A Monte-Carlo HJ Reachability Sampling Scheme

Lekan Patrick Molu
Bala Cynwyd, PA

Venkatraman Renganathan
Cranfield University, London

Namhoon Cho
Seoul National University

Abstract

Backward reachable tubes (BRTs), computed via viscous Hamilton-Jacobi (HJ) partial differential equations, provide principled safety certificates for learned controllers and planning algorithms in trustworthy machine learning. However, classical grid-based HJ solvers require $O(M^n)$ memory footprint for M grid points per n state dimension. This renders them impractical for high-dimensional systems. We address this bottleneck with a local PDE linearization that enables a frozen-coefficient sampling scheme for the viscous HJ PDE: a generalized Cole-Hopf-type transformation reduces the nonlinear HJ equation to a sequence of linear heat equations whose solutions admit Gaussian heat-kernel representations. The value function and its spatial gradient are then recovered via roll-outs of Monte Carlo expectations on Gaussian densities, yielding a storage and grid-free algorithm that scales as $N \cdot n$ for N samples. This decoupling of memory from dimensionality enables reachability analysis on problems where grid-based methods are simply impossible. We prove a finite-sample concentration bound $O(N^{-1/2})$ error and conditional linear convergence for the introduced Monte-Carlo Picard iterative scheme. Numerical validation on pursuit-evasion games demonstrates relative L^2_{rel} errors of $0.03 - 0.20$, with $14 - 26$ second wall-clock times per 2D slice on a CPU. Crucially, the method scales with validation on up to (but not limited to) $n = 45$-dimensional multi-agent games.

1 Introduction

A central challenge in trustworthy machine learning is certifying that a learned controller, neural policy, or planning algorithm will not violate safety constraints at deployment time. This concern appears directly in safe reinforcement learning [Li et al., 2021], model-based RL with learned dynamics [Berkenkamp et al., 2017], and any pipeline where a learned model must operate in a continuous, possibly high-dimensional state space under adversarial or uncertain inputs. Backward reachable (reach-avoid) tubes or BRTs(BRATs) are the set of all states from which a system is guaranteed to reach (avoid) a target region regardless of disturbances that may affect its behavioral evolution during deployment; they provide a principled certificate for solving *reachability* problems. Computing a BRT amounts to solving a Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE) [Mitchell et al., 2005, Lygeros, 2004], typically under two competing inputs.

Central Contribution: Monte-Carlo Trajectory Optimization of Quasi-Linear BRTs

A *backward reachable tube* (BRT) is the set of initial states from which a controlled system is guaranteed to reach a target set within a given time horizon, for all adversarial inputs. Given system dynamics $\dot{\boldsymbol{x}} = f(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})$ with controller $\boldsymbol{u}$ and disturbance $\boldsymbol{w}$, the BRT is the zero sublevel set of a value function $\boldsymbol{v}^\delta$ that solves the viscous HJI PDE (HJI-RCBRT-Visc). Classical grid-based solvers [Mitchell et al., 2005, Molu, 2025] discretize the state space on a grid with M points per dimension; the resulting memory cost is $O(M^n)$, which grows exponentially in the state dimension n. For $n = 6$ with $M = 100$, this already requires 10^{12} grid cells. The proposed method replaces the grid with N Monte Carlo samples of state space trajectories, reducing spatial memory to $O(N \cdot n)$.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

