A Primal-Dual Method for Optimal Control and Trajectory Generation in High-Dimensional Systems*

Matthew Kirchner[1], Gary Hewer[1], Jérôme Darbon[2], and Stanley Osher[3]

Abstract— **Presented is a method for efficient computation of the Hamilton–Jacobi (HJ) equation for time-optimal control problems using the generalized Hopf formula. Typically, numerical methods to solve the HJ equation rely on a discrete grid of the solution space and exhibit exponential scaling with dimension. The generalized Hopf formula avoids the use of grids and numerical gradients by formulating an unconstrained convex optimization problem. The solution at each point is completely independent, and allows a massively parallel implementation if solutions at multiple points are desired. This work presents a primal-dual method for efficient numeric solution and presents how the resulting optimal trajectory can be generated directly from the solution of the Hopf formula, without further optimization. Examples presented have execution times on the order of milliseconds and experiments show computation scales approximately polynomial in dimension with very small high-order coefficients.**

I. Introduction

Hamilton–Jacobi equations play a fundamental role in optimal control theory as they establish sufficient conditions for optimality [26]. Traditionally, numerical solutions to HJ equations require a dense, discrete grid of the solution space [25], [23], [22]. Computing the elements of this grid scales poorly with dimension and has limited use for problems with dimension greater than four. The exponential dimensional scaling in optimization is sometimes referred to as the "curse of dimensionality" [4], [3]. Recent research [10], [18] has discovered numerical solutions based on the generalized Hopf formula that do not require a grid and can be used to efficiently compute solutions of a certain class of Hamilton–Jacobi PDEs that arise in linear control theory and differential games.

A key hurdle in the development of efficient high-dimensional solutions is the time-dependent Hamiltonian that results for general control problems. Kirchner et al. [18] applied the generalized Hopf formula with time-dependent Hamiltonian to efficiently solve multi-vehicle optimal pursuit-evasion using the linearized models found in [27]. In that work, the unique, specific structure of the model was used to derive a closed form solution to the gradient of the objection function, thereby allowing efficient optimization. This same technique cannot be applied to general linear systems and attempts to use numeric gradient approximation would increase computation time.

Darbon and Osher presented a proximal splitting algorithm in [10] using the split Bregman/ADMM approach [13], but this only applies to systems with time-independent Hamiltonians of the form $\dot{x} = f\left(u\left(t\right)\right)$, and has limited use for general linear control problems. Chow et al. [7] developed a coordinate descent method, but this optimization method lacks robustness for the nonsmooth optimization problems that typically result from the Hopf formula for optimal control problems.

This work presents a parallel proximal splitting optimization method [8] for solving time-optimal control problems with the generalized Hopf formula, including those with time-dependent Hamiltonians. This allows efficient solutions to generalized linear models, even when no explicit gradient of the objective function is known and without resorting to consensus-type algorithms [28]. The rest of the paper is structured as follows. Section II reviews using the Hopf formula for solutions to the Hamilton Jacobi equations that arise in optimal linear control and largely follows the work of [18] and [10]. The main contributions of this paper are presented in Section III, with a primal-dual method for solving the Hopf formula, Section IV, which presents obtaining the optimal control, and Section V, where the optimal trajectory can be obtained directly from the solution of the Hopf formula. The new methods are applied on various time-optimal control problems and are presented in Section VI.

II. Solutions to Hamilton–Jacobi Equations with the Hopf Formula

Consider system dynamics represented as

$$\dot{x}\left(t\right) = f\left(u\left(t\right)\right), \tag{1}$$

where $x\left(t\right) \in \mathbb{R}^n$ is the system state and $u\left(t\right) \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, constrained to lie in the convex admissible control set $\mathcal{U}$. The system in (1), describes how the state evolves in time and is considered a dynamic constraint when control inputs $u\left(t\right)$ are to be optimized. We consider a cost functional for a given initial time t, and terminal time T

$$K\left(x,u,t\right) = \int_t^T L\left(u\left(s\right)\right) ds + J\left(x\left(T\right)\right), \tag{2}$$

where $x\left(T\right)$ is the solution of (1) at terminal time, T. We assume that the terminal cost function $J : \mathbb{R}^n \to \mathbb{R}$ is convex.

*This research was supported by the Office of Naval Research, ILIRs 4764 and 5100, and Office of Naval Research grants N00014-16-12119 and N00014-16-12157.

[1] M. Kirchner and G. Hewer are with the Image and Signal Processing Branch, Research Directorate, Code 4F0000D, Naval Air Warfare Center Weapons Division, China Lake, CA 93555, USA {matthew.kirchner, gary.hewer}@navy.mil

[2] J. Darbon is with the Division of Applied Mathematics, Brown University, Providence, RI 02912, USA jerome_darbon@brown.edu

[3] S. Osher is with the Department of Mathematics, University of California, Los Angeles, CA 90095, USA sjo@math.ucla.edu

