)$ of the perturbed problem tends to zero, therefore the value function u_ϵ converges to the original u . We omit the details and refer the interested reader to [BCD97, Theorem 3.10]. \square

Proof of Theorem 13. Immediately, from Lemmas 15 and 16, we have that u_* is the unique e-solution. Furthermore, the dynamics $f(\mathbf{y}, \sigma, a)$ is linear in $\sigma \in \{\pm 1\}$ and $a \in [-1, 1]$ for every \mathbf{y} , therefore, u is lower semicontinuous (see [HL69]). Thus, $u = u_*$ and u is an e-solution. \square

4. NUMERICAL IMPLEMENTATION

In this section, we describe two numerical schemes to solve the Hamilton-Jacobi equation for curvature constrained motions in two dimensions (16) and show numerical results. The three dimensional case works analogously.

First, in section 4.1, a monotone finite difference scheme is described for solving continuous value functions. Next, in section 4.2 we present a semi-Lagrangian scheme for the discontinuous case. This is followed, in section 4.3, by a description of how optimal paths are computed from a given (numerical) value function. We dedicate section 4.4 on the treatment of obstacles, required for accurate computations near obstacle boundaries. Finally, in section 4.5, we demonstrate the performance of the aforementioned schemes under various settings.

4.1. Continuous value functions: finite difference scheme. In order to compute the value function of (16), we exploit the results of [CL84, BS91] that monotone, consistence schemes converge to the (continuous) viscosity solution. In the case of discontinuous viscosity solutions, such as the Dubins' car, we use a different approach, see section 4.2.

Set up a three dimensional uniform Cartesian grid $\{(x_i, y_j, \theta_k)\}$, $i = 1, \dots, N_x, j = 1, \dots, N_y, k = 1, \dots, N_\theta$, with grid refinements (h_x, h_y, h_θ) . Let $u_{i,j,k}$ be the approximation of the solution at the grid node (x_i, y_j, θ_k) . Denote $\xi_k := \text{sgn}(\cos \theta_k)$ and $\nu_k := \text{sgn}(\sin \theta_k)$. The finite difference construction of the scheme is as follows: the

spatial derivative terms in x and y are discretized in the upwind sense,

$$(28) \quad \begin{aligned} (\cos \theta u_x)_{ijk} &= |\cos \theta_k| \frac{u_{i+\xi_k, j, k} - u_{ijk}}{h_x}, \\ (\sin \theta u_y)_{ijk} &= |\sin \theta_k| \frac{u_{i+\nu_k, j, k} - u_{ijk}}{h_y}, \end{aligned}$$

and the derivative in θ with a monotone discretization

$$(29) \quad (|u_\theta|)_{ijk} = \max \left\{ \frac{u_{ijk} - u_{i, j, k+1}}{h_\theta}, \frac{u_{ijk} - u_{i, j, k-1}}{h_\theta}, 0 \right\}.$$

To approximate (16) with the discretization (28) and (29), we solve for u_{ijk} for the cases $\sigma = \pm 1$ and $(|u_\theta|)_{ijk}$ zero or non-zero. Assume for simplicity that $h_x = h_y =: h$. We treat the cases $\sigma = +1$ and -1 separately: for the case $\sigma = 1$, if $(|u_\theta|)_{ijk} \neq 0$,

$$(30) \quad \begin{aligned} u_{ijk} &= \frac{\frac{h}{v_1} + |\cos \theta_k| u_{i+\xi_k, j, k} + |\sin \theta_k| u_{i, j+\nu_k, k} + \frac{h}{\rho_1 h_\theta} \min \{u_{i, j, k \pm 1}\}}{|\cos \theta_k| + |\sin \theta_k| + \frac{h}{\rho_1 h_\theta}} \\ &=: G_{ijk}^{+1} \end{aligned}$$

and if $(|u_\theta|)_{ijk} = 0$,

$$(31) \quad u_{ijk} = \frac{\frac{h}{v_1} + |\cos \theta_k| u_{i+\xi_k, j, k} + |\sin \theta_k| u_{i, j+\nu_k, k}}{|\cos \theta_k| + |\sin \theta_k|} =: F_{ijk}^{+1}.$$

For the case $\sigma = -1$, if $(|u_\theta|)_{i, j, k} \neq 0$,

$$(32) \quad \begin{aligned} u_{ijk} &= \frac{\frac{h}{v_{-1}} + |\cos \theta_k| u_{i-\xi_k, j, k} + |\sin \theta_k| u_{i, j-\nu_k, k} + \frac{h}{\rho_{-1} h_\theta} \min \{u_{i, j, k \pm 1}\}}{|\cos \theta_k| + |\sin \theta_k| + \frac{h}{\rho_{-1} h_\theta}} \\ &=: G_{ijk}^{-1}, \end{aligned}$$

and if $(|u_\theta|)_{ijk} = 0$,

$$(33) \quad u_{ijk} = \frac{\frac{h}{v_1} + |\cos \theta_k| u_{i-\xi_k, j, k} + |\sin \theta_k| u_{i, j-\nu_k, k}}{|\cos \theta_k| + |\sin \theta_k|} =: F_{ijk}^{-1}.$$

We solve the system of nonlinear equations (30)-(33) for $\{u_{ijk}\}$ via the fast sweeping method [Zha04, TCOZ03]: initially set

$$u_{ijk}^0 = \begin{cases} 0 & \text{if } (x_i, y_j, \theta_k) \in \mathcal{T} \\ \infty & \text{otherwise.} \end{cases}$$

for all (i, j, k) , and compute u_{ijk}^n for $n = 1, 2, \dots$ iteratively according to the update scheme

$$(34) \quad u_{ijk}^{n+1} = \min \left\{ G_{ijk}^{\pm 1}, F_{ijk}^{\pm 1}, u_{ijk}^n \right\}.$$

by a Gauss-Seidel sweeping scheme; in each iteration, the grid nodes are visited by eight different orderings, ascending and descending in each coordinates i, j, k . Then, u_{ijk}^n converges to the solution of the nonlinear equations (30)-(33) as $n \rightarrow \infty$. The above discretization is monotone in the sense that the update formula for u_{ijk}^{n+1} is a non-decreasing function of the neighboring values $u_{i \pm 1, j+1, k}^n, u_{i \pm 1, j-1, k}^n, u_{i, j, k \pm 1}^n$. Thus, for continuous

value function, the computed solution converges to the viscosity solution of the PDE as $h_x, h_y, h_\theta \rightarrow 0$.

4.2. Discontinuous value functions: semi-Lagrangian scheme. While a similar finite difference discretization gives somewhat reasonable approximations of the value function (c.f. [TTSL10]), we resort to a *semi-Lagrangian scheme*, which is known to have convergence properties even for Hamilton-Jacobi equations with discontinuous solutions.

The idea of semi-Lagrangian scheme is to directly discretize the dynamic programming principle (13) on an Eulerian (in our case Cartesian) grid, for a short time comparable to the grid refinement. We shall describe the method for the case

$$\begin{aligned} v_1 &= v > 0, & \rho_1 &= \rho > 0, \\ v_{-1} &= \rho_{-1} = 0, \end{aligned}$$

i.e. Dubins' car (Example 3) with speed v and turning radius ρ . The value function is approximated at the grid node $\mathbf{y}_{i,j,k} = (x_i, y_i, \theta_k)$. Denote $\mathbf{z}_{a(\cdot)}(t)$ to be the path along the characteristic curves, with the control $a(\cdot)$ and initial position $\mathbf{y}_{i,j,k}$:

$$(35) \quad \dot{\mathbf{z}}_{a(\cdot)}(t) = -v \left[\cos \theta(t), \sin \theta(t), \frac{a(t)}{\rho} \right]^T, \quad a(t) \in [-1, 1]$$

$$(36) \quad \mathbf{z}_{a(\cdot)}(0) = \mathbf{y}_{i,j,k}.$$

Choose a (small) time step $\Delta t > 0$. We assume that $\mathbf{z}_{a(\cdot)}(t)$ in free space for $0 \leq t \leq \Delta t$, thus the optimal control is bang-bang. For Δt small enough, we may further assume that the optimal control is

$$a^*(t) = \begin{cases} a_R = 1, & \text{(right bang)} \\ a_L = -1, & \text{(left bang)} \\ a_0 = 0, & \text{(no bang, move straight)} \end{cases}$$

for all $0 \leq t \leq \Delta t$. The dynamic programming principle at the point $\mathbf{y}_{i,j,k}$ becomes,

$$(37) \quad u(\mathbf{y}_{i,j,k}) = \Delta t + \min_{a \in \{a_L, a_R, a_0\}} u(\mathbf{z}_a(\Delta t))$$

Since $\mathbf{z}_a(\Delta t)$ will typically not lie on a grid node, $u(\mathbf{z}_a(\Delta t))$ can be approximated by a (three dimensional) interpolation of nearby grid values of u .