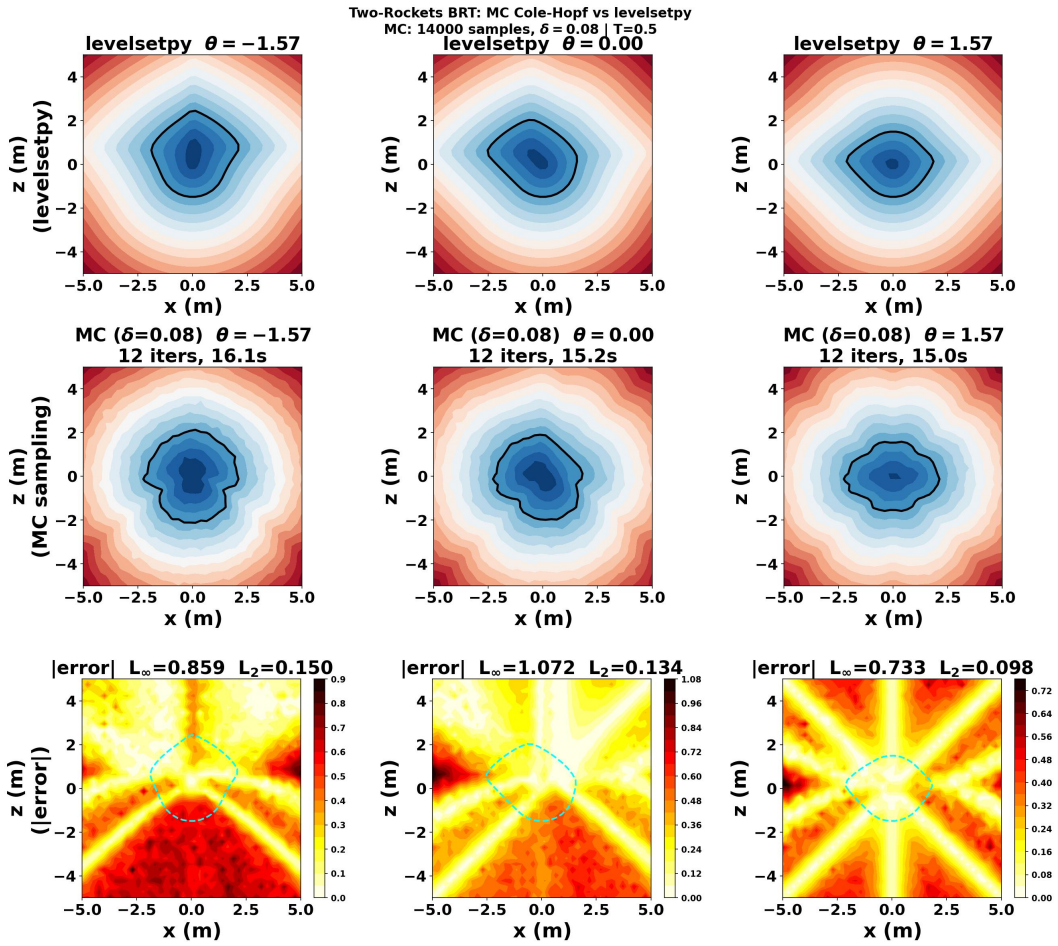

Figure 1: Rockets BRT slices: Grid-based `levelsetpy` from [Molu, 2024b] vs. Monte Carlo (ours). The bottom row shows a heat map of the pointwise errors at various relative vehicle orientations θ; errors are largest near the boundary of the usable part where $|D\boldsymbol{v}^{\delta}|$ is largest.

The major bottleneck for realizing the solutions to the BRT of complex systems is computational: classical HJ solvers scale as $O(M^n)$ in memory, where M is the number of grid points per dimension and n is the state dimension. While GPU-accelerated implementations [Molu, 2024b, 2025] reduce wall-clock time, they retain the same exponential memory footprint. Self-supervised, PINN-based approaches such as DeepReach [Bansal and Tomlin, 2021] train a neural network to minimize the HJ PDE residual directly via a physics-informed loss, without requiring a reference grid solution. However, the method scales moderately in dimensions (reportedly 9D) and accuracy degrades in the limit of higher-dimensional data. We introduce a fundamentally different approach that replaces grid storage with a sampling scheme so that the memory cost is $O(N \cdot n)$ and the algorithm is grid-free.

We consider first-order nonlinear scalar HJ PDEs of the form

$$\boldsymbol{v}_t + \boldsymbol{H}(t; \boldsymbol{x}, D\boldsymbol{v}) = 0 \text{ in } \Omega \times (0, T], \quad \boldsymbol{v}(0; \boldsymbol{x}) = \boldsymbol{g}(0; \boldsymbol{x}) \text{ on } \partial\Omega \times \{t = 0\} \tag{HJ-IVP}$$

where the state $\boldsymbol{x}$ belongs to the open set $\Omega \subseteq \mathbb{R}^n$; $\boldsymbol{v}_t$ denotes the partial time derivative of $\boldsymbol{v}(t; \boldsymbol{x})$; $\boldsymbol{H}(t; \boldsymbol{x}, D\boldsymbol{v})$ is the Hamiltonian, continuously defined on $\mathbb{R}^n$, with $D\boldsymbol{v}$ representing the spatial gradient; and $\boldsymbol{g}(\boldsymbol{x})$ being bounded and uniformly continuous (BUC) on $\mathbb{R}^n$.