The function $L : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is the running cost, and represents the rate that cost is accrued. The value function $v : \mathbb{R}^n \times (-\infty, T] \to \mathbb{R}$ is defined as the minimum cost, K, among all admissible controls for a given state x, and time $t \leq T$ with

$$v(x,t) = \inf_{u \in \mathcal{U}} K(x,u,t). \tag{3}$$

The value function in (3) satisfies the dynamic programming principle [5], [11] and also satisfies the following initial value Hamilton–Jacobi (HJ) equation by defining the function $\varphi : \mathbb{R}^n \times \ \to \mathbb{R}$ as $\varphi(x,t) = v(x, T-t)$, with φ being the viscosity solution of

$$\begin{cases} \frac{\partial \varphi}{\partial t}(x,t) + H(\nabla_x \varphi(x,t)) = 0 & \text{in } \mathbb{R}^n \times (0,+\infty), \\ \varphi(x,0) = J(x) & \forall x \in \mathbb{R}^n, \end{cases} \tag{4}$$

where the Hamiltonian $H : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is defined by

$$H(p) = \sup_{c \in \mathbb{R}^m} \{\langle -f(c), p\rangle - L(c)\}. \tag{5}$$

We proceed with the Hamilton Jacobi formulation for time-optimal control to reach some convex terminal set Ω, though the following methods can be generalized to other optimization problems. To apply the constraint that the control must be bounded, we introduce the following running cost $L = \mathcal{I}_{\mathcal{U}}$, where $\mathcal{I}_{\mathcal{C}} : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is the indicator function for the set $\mathcal{C}$ and is defined by

$$\mathcal{I}_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ +\infty & \text{otherwise.} \end{cases}$$

This reduces the Hamiltonian to

$$H(p) = \max_{c \in \mathcal{U}} \langle -f(c), p\rangle.$$

Solving the HJ equation (4) describes how the value function evolves with time at any point in the state space and from this, optimal control policies can be found.

A. *Viscosity Solutions with the Hopf Formula*

It was shown in [11] that an exact, point-wise viscosity solution to (4) can be found using the Hopf formula [15]. Moreover, since no discrete grid is constructed, the formula can provide a numerical method that is efficient even when the state space is high-dimensional. The value function can be found with the Hopf formula

$$\varphi(x,t) = -\min_{p \in \mathbb{R}^n} \{J^\star(p) + tH(p) - \langle x, p\rangle\}, \tag{6}$$

where the Fenchel–Legendre transform $g^\star : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ of a convex, proper, lower semicontinuous function $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is defined by [14]

$$g^\star(p) = \sup_{x \in \mathbb{R}^n} \{\langle p, x\rangle - g(x)\}. \tag{7}$$

B. *General Linear Models*

Now consider the following linear state space model

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{8}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, state vector $x \in \mathbb{R}^n$, and control input $u \in \mathcal{U} \subset \mathbb{R}^m$. We can make a change of variables

$$z(t) = e^{-tA}x(t), \tag{9}$$

which results in the following system

$$\dot{z}(t) = e^{-tA}Bu(t), \tag{10}$$

with terminal cost function now defined in z with

$$\varphi(z,0) = J_z(z,0) = J_x\left(e^{TA}z\right). \tag{11}$$

For clarity in the sections to follow, we use the notation H_z to refer to the Hamiltonian for systems defined by (10), and H_x for systems defined by (8). Additionally, with a slight abuse of notation, we denote by $J_x^\star(p,T)$ the Fenchel transform of $J(x,t)$ with respect to the variable x at time $t = T$. Notice that the system (10) is now time-varying, and it was shown in [19, Section 5.3.2, p. 215] that the Hopf formula can be generalized for a time-dependent Hamiltonian to solve for the value function of the system in (10) with

$$\varphi(z_0, T) = -\min_{p \in \mathbb{R}^n} \Bigg\{ J_z^\star(p,0) \tag{12}$$

$$+ \int_0^T H_z(p,s)\,ds - \langle z_0, p\rangle \Bigg\},$$

with H_z defined as

$$H_z(p,t) = \max_{c \in \mathcal{U}} \left\langle -e^{-(T-t)A}Bc, p \right\rangle. \tag{13}$$

The change of variable to $(T-t)$ is required for time since the problem was converted to an initial value formulation from a terminal value formulation in (4). The value function found by solving the unconstrained optimization problem in (12) can be thought of as the minimal cost of a system starting at initial state $z(0) = z_0$ and ending at terminal state $z(T)$.

III. Proximal Splitting Methods for Control

Proximal splitting methods [8] are a powerful group of convex optimization algorithms that efficiently solve non-smooth minimization problems in high dimensions. These methods are used for problems of the form

$$\min_{p \in \mathbb{R}^n} G(p) + F(p),$$

where proximal points for G and F can be easily computed. These family of methods have been proven effective for image processing and compressed sensing applications [13]. The proximal point of f at p for some $\alpha > 0$ is given by

$$(I + \alpha\partial f)^{-1}(p) = \arg\min_{w \in \mathbb{R}^n} \left\{ \alpha f(w) + \frac{1}{2}\|w - p\|_2^2 \right\}.$$

The primal-dual algorithm [6] is a proximal splitting algorithm that solves the minimization problem of the form

$$\min_{p\in\mathbb{R}^n} G(p) + F(Kp), \tag{14}$$

with $K \in \mathbb{R}^{m\times n}$ and G, F being assumed convex, by converting (14) to the saddle-point problem

$$\min_{p\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} \langle Kp, y\rangle + G(p) - F^\star(y).$$

The minimizer can be found by iterating the following update procedure

$$\begin{cases} y^{k+1} &= (I+\sigma\partial F^\star)^{-1}\left(y^k + \sigma K\bar{p}^k\right) \\ p^{k+1} &= (I+\tau\partial G)^{-1}\left(p^k - \tau K^\top y^{k+1}\right) \\ \bar{p}^{k+1} &= p^{k+1} + \theta\left(p^{k+1} - p^k\right), \end{cases} \tag{15}$$

until convergence with $\tau, \sigma > 0$ being the primal and dual step sizes and $\theta \in [0,1]$. It was shown in [6] that the primal-dual method of (15) converges at a rate of $\mathcal{O}(1/k)$ for general convex functions F and G provided the condition $\tau\sigma\|K\|^2 < 1$ is satisfied. If one or both of F and G are strongly convex, then [6] provides an alteration to the algorithm in (15) that was shown to converge at a faster rate. While some problems in the Section VI meet this criteria, these accelerated algorithms are outside the scope of this work and all examples used the algorithm in (15).