By a particular choice of Δt , however, the interpolation can be made simpler. In our implementation, we let $v\Delta t = \rho h_\theta$, or equivalently

$$(38) \quad \Delta t = \frac{\rho h_\theta}{v}.$$

Then, by construction, the θ -components of $\mathbf{z}_{a_R}(\Delta t)$ and $\mathbf{z}_{a_L}(\Delta t)$ are precisely $\theta_{k+1} = \theta_k + h_\theta$ and $\theta_{k-1} = \theta_k - h_\theta$, respectively. Note that the θ -component of $\mathbf{z}_{a_0}(\Delta t)$ is θ_k . Since $\mathbf{z}_a(\Delta t)$ for each $a = a_L, a_0$, and a_R lie exactly on the θ grid nodes, namely θ_{k-1}, θ_k , and θ_{k+1} , respectively, an interpolation is only required in the x, y directions.

For completeness, we explicitly write $\mathbf{z}_a(\Delta t)$ for each $a \in \{a_L, a_R, a_0\}$ in the case Δt is chosen as in (38). Denote $\mathbf{Rot}(\theta)$ to be the rotation matrix by θ in two dimensions:

$$\mathbf{Rot}(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}.$$

For simplicity, we consider the case $h_x = h_y =: h$. For the choice of time step (38), we have

$$(39) \quad \begin{aligned} \mathbf{z}_{a_R}(\Delta t) &= (x_i + h\tilde{x}^R, y_i + h\tilde{y}^R, \theta_{k-1}), \\ \mathbf{z}_{a_L}(\Delta t) &= (x_i + h\tilde{x}^L, y_i + h\tilde{y}^L, \theta_{k+1}), \\ \mathbf{z}_{a_0}(\Delta t) &= (x_i + h\tilde{x}^0, y_i + h\tilde{y}^0, \theta_k), \end{aligned}$$

where,

$$(40) \quad \begin{aligned} \begin{bmatrix} \tilde{x}^R \\ \tilde{y}^R \end{bmatrix} &= \mathbf{Rot}(\theta_k) \begin{bmatrix} \rho \sin h_\theta \\ \rho(1 - \cos h_\theta) \end{bmatrix}, & \begin{bmatrix} \tilde{x}^L \\ \tilde{y}^L \end{bmatrix} &= \mathbf{Rot}(\theta_k) \begin{bmatrix} \rho \sin h_\theta \\ \rho(\cos h_\theta - 1) \end{bmatrix}, \\ \begin{bmatrix} \tilde{x}^0 \\ \tilde{y}^0 \end{bmatrix} &= \mathbf{Rot}(\theta_k) \begin{bmatrix} \rho h_\theta \\ 0 \end{bmatrix}. \end{aligned}$$

The interpolation of u at the points (39) can be achieved with higher accuracy with appropriate stencils, using techniques such as ENO [OS91]. We implemented the simplest case, linear interpolation using (x_i, y_j) and its eight closest neighboring grid nodes for each $\theta_k, \theta_{k\pm 1}$. For each point in (39), we approximate u at the point using the nearest four grid nodes (in (x, y)) among the nine grid nodes. For stability, we must choose h and h_θ so that the points in (39) be within the eight neighboring grid nodes about (x_i, y_j) , that is,

$$\rho h_\theta \leq h.$$

For problems with spatially varying turning radius $\rho(x)$, the restriction is

$$\sup_{x \in \Omega} \{\rho(x)\} h_\theta \leq h.$$

Remark 17. Bardi, Falcone and Soravia [BFS94, BFS99] have shown that the semi-Lagrangian scheme converges on compact sets as $\Delta t/h \rightarrow 0$, where the e-solution is continuous. In [KV10], however, numerical tests of control problems with discontinuous value functions have shown that convergence can still be obtained for $\Delta t/h \rightarrow c$, for a constant $c > 0$. We have found that the latter property holds for the Dubins' car value function also, see section 4.5.1.

4.3. Computing optimal paths. Once the value function has been approximated by u_{ijk} , individual optimal paths can be computed by tracing the characteristic curves towards the target \mathcal{T} . Suppose the initial pose is $(x(0), y(0), \theta(0)) = (x_0, y_0, \theta_0)$. We write $\mathbf{z}(t) = (x(t), y(t), \theta(t))$, the pose on the trajectory at time $t \geq 0$. Note that the characteristic curves are precisely the optimal trajectories, but in opposite directions. Thus, for the Dubins' car, the dynamical system describing the optimal path is

$$\begin{cases} \dot{x}(t) = \cos \theta(t) \\ \dot{y}(t) = \sin \theta(t) \\ \dot{\theta}(t) = -\frac{1}{\rho} \text{sgn}(u_\theta(\mathbf{z}(t))). \end{cases}$$

and for the Reeds-Shepp's car,

$$\begin{cases} \dot{x}(t) = \text{sgn}[\cos \theta(t)u_x(\mathbf{z}(t)) + \sin \theta(t)u_y(\mathbf{z}(t))] \cos \theta(t) \\ \dot{y}(t) = \text{sgn}[\cos \theta(t)u_x(\mathbf{z}(t)) + \sin \theta(t)u_y(\mathbf{z}(t))] \sin \theta(t) \\ \dot{\theta}(t) = -\frac{1}{\rho} \text{sgn}(u_\theta(\mathbf{z}(t))). \end{cases}$$

For the general case, the characteristics equations yields

$$(41) \quad \begin{cases} \dot{x}(t) = \sigma v_\sigma(\mathbf{z}(t)) \cos \theta(t) \\ \dot{y}(t) = \sigma v_\sigma(\mathbf{z}(t)) \sin \theta(t) \\ \dot{\theta}(t) = -\frac{v_\sigma(\mathbf{z}(t))}{\rho_\sigma(\mathbf{z}(t))} \operatorname{sgn}(u_\theta(\mathbf{z}(t))), \end{cases}$$

where $\mathbf{z}(t) = (x(t), y(t), \theta(t))$ and $\sigma = \sigma(\mathbf{z}(t))$ is defined as

$$\sigma(\mathbf{y}) = \arg \max_{\sigma=\pm 1} \left\{ -v_\sigma(\mathbf{y}) \sigma [\cos \theta u_x(\mathbf{y}) + \sin \theta u_y(\mathbf{y})] + \frac{v_\sigma(\mathbf{y})}{\rho_\sigma(\mathbf{y})} |u_\theta(\mathbf{y})| \right\},$$

and $\mathbf{y} = (\cdot, \cdot, \theta)$. Recall that σ determined whether to move the vehicle ‘forward’ ($\sigma = +1$) or ‘reverse’ ($\sigma = -1$). In the discrete setting, one can compute $\sigma_{ijk} \in \{\pm 1\}$ at each grid node (i, j, k) , in the process of computing the value function. The minimizer among $G_{ijk}^{+1}, F_{ijk}^{+1}$ and $G_{ijk}^{-1}, F_{ijk}^{-1}$ (recall their definitions in (30)-(33)) in the update scheme (34) determines the value of $\sigma_{ijk} \in \{\pm 1\}$. Explicitly, initially set $\sigma_{ijk}^0 = 0$, for all grid nodes (i, j, k) , and update according to

$$\sigma_{ijk}^{n+1} = \begin{cases} 1 & \text{if } \min \left\{ G_{ijk}^{+1}, F_{ijk}^{+1} \right\} \leq \min \left\{ G_{ijk}^{-1}, F_{ijk}^{-1}, u_{ijk}^n \right\} \\ -1 & \text{if } \min \left\{ G_{ijk}^{-1}, F_{ijk}^{-1} \right\} < \min \left\{ G_{ijk}^{+1}, F_{ijk}^{+1}, u_{ijk}^n \right\} \\ \sigma_{ijk}^n & \text{if } u_{ijk}^n < G_{ijk}^{\pm 1}, F_{ijk}^{\pm 1}. \end{cases}$$