Fixing a viscosity parameter $\delta > 0$, Crandall and Lions [Crandall and Lions, 1984] introduced the *vanishing viscosity* solution, $\boldsymbol{v}^{\delta}$, of (HJ-IVP) i.e. ,

$$\boldsymbol{v}_t^{\delta} + \boldsymbol{H}(t; \boldsymbol{x}, D\boldsymbol{v}^{\delta}) = \frac{\delta}{2}\Delta\boldsymbol{v}^{\delta} \text{ in } \Omega \times (0, T]; \quad \boldsymbol{v}^{\delta}(0; \boldsymbol{x}) = \boldsymbol{g}(0; \boldsymbol{x}) \text{ on } \partial\Omega \times \{t = 0\} \tag{HJ-Visc}$$

such that (HJ-Visc) satisfies the uniform convergence bound

$$\sup_{t\in(0,T]}\sup_{\boldsymbol{x}\in\mathbb{R}^n}\left|\boldsymbol{v}(t;\boldsymbol{x})-\boldsymbol{v}^{\delta}(t;\boldsymbol{x})\right|\leq k\sqrt{\delta} \tag{1}$$

for a constant $k>0$. *Our key insight is that* (HJ-Visc) *can be reduced, via a generalized Cole-Hopf-type transformation, to a sequence of linear heat equations admitting explicit Gaussian heat-kernel solutions.* This yields a locally linearized PDE. The scheme is *not* an exact Cole-Hopf reduction for general Hamiltonians; it is an iterative (Picard) approximation in which the nonlinear coefficient is frozen at each step and updated after the linear solution is found. The exact reduction holds for the quadratic case $\boldsymbol{H}=\frac{1}{2}|p|^2$, where the frozen coefficient is constant and the residual vanishes identically.

Connections to ML. Beyond reachability, the proposed scheme connects to several active ML research directions. (i) *Safe RL and policy verification*: certifying that a learned policy satisfies safety constraints requires computing BRTs for the closed-loop system; the memory bottleneck of grid-based solvers is the primary obstacle to safe RL in high-dimensional spaces [Li et al., 2021, Berkenkamp et al., 2017]. (ii) *Model-based RL*: planning algorithms that learn a dynamics model and then verify safety via HJ reachability are limited to low-dimensional state spaces [Bansal and Tomlin, 2021]. (iii) *Diffusion-based methods*: the Gaussian heat-kernel representation derived here is structurally similar to score-matching objectives and denoising score functions [Chaudhari et al., 2018, Heaton et al., 2024], opening potential bridges to score-based generative models for safety certificates.

Contributions. Our contributions are as follows. (i) A generalized Cole-Hopf-type transformation decouples the nonlinear viscous HJ equation into linear heat equations with Gaussian heat-kernel solutions; this enables a frozen-coefficient quasi-linearization. (ii) A sampling-based Algorithm 1) with $O(N\cdot n)$ memory complexity that scales with sample count rather than discretization dimension, offers a viable alternative to impractical grid-based $O(M^n)$ approaches in high dimensions. (iii) Theoretical guarantees: (a) a finite-sample concentration bound of $O(N^{-1/2})$ for the Monte Carlo value estimator, independent of state dimensions (Theorem 2.5), and (b) conditional linear convergence of the Picard iteration with explicit contraction constant (Theorem 2.9). (iv) Validation on two-player pursuit-evasion games achieving L^2_{rel} errors of 0.03–0.20 (below the $O(\sqrt{\delta})\approx 0.28$ viscosity bound) in 12–15 iterations; and scalable demonstration on 45-dimensional multi-agent problems where grid-based discretization are currently computationally prohibitive.

The rest of this work is structured as follows. Readers not familiar with existing literature can consult the background in Appendix A. Section 2 derives the quasi-linearization of (HJ-Visc), states the algorithm, and establishes the theoretical guarantees. Section 3 presents numerical results and § 4 concludes the paper. All proofs appear in Appendix B, while additional numerical experiments are provided in Appendix C.

2 Transformation of the HJ PDEs

We now transform the nonlinear viscous HJ PDE (HJ-Visc) into a linearized form and extract its solution via Gaussian heat-kernel expectations. All proofs are in Appendix B.

HJ Sign Convention

Throughout this paper we adopt the backward-reachability viscosity-solution convention:

$$\boldsymbol{v}_t+\boldsymbol{H}(t;\boldsymbol{x},D\boldsymbol{v})=0,$$

where the Hamiltonian is defined via the zero-sum differential game formulation (Appendix A.5):

$$\boldsymbol{H}(t;\boldsymbol{x},p)=\max_{\boldsymbol{u}}\min_{\boldsymbol{w}}\langle p,f(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w})\rangle.$$

Negative signs appearing in the Hamiltonian expressions in Appendix A arise from the backward-time transformation [Mitchell, 2004] and the min-over-max structure of reachability propagation. This convention is standard in reachability literature [Mitchell et al., 2005] and is consistent with the viscosity solution.

2.1 Exact Reduction and Quasi-Linearization: Two Cases

Let us first distinguish for the interpretation of our contribution.