A. Primal-Dual Solutions to the Generalized Hopf Formula

Suppose $\mathcal{U}$ is a closed convex set such that $0 \in \operatorname{int}\mathcal{U}$. Then $(\mathcal{I}_\mathcal{U})^\star$ defines a norm $\|(\cdot)\|$ and we denote by $\|(\cdot)\|_*$ its dual norm [14]. Also consider that the set $\mathcal{U}$ can be scaled by an injective linear transformation, Q^{-1}, to give the appropriate, problem-specific control bound, then (13) can be written as

$$H_z(p,t) = \left\|\left(-e^{-(T-t)A}BQ\right)^\top p\right\|_*. \tag{16}$$

For simplicity, we follow [18] by approximating the integral in (12) with a left Riemann sum quadrature with N equally spaced terms defined by

$$t_i = i\Delta t, \tag{17}$$

with $i \in \{0,\dots,N-1\}$ and $\Delta t = \frac{T}{N}$. The generalized Hopf formula, for some terminal time T, becomes

$$\begin{aligned}\varphi(z_0,T) = -\min_{p\in\mathbb{R}^n}\Bigg\{ & J_z^\star(p,0) \\ & + \Delta t\sum_{i=0}^{N-1}\left\|\left(-e^{-(T-t_i)A}BQ\right)^\top p\right\|_* \\ & - \langle z_0, p\rangle\Bigg\}. \end{aligned}\tag{18}$$

Other forms of quadrature can be used for integral approximations, such as trapezoidal or Simpson's rule [21], that increase approximation accuracy without sacrificing computational performance. Additionally, other advanced approximations could be considered, such as those employed by pseudospectral methods [30], [31], and will be investigated in future work.

To formulate as a primal-dual optimization, first set

$$G(p) = J_z^\star(p,0) - \langle z_0, p\rangle. \tag{19}$$

Now define

$$K_i = \left(-e^{-(T-t_i)A}BQ\right)^\top \tag{20}$$

and $F_i = \Delta t\|(\cdot)\|_* = (\mathcal{I}_{\Delta t\mathcal{U}})^\star$, where we denote by $\mathcal{I}_{\alpha\mathcal{C}}$ the indicator function of a set $\mathcal{C}$ scaled by a constant $\alpha > 0$ defined as

$$\mathcal{I}_{\alpha\mathcal{C}}(x) = \begin{cases} 0 & \text{if } \frac{x}{\alpha} \in \mathcal{C} \\ +\infty & \text{otherwise.}\end{cases}$$

The sum term in (18) can be written as

$$\Delta t\sum_{i=0}^{N-1}\left\|\left(-e^{-(T-t_i)A}BQ\right)^\top p\right\|_* = \sum_{i=0}^{N-1} F_i(K_i p).$$

We can now form the matrix K as

$$K = \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_N \end{bmatrix},$$

which gives

$$F(Kp) = \sum_{i=0}^{N-1} F_i(K_i p), \tag{21}$$

and combined with (19) is now in the form of (14). Note that K in this formulation is non square (preconditioning [29] can be used to enhance convergence rate). If the action of the matrix exponential in (20) is not known, then it can be quickly evaluated, for all time samples, without resorting to computing the matrix exponential with [1]. The structure of (21) forms a *separable sum*. This implies that for some $y = (y_1, y_2, \dots, y_N)$

$$F^\star(y_1, y_2, \dots, y_N) = F_1^\star(y_1) + F_2^\star(y_2) + \cdots + F_N^\star(y_N).$$

Recall that $F_i = (\mathcal{I}_{\Delta t\mathcal{U}})^\star$, thus

$$F_i^\star(y_i) = \mathcal{I}_{\Delta t\mathcal{U}}. \tag{22}$$

The proximal operator of a separable sum is simply

$$(I+\partial F^\star)^{-1}(y_1, y_2, \dots, y_N) = \begin{bmatrix} (I+\partial F_1^\star)^{-1}(y_1) \\ (I+\partial F_2^\star)^{-1}(y_2) \\ \vdots \\ (I+\partial F_N^\star)^{-1}(y_N) \end{bmatrix}. \tag{23}$$

The proximal operators in (23) are independent of each other, and as a result can be computed in parallel. This can be advantageous for real-time implementations in hardware such as multi-core embedded CPUs and field programmable gate arrays (FPGAs).

B. Stopping Criteria

Care must be taken to choose the stopping criterion of the algorithm in (15). The step sizes τ and σ are in general not equal and any stopping criteria must account for this. As a result we chose the primal and dual residuals [12, See Eqs. 10 and 12] as a step-size dependent stopping criteria and stop iterating (15) when the conditions

$$\left\| \left(p^k - p^{k-1}\right) / \tau - K^\top \left(y^k - y^{k-1}\right) \right\| < \varepsilon,$$

and

$$\left\| \left(y^k - y^{k-1}\right) / \sigma - K \left(p^k - p^{k-1}\right) \right\| < \varepsilon,$$

are both met. ϵ is set to 10^{-4} for all the experiments listed in Section VI.