To solve the characteristic equations we compute $(x(t + \Delta t), y(t + \Delta t), \theta(t + \Delta t))$ from $(x(t), y(t), \theta(t))$ by first approximating

$$\begin{aligned} u_x(x(t), y(t), \theta(t)) &= \frac{u(x(t) + \bar{h}_x, y(t), \theta(t)) - u(x(t) - \bar{h}_x, y(t), \theta(t))}{2\bar{h}_x} + \mathcal{O}(\bar{h}_x^2) \\ u_y(x(t), y(t), \theta(t)) &= \frac{u(x(t), y(t) + \bar{h}_y, \theta(t)) - u(x(t), y(t) - \bar{h}_y, \theta(t))}{2\bar{h}_y} + \mathcal{O}(\bar{h}_y^2) \\ u_\theta(x(t), y(t), \theta(t)) &= \frac{u(x(t), y(t), \theta(t) + \bar{h}_\theta) - u(x(t), y(t), \theta(t) - \bar{h}_\theta)}{2\bar{h}_\theta} + \mathcal{O}(\bar{h}_\theta^2), \end{aligned}$$

where the the value function is approximated at non-grid nodes via linear approximation; to ensure that the maximum possible speed is attained in either direction, we approximate $\sigma(x(t), y(t), \theta(t))$ by a nearest-neighbor approximation. Next, we solve (41) by the 4th order Runge-Kutta method, until $(x(t), y(t), \theta(t))$ reaches within a prescribed tolerance distance from \mathcal{T} . Where the value function is continuous, the computed paths converge to the analytical characteristic curves as $\{\bar{h}_x, \bar{h}_y, \bar{h}_\theta, \Delta t\} \sim \mathcal{O}(h_x, h_y, h_\theta) \rightarrow 0$ by virtue of the approximation schemes.

4.4. Treatment of obstacles. It is standard practice in the PDE optimal control community to implement obstacles in domains by setting the value function to be a very large number at grid nodes inside the obstacle. This technique produces decent results for typical problems in optimal control, such as the eikonal equation, see for example [Fal07]. This property is carried over to the schemes described in section 4.1, where the dynamics is STLC and the value function is continuous in Ω . However, for cases where the dynamics is not STCT and the value function is discontinuous, we have found that such implementation of the obstacles degrades the computed solutions (using the semi-Lagrangian scheme, section 4.2).

We illustrate the reason for such sensitivity by an example for the Dubins' car problem. Assume that the value function at the grid nodes in the obstacles are forced to be infinite (although in practice the values are set to a large number, the essence of the example is still applicable). Fix a grid node $X_0^k := (x_i, y_j, \theta_k)$ where θ_k is chosen so that $\mathbf{z}_{a_L}(\Delta t), \mathbf{z}_{a_R}(\Delta t), \mathbf{z}_{a_0}(\Delta t)$ all lie in the domain $(x_{i-1}, x_i) \times (y_j, y_{j+1}) \times (\theta_{k-1}, \theta_{k+1})$ but not in the obstacle, see Figure 2. Suppose that both the “west” and “north-west” neighboring grid nodes (assume as before that h is the spatial resolution in x)

$$X_W^k := (x_{i-1}, y_j, \theta_k) = (x_i - h, y_j, \theta_k),$$

$$X_{NW}^k := (x_{i-1}, y_{j+1}, \theta_k) = (x_i - h, y_j + h, \theta_k),$$

respectively, lie inside the obstacle as depicted in Figure 2, thus $u_{i-1,j,k} = u_{i-1,j+1,k} = \infty$ for all k . Also define, $X_N^k := (x_i, y_{j+1}, \theta_k) = (x_i, y_j + h, \theta_k)$, the “north” neighboring grid node to X_0^k . By the choice of θ_k , our semi-Lagrangian scheme implementation performs three bilinear interpolations, namely,

- (1) among $X_0^{k+1}, X_W^{k+1}, X_{NW}^{k+1}, X_N^{k+1}$ for approximating $u(\mathbf{z}_{a_R}(\Delta t))$,
- (2) among $X_0^{k-1}, X_W^{k-1}, X_{NW}^{k-1}, X_N^{k-1}$ for approximating $u(\mathbf{z}_{a_L}(\Delta t))$, and
- (3) among $X_0^k, X_W^k, X_{NW}^k, X_N^k$ for approximating $u(\mathbf{z}_{a_0}(\Delta t))$.

Since all three interpolations involve the value function evaluated at grid nodes in the obstacle, the approximations will be infinite. Therefore, the update formula (37) assigns the value function at X_0^k to also be infinite. But clearly, the value function at X_0^k should be finite! It is clear that the finite difference approximation will yield the same result. Hence, very restricted dynamics are permitted for reasonable approximations on grids near the obstacle boundaries.

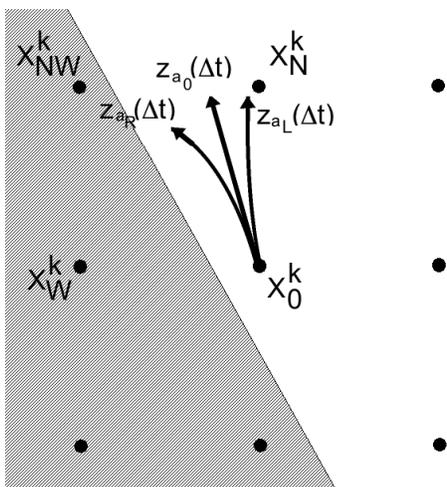


FIGURE 2. Illustration of the example. Black dots are the grid nodes projected on the (x, y) plane, and the shaded area represent the obstacle. The tips of the three arrows are the locations $\mathbf{z}_{a_L}(\Delta t), \mathbf{z}_{a_R}(\Delta t), \mathbf{z}_{a_0}(\Delta t)$.

Rather than forcing the value function to be infinite in the obstacles, we take a more natural and versatile approach to modeling state constraints: we interpret obstacles as

regions where *the speed is zero (or close to zero)* or, equivalently, where *the local cost-to-go is infinite (or very large)*, thus extending the Hamilton-Jacobi equation into the obstacles. Assume that the speed $v = v(\mathbf{x})$ only varies in the spatial domain Ω_0 . We introduce the *slowness function* $r: \Omega_0 \rightarrow \mathbb{R}$ as the inverse of the speed: for a predefined large number $r_{max} \gg 1$, set

$$(42) \quad r(\mathbf{x}) = \begin{cases} 1/v(\mathbf{x}) & \text{if } \mathbf{x} \text{ is not in an obstacle,} \\ r_{max} & \text{if } \mathbf{x} \text{ is in an obstacle.} \end{cases}$$

The key idea is to regularize r by smoothing out the discontinuities over a narrow band of thickness $\epsilon = \mathcal{O}(h)$. Let us write the regularized slowness function as r^ϵ . We implemented the regularization

$$r^\epsilon(\mathbf{x}) = \begin{cases} r(\mathbf{x}) & \text{if } w(\mathbf{x}) > 0 \\ r_{max} & \text{if } w(\mathbf{x}) < -2\epsilon \\ \frac{r_{max}-r(\mathbf{x})}{2} \left[\cos\left(\frac{(w(\mathbf{x})+2\epsilon)\pi}{2\epsilon}\right) + 1 \right] + r(\mathbf{x}) & \text{if } w(\mathbf{x}) \in [-2\epsilon, 0], \end{cases}$$

where w is the signed distance function to the obstacles:

$$w(\mathbf{x}) = \inf_{r(\mathbf{x}_0)=r_{max}} |\mathbf{x} - \mathbf{x}_0| - \inf_{r(\mathbf{x}_0)<r_{max}} |\mathbf{x} - \mathbf{x}_0|.$$

We illustrate the above regularization r^ϵ in Figure 3 with $r_{max} = 100$, $\epsilon = 5h$ on a 80×80 discretization of Ω_0 . The obstacle is a circle of radius $1/2$ centered at the origin.

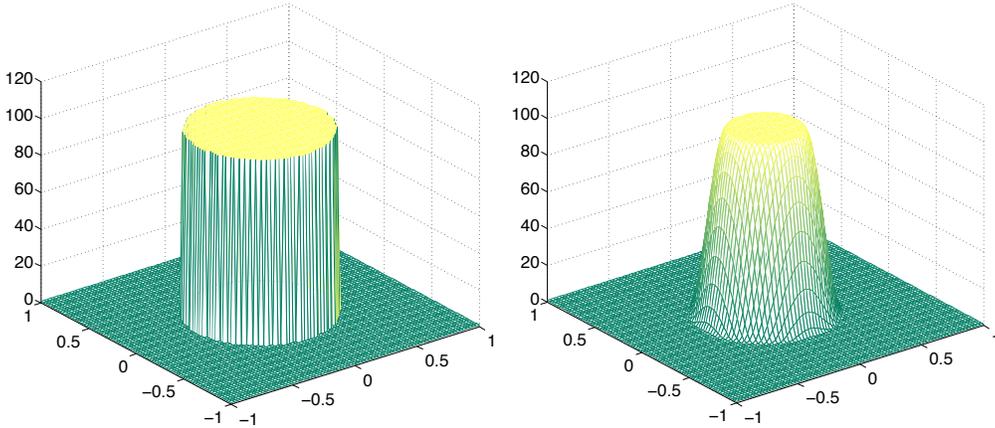


FIGURE 3. Left: the slowness function $r(\mathbf{x})$. Right: the regularized slowness function $r^\epsilon(\mathbf{x})$.