IV. Time-Optimal Control

To find the time-optimal control to some convex terminal set Ω, choose a convex terminal cost function J_x such that

$$\begin{cases} J_x(x) < 0 & \text{for any } x \in \operatorname{int}\Omega, \\ J_x(x) > 0 & \text{for any } x \in (\mathbb{R}^n \setminus \Omega), \\ J_x(x) = 0 & \text{for any } x \in (\Omega \setminus \operatorname{int}\Omega), \end{cases}$$

where $\operatorname{int}\Omega$ denotes the interior of Ω. The intuition behind defining the terminal cost function this way is simple. If the value function $\varphi(x_0, T) < 0$ for some x_0 and T, then there exists a control $u(t)$ that drives the state from the initial condition at x_0, to the final state, $x(T)$ inside the set Ω. The smallest value of time T, such that $\varphi(x_0, T) = 0$ is the minimum time to reach the set Ω, starting at state x_0. Recall that the initial value function, $J_z(p, 0)$, and its associated Fenchel–Legendre transform $J_z^\star(p, 0)$, of the Hopf formula in (12) is defined in z, and must be transformed with (11). The minimum time to reach Ω is denoted by T^*, and the control computed at T^* is the time-optimal control. As first noted in [20], Hopf formula is itself a Fenchel–Legendre transform. It follows from a well known property of the Fenchel–Legendre transform [9] that the unique minimizer of (12) is the gradient of the value function

$$\nabla_z \varphi(z_0, T) = \arg\min_{p \in \mathbb{R}^n} \Bigg\{ J_z^\star(p, 0) + \int_0^T H_z(p, s)\, ds - \langle z_0, p \rangle \Bigg\}, \tag{24}$$

provided the gradient exists. So when solving for the value function using (12), we automatically solve for the gradient. We will refer to the minimizer in (24) as $p^* = \nabla_z(z_0, T)$.

We propose solving for the minimum time to reach, T^*, by a hybrid method of the bisection method and Newton's method. Newton's method has been shown to have faster convergence (quadratic) than bisection, but is unstable when the gradient is small and motivates the use of a hybrid method. We can iterate time, t_{n+1}, with Newton's as

$$t_{n+1} = t_n - \frac{\varphi(x_0, t_n)}{\frac{\partial \varphi}{\partial t}(x_0, t_n)}. \tag{25}$$

As noted in [10], $\frac{\partial \varphi}{\partial t}(x_0, t)$ must satisfy the Hamilton–Jacobi equation (4). Therefore we have

$$\frac{\partial \varphi}{\partial t}(x_0, t_n) = -H_x\left(\nabla_x \varphi(x_0, t_n), x_0\right).$$

We also see from (9) and applying the chain rule that

$$\begin{aligned} \nabla_x \varphi(z(t), T - t) &= \nabla_x \varphi\left(e^{-tA} x(t), T - t\right) \\ &= e^{-tA^\top} \nabla_z(z(t), T - t). \end{aligned}$$

Therefore when $t = 0$, then $z_0 = x_0$, $\varphi(z_0, T) = \varphi(x_0, T)$, and

$$\nabla_x(x_0, T) = \nabla_z(z_0, T).$$

This implies that (25) can be written as

$$t_{n+1} = t_n + \frac{\varphi(z_0, t_n)}{H_x\left(\nabla_z \varphi(z_0, t_n), z_0\right)}. \tag{26}$$

For the purpose of evaluating (26), there is no need to apply the change of variables as in (9). Therefore we have

$$H_x(p^*, z_0) = -z_0^\top A^\top p^* + \left\| -Q^\top B^\top p^* \right\|_*.$$

If its known that a single zero exists on the interval $T^* \in [0, t_{max}]$, then we use the Newton update from (26) to find t_{n+1}. With the value function computed at each Newton iteration, we can keep track of the updated interval $T^* \in [t_{min}, t_{max}]$, and use a bisection update if the Newton update of t_{n+1} is out side this interval. Once the minimum time to reach, T^* is found, the optimal control $u^*(t)$ can be found from the relation

$$\begin{aligned} \nabla_p H_z\left(\nabla_z \varphi(z_0, T^*), T^*\right) &= \nabla_p H_z(p^*, T^*) \\ &= e^{-(T^* - t)A} B u^*(t). \end{aligned} \tag{27}$$