For the update scheme (37), at the grid node (x_i, y_j, θ_k) , we set Δt according to (38) and (42):

$$\Delta t = \rho h_\theta r^\epsilon(x_i, y_j, \theta_k).$$

Indeed, the numerical solution will be influenced by ϵ and r_{max} ; we have found that certain choices of these parameters noticeably improves the accuracy of the computed optimal trajectories, see section 4.5.5.

4.5. Numerical results. We give results of the numerical schemes described above. All test cases were computed on the domain $\Omega = [-1, 1]^2$.

4.5.1. *Dubins' car.* Recall that Dubins' car is the case $v_1 = 1$, $v_{-1} = 0$ and $\rho_1 > 0$. For all test cases, we used $\rho_1 = 0.2$. We computed the value function in the domain $\Omega = [-1, 1]^2$ using the semi-Lagrangian scheme (see section 4.2). Two targets were considered: a final pose target $\mathcal{T}_{\text{Pose}} = \{(0, 0, 0)\}$ and a final location target $\mathcal{T}_{\text{Loc}} = \{(0, 0, \theta) \mid \theta \in [0, 2\pi)\}$. Let

$$(43) \quad U(x, y) := \min_{\theta \in [0, 2\pi)} u(x, y, \theta),$$

the minimum arrival time to the target from (x, y) over all initial directions. Level sets of $u(x, y, \theta)$ as well as contour plots of $U(x, y)$ for the target $\mathcal{T}_{\text{Pose}}$ is shown in Figure 4 and for \mathcal{T}_{Loc} in Figure 5.

As a convergence test, we compared the numerical solution of $u(x, y, \theta)$ for the case $\mathcal{T}_{\text{Pose}} = \{(0, 0, 0)\}$ along the line segment $\{(x, 0, 0) \mid x \in [-1, 1]\}$. The exact solution is

$$u(x, 0, 0) = \begin{cases} -x & \text{for } x \leq 0 \\ 2\pi + x & \text{for } x > 0. \end{cases}$$

Figure 6 shows the a semi-log plot, demonstrating a $\mathcal{O}(h^{1/2})$ convergence rate in the L^2 norm for the semi-Lagrangian scheme. Note that $\mathcal{O}(h^{1/2})$ is a standard rate of convergence for numerical schemes of Hamilton-Jacobi equations [CL84]. As a comparison, we also plotted the same test results for the finite difference scheme [TTSL10], which gave a significantly slower convergence rate.

Remark 18. A possible cause of the significantly low convergence rate for the finite difference scheme is the presence of contact discontinuities [LeV90], i.e. where discontinuities are parallel to the characteristic direction.

We also performed a convergence test for the case $\mathcal{T}_{\text{Loc}} = \{(0, 0, \theta) \mid \theta \in [0, 2\pi)\}$ on the same line segment $\{(x, 0, 0) \mid x \in [-1, 1]\}$. The exact solution is

$$u(x, 0, 0) = \begin{cases} -x & \text{for } x \leq 0 \\ 2\rho(\pi - \tan^{-1}(x/\rho)) & \text{for } x > 0. \end{cases}$$

Figure 7 shows the L^2 norm convergence of both the semi-Lagrangian and finite difference schemes for the case \mathcal{T}_{Loc} . Note how the semi-Lagrangian scheme is more accurate than the finite difference scheme by a small margin.

Figure 8 shows a sample numerical path for a domain with obstacles.

4.5.2. *Reeds-Shepp's car.* Recall that the Reeds-Shepp's car is the case $v_1 = 1$, $v_{-1} = 1$ and $\rho_1 = \rho_{-1} =: \rho > 0$. Since the dynamics is STLC and the value function is continuous, we employ the finite difference approximation as described in section 4.1. Also, note that the value function is computed only on $[0, \pi)$ in the θ domain, with periodic boundary conditions (see discussion after Example 4).

Figures 9 and 10 show the level sets of $u(x, y, \theta)$ and contour plots of $U(x, y)$ (as defined in (43)), for the target pose $\mathcal{T}_{\text{Pose}} = \{(0, 0, 0)\}$ and target location $\mathcal{T}_{\text{Loc}} = \{(0, 0, \theta) \mid \theta \in [0, 2\pi)\}$, respectively. The turning radius is $\rho = 0.2$.

Sample optimal paths for the Reeds-Shepp's car are shown in Figure 11.

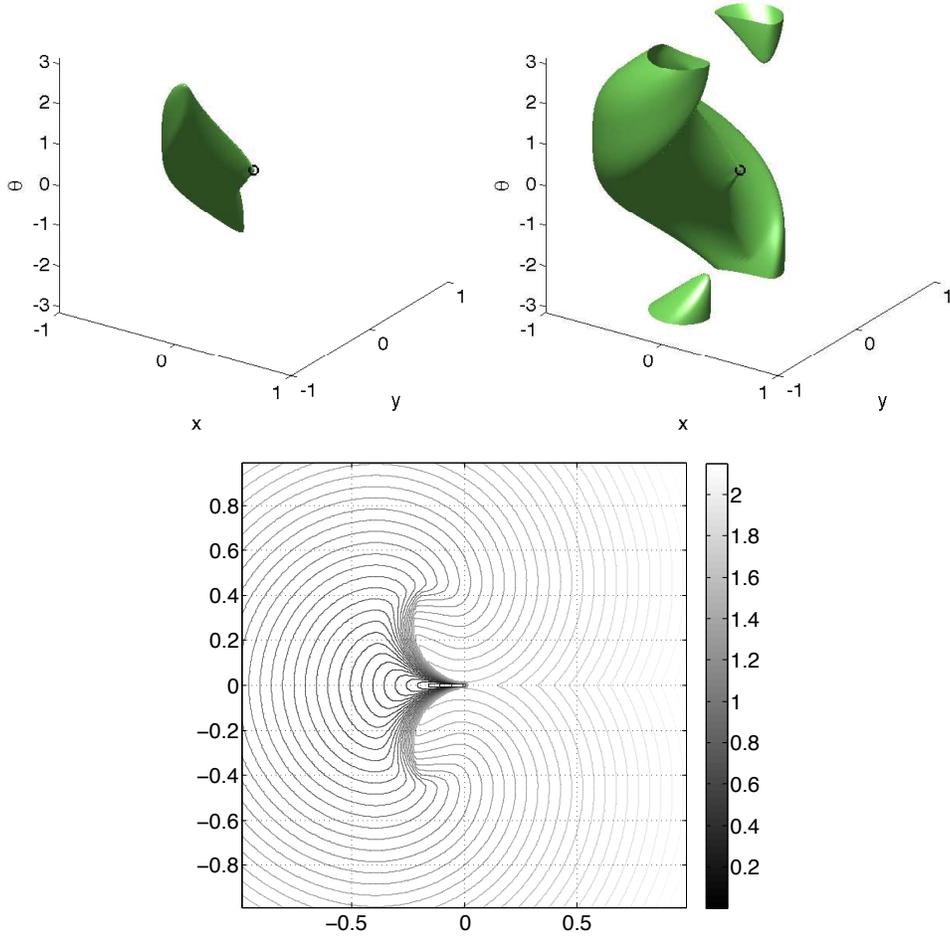


FIGURE 4. Numerical solution to Dubins' car with a target pose. Top: 0.7 and 1.1 level sets of u for $\mathcal{T}_{\text{Pose}} = \{(0, 0, 0)\}$ and turning radius $\rho_1 = 0.2$. The boundary point (target) is shown by a small black circle. Bottom: contour plot of $U(x, y)$.

4.5.3. *Different forward and reverse speed.* To demonstrate the versatility of our algorithm, we consider the steering of a vehicle with different forward and reverse speeds, see Figure 12. Treating different forward and reverse turning radii works analogously. Since the forward-reverse symmetry (as in the Reeds-Shepp's car) no longer holds, the θ domain is set to $[0, 2\pi)$.