V. Trajectory Generation with the Generalized Hopf Formula

Using the solution of the Hopf formula, we illustrate a dynamic programming point of view [3], [11] of the associated Hamilton–Jacobi equation to compute the optimal trajectory. We denote by $\gamma_z(s) \in \mathbb{R}^n$, with $s \in (0, T)$ as the state trajectory with $\gamma_z(0) = z_0$. Recall the fact that the solution of short-time Hopf formula (6) is itself a Fenchel transform [20]

$$\varphi(z, t) = (J^\star + tH)^\star(z). \tag{28}$$

Note that the minimizer of $J_z^\star(p)$ is the optimal terminal state, $\gamma_z(T)$, and if $J_z^\star$ is differentiable then can be found with

$$\gamma_z(T) = \frac{\partial}{\partial p} J_z^\star(p^*, 0), \tag{29}$$

where p^* is the solution to the Hopf formula in (6). Now consider the case where p^* is the solution to the generalized Hopf formula given in (12), and what follows is an analysis in the variable z. This implies that

$$\varphi(z_0, T) = -\Bigg\{ J_z^\star(p^*, 0) + \Delta t \sum_{i=0}^{N-1} H_z(p^*, t_i) - \langle z_0, p^* \rangle \Bigg\}, \tag{30}$$

where each of the quadrature time samples t_i are equally spaced by Δt seconds on the interval $[0, T]$ as defined by (17). Recall from Section II-B that we denote by $\varphi_z^\star(p, T)$ the Fenchel transform of $\varphi(z, t)$ with respect to the variable z at time $t = T$. If we write the Hopf formula with initial convex data $J(z, T-\Delta t) = \varphi(z, T-\Delta t)$, then we can find the level set evolution only for a short time, Δt, starting at value $\varphi(z, T-\Delta t)$ with

$$\varphi(z_0, T) = -\Big\{\varphi_z^\star(p^*, T-\Delta t) \\ = +\Delta t H_z(p^*, T-\Delta t) - \langle z_0, p^* \rangle\Big\}.$$

Following (29), the optimal state, with respect to z is

$$\gamma_z(\Delta t) = \frac{\partial}{\partial p}\varphi_z^\star(p^*, T-\Delta t). \tag{31}$$

From (28) we conclude that

$$\begin{aligned}\varphi_z^\star(p^*, T-\Delta t) &= (J^\star + \Delta t H_z)_p^{\star\star}(z, T-\Delta t) \\ &= J_z^\star(p^*, T-2\Delta t) \\ &+ \Delta t H_z(p^*, T-2\Delta t).\end{aligned} \tag{32}$$

The last line in (32) is due to the fact that if f is convex, proper, and lower semicontinuous, then $f^{\star\star} = f$. We form a recursive operation, repeating (32) until we reach time zero and get

$$\varphi(z_0, T) = -\Big\{\varphi_z^\star(p^*, 0) \\ + \Delta t \sum_{i=0}^{N-1} H_z(p^*, t_i) - \langle z_0, p^* \rangle\Big\},$$

which is equivalent to (30) since by definition $J(x, 0) = \varphi(x, 0)$. This suggests that the generalized Hopf formula with the integral in (12) approximated by quadrature is equivalent to the composition of many short-time Hopf formulas (6) of length Δt. Also, the recursion can be applied to the optimal terminal point from (31) as

$$\begin{aligned}\gamma_z(\Delta t) &= \frac{\partial}{\partial p}\varphi_z^\star(p^*, T-\Delta t) \\ &= \frac{\partial}{\partial p}\left\{J_z^\star(p^*, 0) + \Delta t \sum_{i=0}^{N-1} H_z(p^*, t_i)\right\}.\end{aligned}$$

This can equivalently used for the optimal trajectory at any time sample t_i as

$$\begin{aligned}\gamma_z(t_i) &= \frac{\partial}{\partial p}\left\{J_z^\star(p^*, 0) + \Delta t \sum_{k=0}^{i} H_z(p^*, t_k)\right\} \\ &= \frac{\partial}{\partial p}J_z^\star(p^*, 0) + \Delta t \sum_{k=0}^{i} \frac{\partial}{\partial p}H_z(p^*, t_k).\end{aligned} \tag{33}$$

If we are interested in the trajectory at each quadrature time sample, we don't have to recompute the sum for each t_k, since we can incrementally build the trajectory point from t_{k+1}. Note that each point in (33) is in terms of the state variable z, and can be found for x by applying inverse of the transform given in (9). Non rigorously we see that the time rate of change of the state trajectory in (33) is equal to the gradient with respect to p of the Hamiltonian, which satisfies Pontryagin's Maximum Principal [30], though it was derived using only the Hopf formula and basic principals of convex analysis.

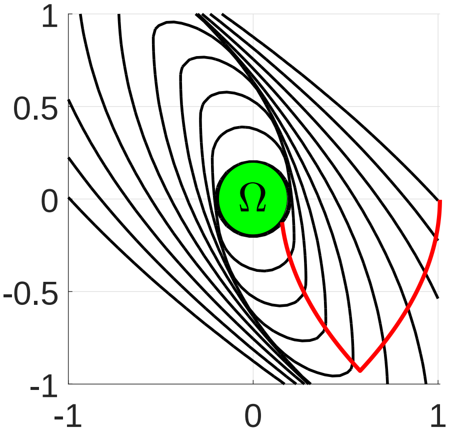


Fig. 1: The double integrator example. The zero level set evolution solved at various times shown in black. Direct trajectory generation shown in red. The terminal set, Ω, is shaded green.

VI. Results

The primal-dual method presented in Section III was implemented in MATLAB R2017a on a laptop equipped with an Intel Core i7-7500 CPU running at 2.70 GHz. For all experiments, 100 time samples were used for the quadrature in (18) and the dual step size was set to $\sigma = \frac{1}{\tau\|K\|^2}$, and $\theta = 1$. The initial conditions for all examples is $p^0 = x_0$ for the primal variable and $y^0 = Kp^0$ for the dual variable. Note that for some of the examples to follow, the action of the matrix exponential is known in closed form and that can be used for increased computational enhancement. Since in general this is not the case, we used [1] to numerically compute the action of the matrix exponential as to show dimensional scaling properties even for the most general case.