4.5.4. *Spatially varying turning radius.* We tested our algorithm on a problem of steering a Dubins' car along a sequence of contiguous rectangles, each with difference turning radii, see Figure 13. This is a problem posed in [CT10], where it was solved using purely geometrical arguments. While explicit geometrical calculations become increasingly difficult

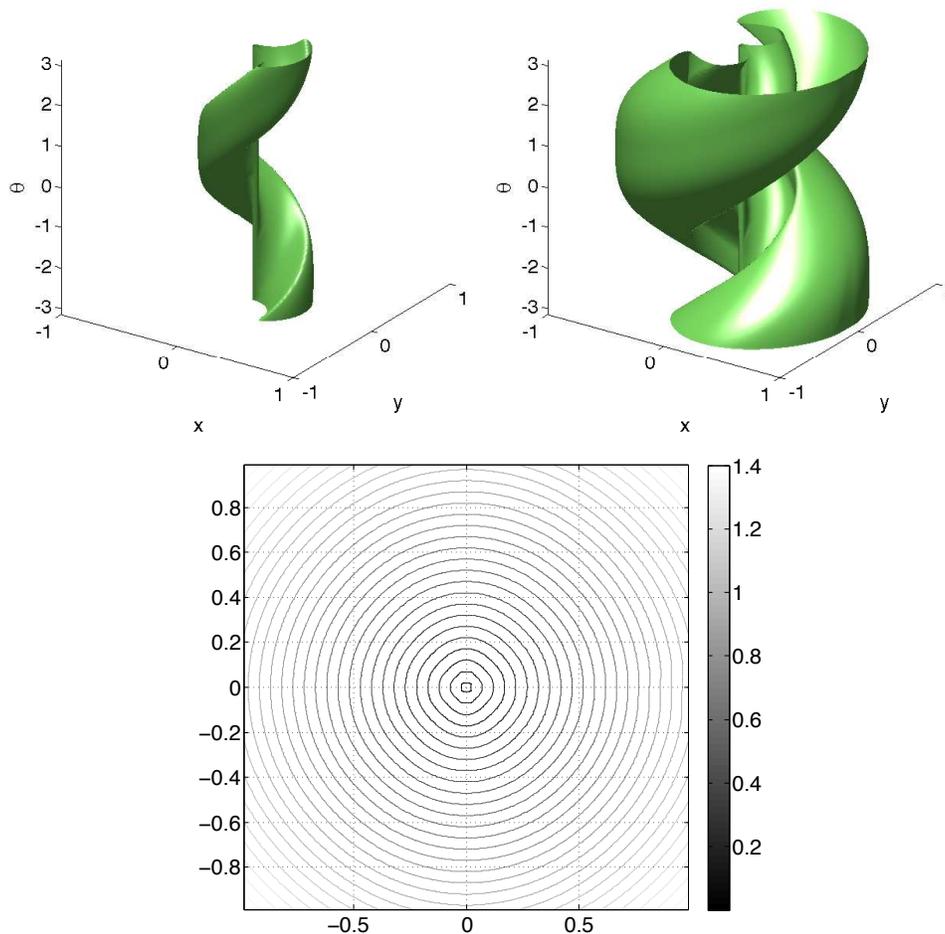


FIGURE 5. Numerical solution to Dubins' car with a target location. Top: 0.4 and 0.9 level sets of u for $\mathcal{T}_{\text{Loc}} = \{(0, 0, \theta) \mid \theta \in [0, 2\pi)\}$ and turning radius $\rho_1 = 0.2$. Bottom: contour plot of $U(x, y)$.

in complicated environments, our algorithm requires only a trivial modification: replace ρ in (40) by $\rho_{i,j,k} = \rho(x_i, y_j, \theta_k)$.

4.5.5. *Obstacle implementation and limitations of the algorithm.* Our Hamilton-Jacobi equations assume the bang-bang principle, which is valid only away from the obstacles. In the absence of obstacles, the works of Dubins and Reeds-Sheep prove that the steering must obey the bang-bang principle. However, for steering in domains amongst obstacles, the bang-bang principle may not yield exact solutions. A simple example is when the optimal path traces along a obstacle boundary of curvature less than ρ , the turning radius. Our algorithm forces the bang-bang principle in the grid directions of the underlying stencil. While a wider stencil may partially alleviate this issue (see wide stencil

N	L^2 error SL	rate
100	0.7582	-
200	0.5542	0.4523
300	0.4508	0.5093
400	0.3665	0.7193

N	L^2 error FD	rate
100	0.6985	-
200	0.6450	0.1150
300	0.6087	0.1429
400	0.5832	0.1485

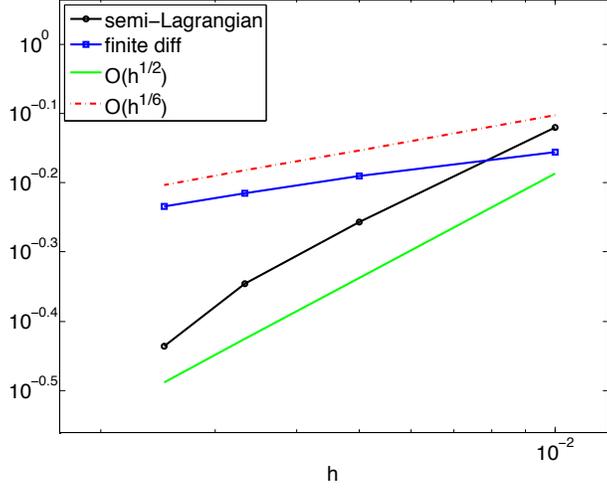


FIGURE 6. Convergence rates for the case $\mathcal{T}_{\text{Pose}} = \{(0,0,0)\}$ for the semi-Lagrangian (SL) scheme and the finite difference (FD) scheme. N is the number of grid points in each of the (x, y) directions, in the domain $\Omega = [-1, 1]^2$. In all cases, $h_\theta = 2\pi/300 \approx 0.0209$.

N	L^2 error SL	rate
100	0.3759	-
200	0.2754	0.4488
300	0.2270	0.4764
400	0.1969	0.4948

N	L^2 error FD	rate
100	0.3754	-
200	0.2779	0.4337
300	0.2343	0.4209
400	0.2082	0.4105

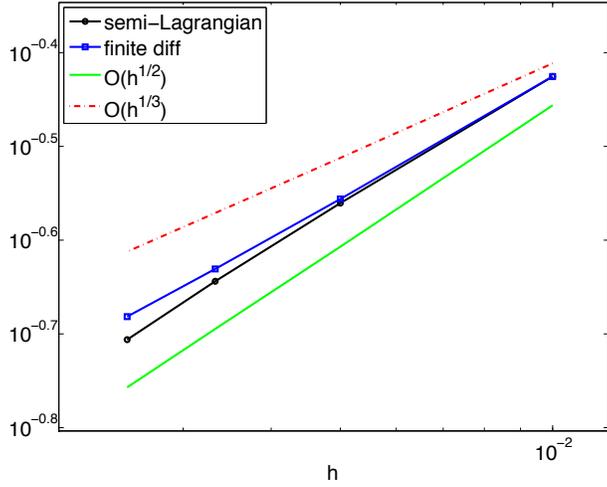


FIGURE 7. Convergence rates for the case $\mathcal{T}_{\text{Loc}} = \{(0,0,\theta) \mid \theta \in [0, 2\pi)\}$ for the semi-Lagrangian (SL) scheme and the finite difference (FD) scheme. N is the number of grid points in each of the (x, y) directions, in the domain $\Omega = [-1, 1]^2$. In all cases, $h_\theta = 2\pi/300 \approx 0.0209$.

implementation in [TTSL10]), a convergence in the limit $h_x, h_y, h_\theta \rightarrow 0$ requires a full implementation of the minimization over the control $a \in [-1, 1]$. While this is possible in principle, it will significantly complicate and slow down the current implementation.