A. Double Integrator

We begin with the simple double integrator problem with system $\dot{x} = Ax + Bu$ with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where the state $x \in \mathbb{R}^2$ is position and velocity. This problem is selected since closed form optimal solutions exist and is low enough dimension to compare the level set evolution to that of grid based numeric techniques such as that in [23], [24]. We consider the control to be constrained to $u \in [-1, 1]$ and implies a control of $\mathcal{C} = \{u : |u| \leq 1\}$. After a change of variables (9), the Hamiltonian becomes

$$H_z(p, t) = \left| \left(-e^{-(T-t)A} B\right)^\top p \right| .$$

We chose the terminal set to be an ellipsoidal with

$$\Omega = \left\{x : \left\langle x, W^{-1} x \right\rangle \leq 1 \right\} . \quad (34)$$

where W is symmetric positive definite. For the initial cost function J_x, the elements of W are selected such that Ω is a circle with radius $r = 0.2$. The terminal cost function becomes

$$J_z(z, 0) = \langle z, V(0) z \rangle - 1$$

where $V(t) = e^{(T-t)A^\top} W^{-1} e^{(T-t)A}$. This gives

$$J_z^\star(p, 0) = \frac{1}{4} \left\langle p, V(0)^{-1} p \right\rangle + 1. \quad (35)$$

In this example (19) becomes

$$G(p) = \frac{1}{4} \left\langle p, V(0)^{-1} p \right\rangle - \langle z_0, p \rangle + 1, \quad (36)$$

which is quadratic and results in the following proximal point of G at p

$$(I + \tau \partial G)^{-1}(p) = \left(I + \frac{1}{2} \tau V(0)^{-1}\right)^{-1} (p + \tau z_0) . \quad (37)$$

Note that we don't need to compute the inverse of $V(0)$ since

$$V(0)^{-1} = e^{-TA} W e^{-TA^\top} .$$

Likewise, with $F_i^\star$ defined in (22), the proximal points of each $F_i^\star$ at y_i in (23) is given by

$$(I + \sigma \partial F_i^\star)^{-1}(y_i) = \Delta t \operatorname{sign}(w_i) \min(|w_i|, 0),$$

where $w_i = \frac{y_i}{\Delta t}$ andsign $(\beta) = 1$ if $\beta \geq 0$ and -1 otherwise.

We computed the solution to the Hamilton–Jacobi equation at each point on a grid, $[-1, 1]^2$, of 50 equally spaced points in each dimension. This was done to average execution time for a large number of initial conditions. The average computational time was $2.1\, ms$ per point on the grid and the zero contours of the value function for ten different times equally spaced on $t \in [0, T^*]$ are shown in Figure 1. The value of T^* we set as by solving for the minimum time to the zero level set for the initial state $z_0 = x_0 = (1, 0)^\top$ using the method presented in Section IV. The primal step size was set to $\tau = 10$.

The optimal trajectory starting at z_0 was computed following (33) with

$$\gamma_z(T^*) = \frac{\partial}{\partial p} J_z^\star(p^*, 0) = \frac{1}{2} V(0)^{-1} p^*,$$

since $J_x(x)$ is quadratic. The gradient of the Hamiltonian is

$$\frac{\partial}{\partial p} H_z(p^*, t_k) = -e^{-(T^* - t_k)A} B \\ \times \operatorname{sign}\left(\left(-e^{-(T^* - t_k)A} B\right)^\top p^*\right) .$$

The trajectory as computed in (33) is shown in red in Figure 1.

B. Unscented Optimal Control

We can utilize the favorable dimensional scaling of the proposed methods to generate solutions that are robust to system uncertainty. These uncertainties could be the initial state or system parameters such as mass, lift coefficients, or other properties. The idea is to augment the system to include samples that represent different initial conditions or parameters that share one, common control input. This system forms a tychastic [2] differential equation, since the parameters are fixed, but unknown at run time. The goal is to select a single control so the aggregate cost of all samples is optimized.

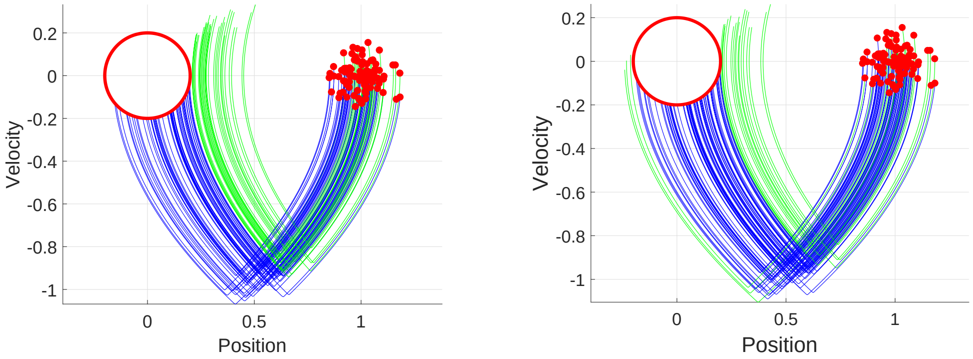
For uncertainties that can be modeled as Gaussian, we can choose the samples deterministically with the unscented transform [16], and these samples are typically referred to in literature as sigma points. This transform provides a second-order approximation of the moments of a Gaussian distribution propagated through a nonlinear function. It was developed for state estimation problems and in this context is known as the unscented Kalman filter [17], [34]. Using the unscented transform for sample selection to approximate the tychastic optimal control problem was developed by Ross et al. [33], [32]. The extra state dimensions that result from this technique is not as problematic with proximal splitting as it may be with other methods.