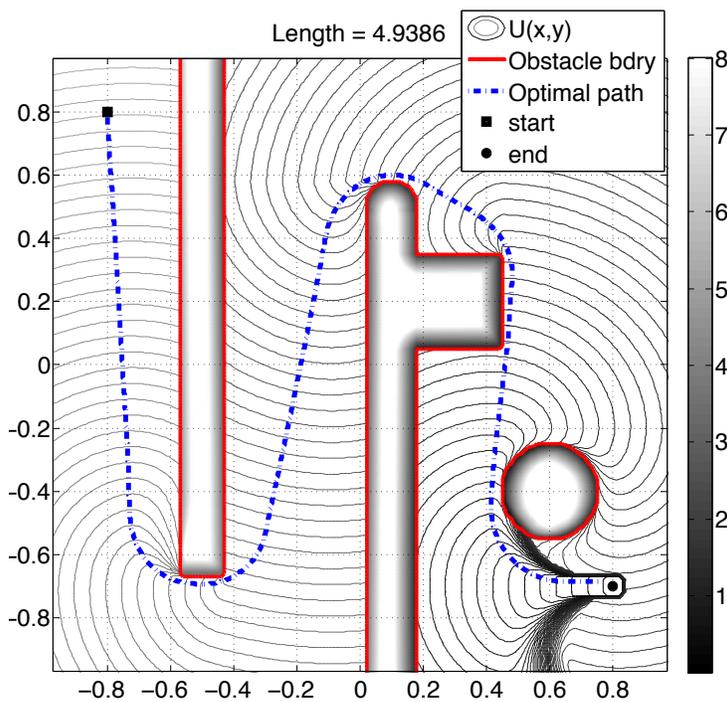


FIGURE 8. Numerical solution to Dubins' car in a domain among obstacles. The contour plot represents $U(x, y)$. Parameters are: initial pose $(x_0, y_0, \theta_0) = (-0.8, 0.8, -\pi/2)$, target pose $\mathcal{T} = \{(0.8, -0.7, 0)\}$, turning radius $\rho = 0.2$. The solution was approximated on a grid size $200 \times 200 \times 200$.

However, we have found that employing the regularized slowness function r_{ijk}^ϵ (see section 4.4) can also partially alleviate this issue. In Figure 14, we tested a case where the optimal path of a Dubins' car traces around a large circular obstacle with radius larger than the turning radius ρ . The maximum slowness was set to $r_{max} = 100$. Four different regularizations were tested, $\epsilon = 0$ (no regularization), $3h, 10h, 40h$. As the regularization parameter ϵ increases, note how the computed path invoking the bang-bang principle becomes closer to the true optimal path. However, if the regularization is too strong, the computed path can enter the obstacle (the case $\epsilon = 40h$).

5. CONCLUSION

We presented a Hamilton-Jacobi formulation for computing optimal trajectories with lower bounds on curvature. It models the optimal path planning problem of simple cars, such as the Dubins' and Reeds-Shepp's cars. In addition to formally deriving the Hamilton-Jacobi equations for such problems, we give a proof that the value functions solve the equations in the viscosity sense. Furthermore, approximation schemes are presented, as well as numerical results verifying the theory.

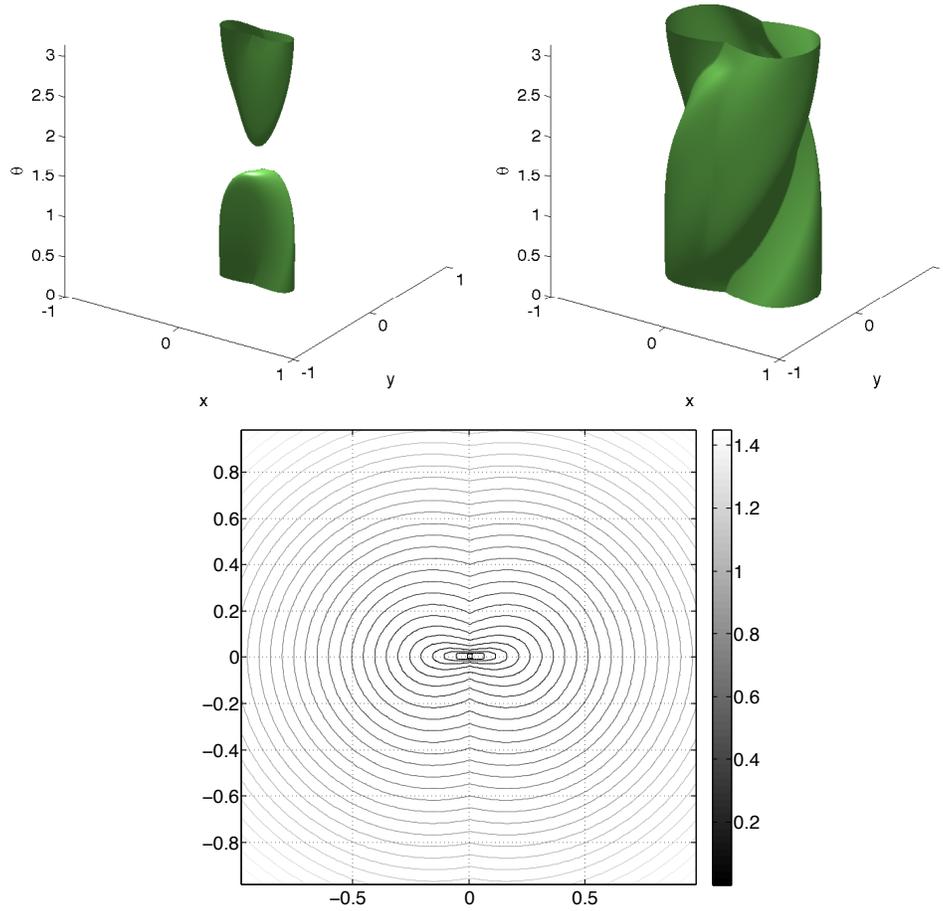


FIGURE 9. Numerical solution to Reeds-Shepp's car with a target pose. Top: 0.3 and 0.6 level sets of u for $\mathcal{T}_{\text{Pose}} = \{(0, 0, 0)\}$ and turning radius $\rho_1 = \rho_2 = 0.2$. Bottom: contour plot of $U(x, y)$.

6. ACKNOWLEDGEMENT

The authors thank Professors S. Osher, P. Souganidis and A. Vladimirsky for helpful discussions, and Dr. Y. Landa and Mr. H. Shen for taking part in the early stages of this project. Takei was supported by ONR grants N00014-07-1-0810 and N00014-08-1-1119. Tsai was supported by a Sloan Fellowship, National Science Foundation grants DMS-0914465, DMS-0914840, and a MURI sub-contract from Univ. of S. Carolina grant No. W911NF-07-1-0185.

REFERENCES

- [BCD97] Martino Bardi and Italo Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Systems & Control: Foundations & Applications.

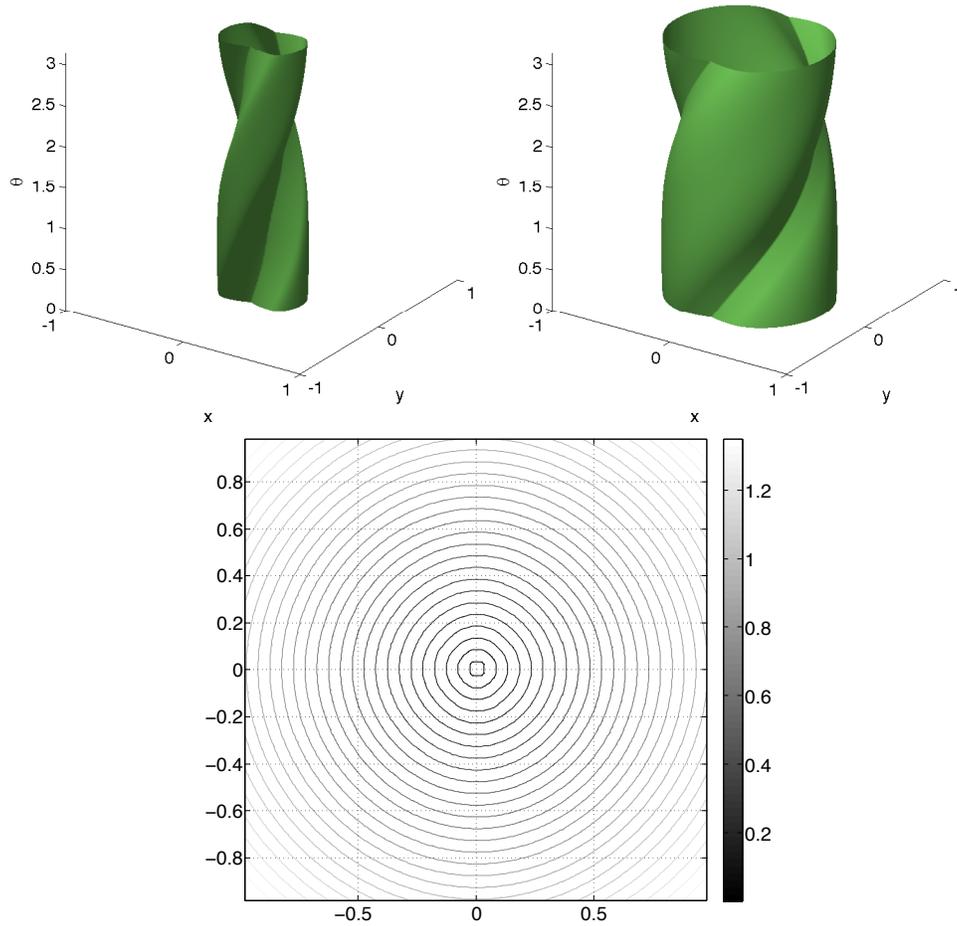


FIGURE 10. Numerical solution to Reeds-Shepp’s car with a target location. Top: 0.3 and 0.6 level sets of u for $\mathcal{T}_{\text{Loc}} = \{(0, 0, \theta) \mid \theta \in [0, 2\pi)\}$ and turning radius $\rho_1 = \rho_2 = 0.2$. Bottom: contour plot of $U(x, y)$.