Let $x = \left(\chi^{[1]}, \chi^{[2]}, \ldots, \chi^{[2n+1]}\right)^\top$, with $\chi^{[i]} \in \mathbb{R}^n$ represent the new state vector augmented by $2n + 1$ states generated from the unscented transform and the new system becomes

$$\dot{x} = \begin{bmatrix} \dot{\chi}_1 \\ \dot{\chi}_2 \\ \vdots \\ \dot{\chi}_{2n+1} \end{bmatrix} = \begin{bmatrix} f_1\left(\chi^{[1]}, u\right) \\ f_2\left(\chi^{[2]}, u\right) \\ \vdots \\ f_{2n+1}\left(\chi^{[2n+1]}, u\right) \end{bmatrix},$$

subject to initial condition $z_0 = x_0 = \left(\chi_0^{[1]}, \chi_0^{[2]}, \ldots, \chi_0^{[2n+1]}\right)$. Take for example the problem of uncertainty in initial condition, $x_0 \sim \mathcal{N}(\mu, \Sigma)$. As an unscented control problem, the dynamics are the same, so $f_1 = f_2 = \cdots = f_{2n+1} = f$, and our augmented system is $\dot{x} = \tilde{A} x + \tilde{B} u$ with

$$\tilde{A} = \begin{bmatrix} A & \cdots & & 0 \\ \vdots & A & & \vdots \\ & & \ddots & \\ 0 & \cdots & & A \end{bmatrix},$$



(a) 100 random samples of initial conditions with the existing control formulation. 66 samples reached the goal state and 34 missed.

(b) 100 random samples of initial conditions with uncertainty modeled as an unscented control problem. 79 samples reached the goal state and 21 missed.

Fig. 2: An example of unscented control applied to a time-optimal double integrator problem with uncertain initial state. Red are samples of the initial states. Blue trajectories are random initial states that reached the goal state and green are trajectories that missed the goal state. Best viewed in color.

and

$$\tilde{B} = \begin{bmatrix} B \\ B \\ \vdots \\ B \end{bmatrix}.$$

The initial state becomes $x_0 = \left\{\chi_0^{[i]}\right\}_{i=1,\ldots,2n+1}$, where each $\chi_0^{[i]}$ is formed by the unscented transform with mean μ and covariance Σ. The mean square error of the terminal state relative to some goal state, $\hat{x}_T$ is found with the unscented transform by

$$\text{MSE} \approx \sum_{i=1}^{2n+1} w_m^{[i]} \left(\chi_T^{[i]} - \hat{x}_T\right)^\top \left(\chi_T^{[i]} - \hat{x}_T\right), \tag{38}$$

where w_m is the mean weight factor[1], and $\chi_T^{[i]}$ is the terminal state for the i-th unscented sigma point. For this example, we wish to find the minimum time to reach the origin subject to the constraint that the mean square error is less than some threshold ℓ. This can be found from (38) with

$$\text{MSE} = \mathbb{E}\left[\left(\hat{x}_T - x\left(T\right)\right)^\top \left(\hat{x}_T - x\left(T\right)\right)\right] \leq \ell.$$

We select the origin as the goal with $\hat{x} = (0, 0)^\top$, and formulate as an unscented control problem. The trace of the terminal covariance can be represented by the quadratic

$$J\left(z\right) = \langle z, V\left(0\right) z\rangle - \ell \tag{39}$$

with $V\left(t\right) = e^{(T-t)A^\top} W^{-1} e^{(T-t)A}$ and

$$W = \begin{bmatrix} \left(w_m^{[1]}\right)^{-1} I_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left(w_m^{[2n+1]}\right)^{-1} I_2 \end{bmatrix},$$

where I_2 is the 2×2 identity matrix.

Figure 2 shows example trajectories when the initial condition has random perturbations, with $\mu = x_0$ and $\Sigma = \pi^2 I_2$, for a double integrator problem. The standard deviation was set to $\pi = 0.0667$. If the initial state is *exactly* what was used to compute the optimal control, then the trajectory generated reaches the goal and is time-optimal. However, if the initial state is perturbed, some trajectories miss the intended goal entirely. In this particular example, it is especially sensitive to perturbations to the "right" in the spatial x direction and is shown in the Figure 2a on the left. Of 100 random initial states, 34 miss the goal state. For the resulting 10 dimensional unscented control problem with $\tau = 0.5$, the HJ solutions were found on average $4.0\,ms$ per point. Trajectory samples are shown in the Figure 2a on the right, the number of trajectories that miss the goal is reduced to only 5.

C. Dimensional Scaling

Next we seek to analyze how the proposed method scales with dimension. We can construct a problem similar to that presented in Section VI-B but by selecting samples at random as opposed to using the unscented transform. Constructing a problem in this fashion is something not typically done in practice, but allows us to vary the number of random samples, and hence alter the dimension of the problem in a consistent and uniform way. The initial state for k samples becomes $x_0 = \left\{\chi_0^{[i]}\right\}_{i=1,\ldots,k}$, where each $\chi_0^{[i]}$ is an independent and identically distributed (iid) random vector drawn according to $\chi_0^{[i]} \sim \mathcal{N}\left(\mu, \Sigma\right)$. To again penalize the trace, the terminal cost function is the same as (39) except W is now defined by

$$W = \begin{bmatrix} kI_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & kI_2 \end{bmatrix}.$$

Figure 3 shows the average computational time and average iterations to convergence for dimensions ranging from 30

[1] For more information on the generation of the unscented sigma points, and their weights, see [34].