Birkhäuser Boston Inc., Boston, MA, 1997. With appendices by Maurizio Falcone and Pierpaolo Soravia.

[Bel10] Richard Bellman. *Dynamic programming*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 2010. Reprint of the 1957 edition, With a new introduction by Stuart Dreyfus.

[BFS94] M. Bardi, M. Falcone, and P. Soravia. Fully discrete schemes for the value function of pursuit-evasion games. In *Advances in dynamic games and applications (Geneva, 1992)*, volume 1 of *Ann. Internat. Soc. Dynam. Games*, pages 89–105. Birkhäuser Boston, Boston, MA, 1994.

[BFS99] Martino Bardi, Maurizio Falcone, and Pierpaolo Soravia. Numerical methods for pursuit-evasion games via viscosity solutions. In *Stochastic and differential games*, volume 4 of *Ann. Internat. Soc. Dynam. Games*, pages 105–175. Birkhäuser Boston, Boston, MA, 1999.

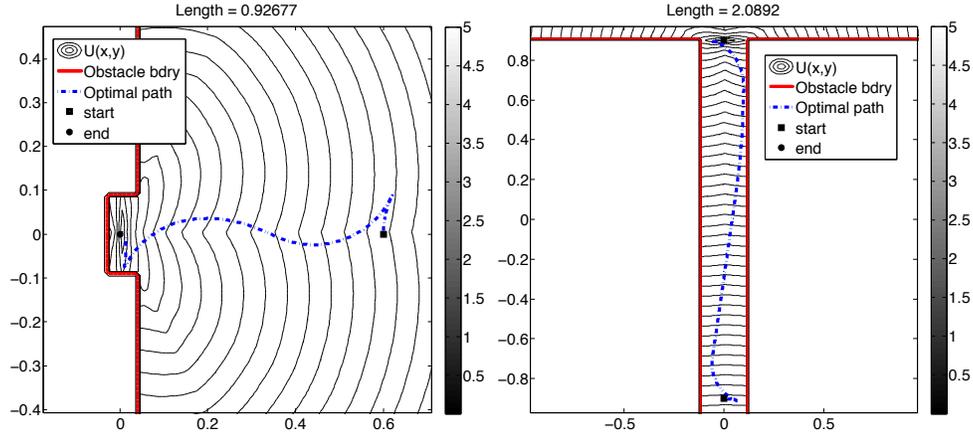


FIGURE 11. Optimal paths for the Reeds-Shepp's car, with $\rho_1 = \rho_2 = 0.2$. Left: initial pose $(x_0, y_0, \theta_0) = (0.6, 0, \pi/2)$ and target pose $\mathcal{T} = \{(0, 0, \pi/2)\}$. Right: initial pose $(x_0, y_0, \theta_0) = (0, -0.9, 0)$ and target pose $\mathcal{T} = \{(0, 0.9, 0)\}$.

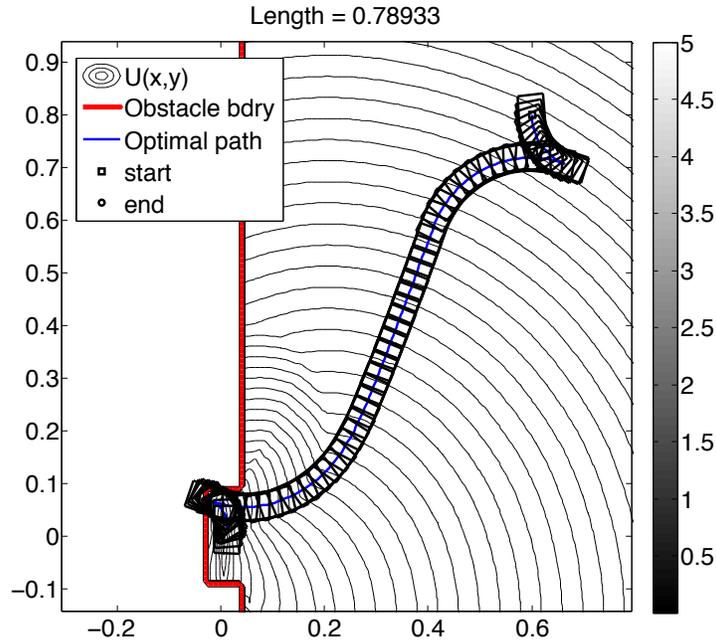


FIGURE 12. A vehicle optimal path with $\rho_1 = 1/6$, $\rho_{-1} = 1/10$, $v_1 = 2$, $v_{-1} = 1$. The initial pose is $(x_0, y_0, \theta_0) = (0.6, 0.8, \pi/2)$ and the target pose is $\mathcal{T} = \{(0, 0, \pi/2)\}$. The path is overlaid with a rectangular box over fixed time intervals to depict the difference in forward and reverse speeds.

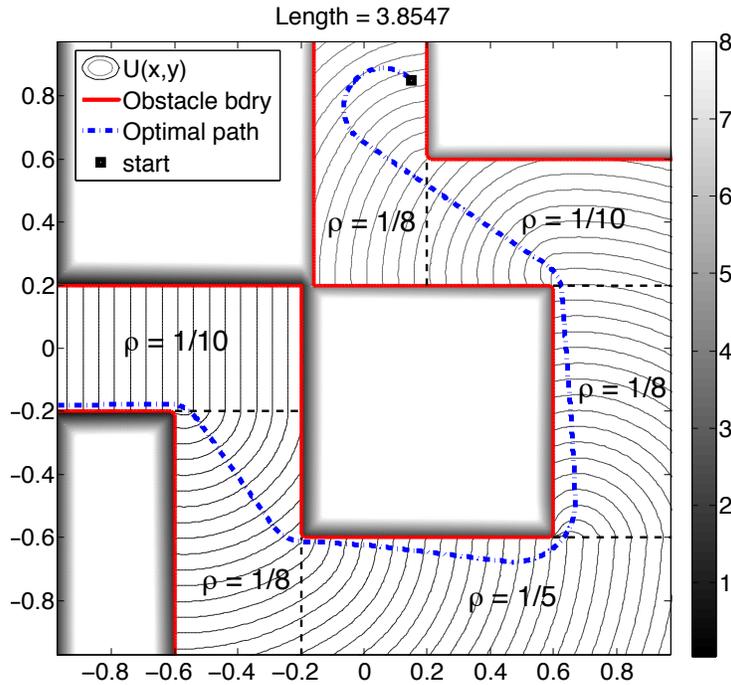


FIGURE 13. Dubins' car problem for a spatially varying turning radius. The thin dotted lines are where $\rho(x, y)$ changes value. The initial position of the optimal path is $(x_0, y_0, \theta_0) = (0.15, 0.85, 3\pi/4)$. The target is $T = \{(-1, y, \theta) \mid y \in [-0.2, 0.2], \theta \in [0, 2\pi)\}$.

[BK07] Jonathan Backer and David Kirkpatrick. Finding curvature-constrained paths that avoid polygonal obstacles. In *SCG '07: Proceedings of the twenty-third annual symposium on Computational geometry*, pages 66–73, New York, NY, USA, 2007. ACM.

[BL93] Jerome Barraquand and Jean-Claude Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10(2):121–155, October 1993.

[BS91] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, 4(3):271–283, 1991.

[CL83] Michael G. Crandall and Pierre-Louis Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277(1):1–42, 1983.

[CL84] Michael G. Crandall and Pierre-Louis Lions. Two approximations of solutions of hamilton-jacobi equations. *Mathematics of Computation*, 43(167):1–19, July 1984.

[CL07] H. Chitsaz and S. M. LaValle. Time-optimal paths for a Dubins airplane. In *Proceedings IEEE Conference Decision and Control*, 2007. To appear.

[CS02] Gui-Qiang Chen and Bo Su. On global discontinuous solutions of hamilton-jacobi equations. *Comptes Rendus Mathematique*, 334(2):113 – 118, 2002.

[CS07] Yingda Cheng and Chi-Wang Shu. A discontinuous galerkin finite element method for directly solving the hamilton-jacobi equations. *Journal of Computational Physics*, 223(1):398 – 415, 2007.

[CT10] R. V. Cowlagi and P. Tsiotras. Existence and synthesis of curvature-bounded paths inside nonuniform rectangular channels. In *Proceedings of the 2010 American Control Conference, Baltimore, MD, USA*, 2010.

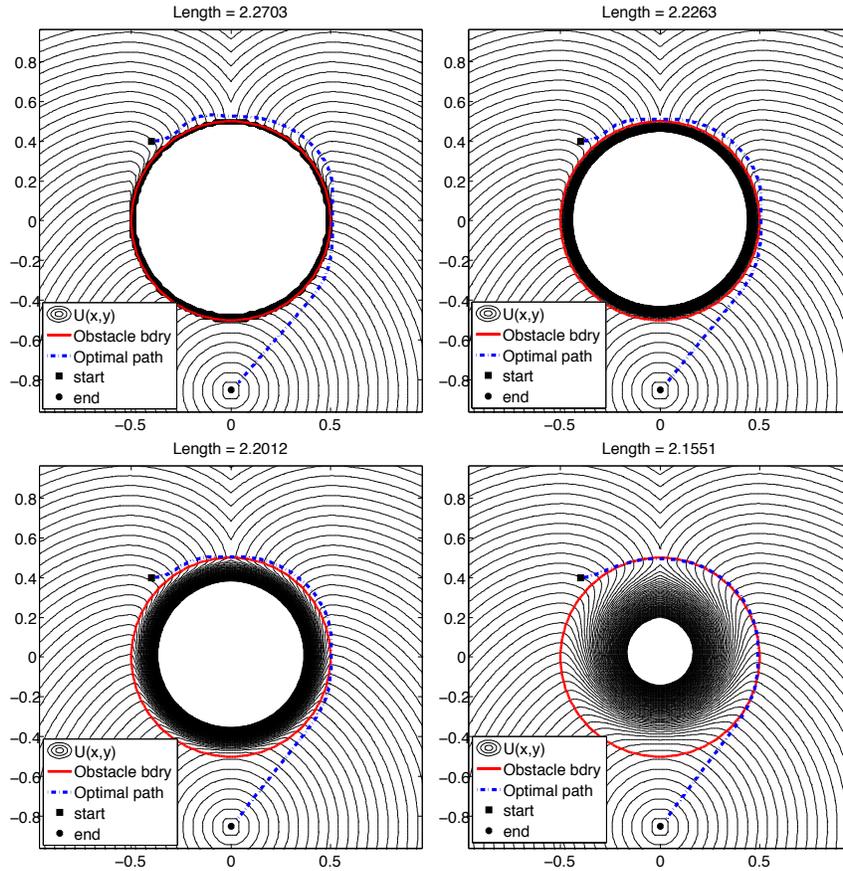


FIGURE 14. Implementation of the obstacle boundaries and the limitations of implementing the bang-bang control. Thickness of regularization of obstacles in pixels, from top left, clockwise: $\epsilon = 0, 3h, 40h, 10h$. Since the turning radius ($\rho = 0.2$) is smaller than the radius of the circular obstacle (radius 0.4), the computed path is suboptimal. By regularizing the slowness function $r(x) = 1/v(x)$ some improvement is made. However, too much regularization causes paths to enter the obstacle.

- [Dub57] 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(3):497–516, 1957.
- [Fal07] M. Falcone. Fast marching methods for front propagation. Lecture notes at “Introduction to Numerical Methods for Moving Boundaries”, November 2007.
- [FF02] M. Falcone and R. Ferretti. Semi-lagrangian schemes for hamilton-jacobi equations, discrete representation formulae and godunov methods. *J. Comput. Phys.*, 175(2):559–575, 2002.
- [GS01] Yoshikazu Giga and Moto-Hiko Sato. A level set approach to semicontinuous viscosity solutions for Cauchy problems. *Comm. Partial Differential Equations*, 26(5-6):813–839, 2001.
- [HL69] H. Hermes and J.P. LaSalle. *Functional analysis and time optimal control*. Academic Press, New York, 1969.

- [HS] Changqing Hu and Chi-Wang Shu. A discontinuous galerkin finite element method for hamilton-jacobi equations. *SIAM J. Sci. Comput.*, 21:666–690.
- [Ish87] Hitoshi Ishii. Perron’s method for Hamilton-Jacobi equations. *Duke Math. J.*, 55(2):369–384, 1987.
- [JC92] P. Jacobs and J. Canny. *Planning smooth paths for mobile robots*, pages 271–342. Kluwer Academic, Norwell, MA, 1992.
- [KL99] Prashanth Konkimalla and Steven M. Lavalle. Efficient computation of optimal navigation functions for nonholonomic planning. In *In Proc. First IEEE Int’l Workshop on Robot Motion and Control*, pages 187–192, 1999.
- [KOQ04] Chiu Yen Kao, Stanley Osher, and Jianliang Qian. Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations. *J. Comput. Phys.*, 196(1):367–391, 2004.
- [KOT04] Chiu-Yen Kao, Stanley Osher, and Yen-Hsi Tsai. Fast sweeping methods for static hamilton-jacobi equations. *SIAM J. Numer. Anal.*, 42(6):2612–2632, 2004.
- [KV10] Ajeet Kumar and Alexander Vladimirovsky. An efficient method for multiobjective optimal control and optimal control subject to integral constraints. *Journal of Computational Mathematics*, 28(4):517–551, 2010.
- [LaV06] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Also available at <http://planning.cs.uiuc.edu/>.
- [LeV90] Randall J. LeVeque. *Numerical Methods for Conservation Law*. Lectures in Mathematics, ETH-Zurich. Birkhauser-Verlag, 1990.
- [LJTM94] J.-P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray. A motion planner for nonholonomic mobile robots. *Robotics and Automation, IEEE Transactions on*, 10(5):577–593, oct 1994.
- [Obe06] Adam M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton-Jacobi equations and free boundary problems. *SIAM J. Numer. Anal.*, 44(2):879–895 (electronic), 2006.
- [OS91] Stanley Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for hamilton-jacobi equations. *SIAM Journal on Numerical Analysis*, 28(4):907–922, August 1991.
- [PT09] V. S. Patsko and V. L. Turova. From dubins’ car to reeds and shepp’s mobile robot. *Computational Visul Science*, 12:345–364, 2009.
- [RS90] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.*, 145(2):367–393, 1990.
- [RT92] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29(3):867–884, 1992.
- [RW00] John H. Reif and Hongyan Wang. Nonuniform discretization for kinodynamic motion planning and its applications. *SIAM J. Comput.*, 30(1):161–190, 2000.
- [Set95] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci.*, pages 1591–1595, 1995.
- [Sor93] Pierpaolo Soravia. Discontinuous viscosity solutions to dirichlet problems for hamilton-jacobi equations with convex hamiltonian. *Comm. in Partial Differential Equations*, 18:1493–1514, 1993.
- [ST91] Héctor J. Sussmann and Guoqing Tang. Shortest paths for the reeds-shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Technical report, SYCON Report 91-10, 1991.
- [SVV65] L. Sonneborn and F. Van Vleck. The bang-bang principle for linear control systems. *SIAM J. Control*, 2:151–159, 1965.
- [TCOZ03] Yen-Hsi Richard Tsai, Li-Tien Cheng, Stanley Osher, and Hong-Kai Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 41(2):673–694 (electronic), 2003.
- [TGO03] Yen-Hsi Richard Tsai, Yoshikazu Giga, and Stanley Osher. A level set approach for computing discontinuous solutions of Hamilton-Jacobi equations. *Math. Comp.*, 72(241):159–181 (electronic), 2003.
- [Tsi95] John N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

- [TTSL10] Ryo Takei, Richard Tsai, Haochong Shen, and Yanina Landa. A practical path-planning algorithm for a vehicle with a constrained turning radius: a Hamilton-Jacobi approach. In *American Control Conference, 2010. ACC '10.*, 7 2010.
- [WA96] Hongyan Wang and Pankaj K. Agarwal. Approximation algorithms for curvature-constrained shortest paths. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 409–418, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.
- [Zha04] Hongkai Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627, May 2004.

DEPARTMENT OF MATHEMATICS, UCLA
E-mail address: `rrtakei@ucla.edu`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF TEXAS, AUSTIN
E-mail address: `ytsai@math.utexas.edu`