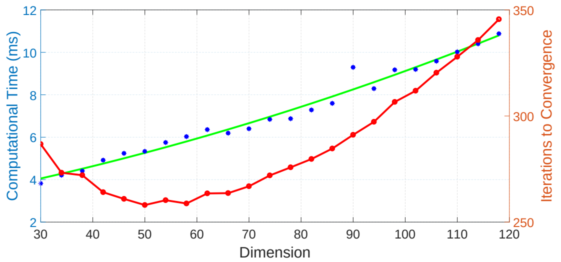


Fig. 3: Dimensional scaling properties as described by the experiment in Section VI-C. The blue dots represent average computational time for an experiment and the green line is the least squares polynomial fit to computational time. The red displays average iterations to convergence.

to 120. The green line in the figure is the least squares polynomial fit for average computation time in milliseconds. If we let d denote problem dimension for the experiments, the fit is $t_{\text{comp}} = 2.382 \times 10^{-4}d^2 + 0.0414d + 2.598$. Note the extremely small quadratic coefficient.

VII. Conclusion

Presented is a parallel primal-dual method to solve the Hamilton–Jacobi equation for time-optimal control using the generalized Hopf formula. We empirically showed how the method scales approximately quadratic with dimension, though with small quadratic coefficient. The experiments were shown using the Matlab, and simple implementation in a compiled language could provide significant computational improvement. Future work includes increased experimentation with different systems, the use of more advanced quadrature methods, adaptive step sizes, time-varying systems, and state dependent Hamiltonians.

References

[1] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33(2):488–511, 2011.

[2] J.-P. Aubin, L. Chen, O. Dordan, A. Faleh, G. Lezan, and F. Planchet. Stochastic and tychastic approaches to guaranteed ALM problem. *Bulletin Français d'Actuariat*, 12(23):59–95, 2012.

[3] R. E. Bellman. *Dynamic Programming*, volume 1. Princeton University Press, 1957.

[4] R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 2015.

[5] A. R. Bryson and Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation and Control*. CRC Press, 1975.

[6] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[7] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton–Jacobi equations arising from optimal control and differential games problems. *Journal of Scientific Computing*, pages 1–27, 2016.

[8] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[9] J. Darbon. On convex finite-dimensional variational methods in imaging sciences and Hamilton-Jacobi equations. *SIAM Journal on Imaging Sciences*, 8(4):2268–2293, 2015.

[10] J. Darbon and S. Osher. Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere. *Research in the Mathematical Sciences*, 3(1):19, 2016.

[11] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010.

[12] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk. Adaptive primal-dual hybrid gradient methods for saddle-point problems. *arXiv preprint arXiv:1305.0546*, 2013.

[13] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[14] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[15] E. Hopf. Generalized solutions of non-linear equations of first order. *Journal of Mathematics and Mechanics*, 14:951–973, 1965.

[16] S. J. Julier. The scaled unscented transformation. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, pages 4555–4559. IEEE, 2002.

[17] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[18] M. R. Kirchner, R. Mar, G. Hewer, J. Darbon, S. Osher, and YT Chow. Time-optimal collaborative guidance using the generalized Hopf formula. *arXiv preprint arXiv:1709.06590*, 3, 2017.

[19] A. B. Kurzhanski and P. Varaiya. *Dynamics and Control of Trajectory Tubes: Theory and Computation*, volume 85. Springer, 2014.

[20] P. L. Lions and J.-C. Rochet. Hopf formula and multitime Hamilton-Jacobi equations. *Proceedings of the American Mathematical Society*, 96(1):79–84, 1986.

[21] J. H. Mathews and K. D. Fink. *Numerical Methods using MATLAB*, volume 4.

[22] I. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[23] I. M. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35(2):300–329, 2008.

[24] I. M. Mitchell and J. A. Templeton. A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems. In *HSCC*, volume 5, pages 480–494. Springer, 2005.

[25] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153. Springer Science & Business Media, 2006.

[26] N. P. Osmolovskii. *Calculus of Variations and Optimal Control*, volume 180. American Mathematical Society, 1998.

[27] N. F. Palumbo, R. A. Blauwkamp, and J. M Lloyd. Modern homing missile guidance theory and techniques. *Johns Hopkins APL Technical Digest*, 29(1):42–59, 2010.

[28] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[29] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 1762–1769. IEEE, 2011.

[30] I. M. Ross. *A primer on Pontryagin's principle in optimal control*. Collegiate Publishers, 2015.

[31] I. M. Ross and M. Karpenko. A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2):182–197, 2012.

[32] I. M. Ross, M. Karpenko, and R. J. Proulx. Path constraints in tychastic and unscented optimal control: Theory, application and experimental results. In *American Control Conference (ACC), 2016*, pages 2918–2923. IEEE, 2016.

[33] I. M. Ross, R. J. Proulx, and M. Karpenko. Unscented guidance. In *American Control Conference (ACC), 2015*, pages 5605–5610. IEEE, 2015.

[34] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT press, 2005.