Exact case ($\boldsymbol{H} = \frac{1}{2}|p|^2$). When the Hamiltonian is purely quadratic in the co-state, setting $\boldsymbol{c} = 1/\delta$ (a constant) and $\boldsymbol{\omega}^\delta := \exp(-\boldsymbol{v}^\delta/\delta)$ is an *exact* Cole-Hopf transformation: $\boldsymbol{\omega}^\delta$ satisfies the homogeneous heat equation $\boldsymbol{\omega}^\delta_t = (\delta/2)\Delta\boldsymbol{\omega}^\delta$ with no residual. In this case what follows below is exact.

Quasi-linearization. For a general Hamiltonian $\boldsymbol{H}(t; \boldsymbol{x}, p)$, define the spatially-varying coefficient,

$$\boldsymbol{c}(t; \boldsymbol{x}) = \frac{2}{\delta} \cdot \boldsymbol{H}^\delta / |D\boldsymbol{v}^\delta|^2, \tag{2}$$

where $\boldsymbol{H}^\delta := \boldsymbol{H}(t; \boldsymbol{x}, D\boldsymbol{v}^\delta)$. The transformation $\boldsymbol{\omega}^\delta := \exp(-\boldsymbol{c}\boldsymbol{v}^\delta)$ applied to (HJ-Visc) yields a heat equation with a non-zero residual term $\boldsymbol{R}(t; \boldsymbol{x})$ that involves spatial and temporal derivatives of $\boldsymbol{c}$ (see Appendix B for the derivation). Our proposed algorithm treats this as a *Picard quasi-linearization*: at iteration k, the coefficient $\boldsymbol{c}^{(k)}$ is *frozen* at the current iterate, the resulting linear heat equation is solved for $\boldsymbol{\omega}^{(k+1)}$, the value $\boldsymbol{v}^{(k+1)}$ is recovered, the gradient $D\boldsymbol{v}^{(k+1)}$ is updated, and $\boldsymbol{c}^{(k+1)}$ is recomputed. When $\boldsymbol{c}^{(k)}$ stabilizes, $\boldsymbol{R} \to 0$ effectively. There is no joint solve of $\boldsymbol{c}$ with the unknown iterate; the apparent circularity in (2) is resolved by the frozen-coefficient interpretation. This distinction is stated precisely in Remark 2.4 and is the basis of the central Algorithm 1.

2.2 HJ Linearization

Throughout this section we replace $\boldsymbol{H}(t; \boldsymbol{x}, D\boldsymbol{v}^\delta)$ with $\boldsymbol{H}^\delta$ for brevity and we write $\boldsymbol{g}(\boldsymbol{x})$ for the terminal cost $\boldsymbol{g}(0; \boldsymbol{x})$. With $\boldsymbol{c}^{(k)}$ frozen, the transformation,

$$\boldsymbol{\omega}^\delta := \exp(-\boldsymbol{c}\boldsymbol{v}^\delta), \tag{Linear-Trans}$$

reduces (HJ-Visc) to a heat equation under the frozen-coefficient approximation (see Proposition 2.1). *For general nonlinear Hamiltonians, this transformation induces a residual term and should be interpreted as a quasi-linearization*, i.e. , the approximation is exact only when $\boldsymbol{H}^\delta$ is quadratic in p.

Proposition 2.1 (Heat-equation reduction under frozen coefficient)**.** *With $\boldsymbol{c}^{(k)}$ frozen as a spatially-varying coefficient (exact when $\boldsymbol{H} = \frac{1}{2}|p|^2$ and $\boldsymbol{c} = 1/\delta$; approximate otherwise), the transformed variable $\boldsymbol{\omega}^\delta := \exp(-\boldsymbol{c}\boldsymbol{v}^\delta)$ satisfies,*

$$\boldsymbol{\omega}^\delta_t - \frac{\delta}{2}\Delta\boldsymbol{\omega}^\delta = 0 \text{ in } \Omega \times (0, T], \quad \boldsymbol{\omega}^\delta = \exp\left(-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{y})\right) \text{ on } \partial\Omega \times \{t = 0\}. \tag{Linear-HJ}$$

The solution to (Linear-HJ) *admits the following unique, explicit Green's convolution representation,*

$$\boldsymbol{\omega}^\delta(t; \boldsymbol{x}) = \frac{1}{(\sqrt{2\pi\delta t})^n} \int_\Omega \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{2\delta t}\right) \exp\left(-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{y})\right) d\boldsymbol{y}, \quad \boldsymbol{x} \in \Omega,\ t > 0, \tag{3}$$

which can equivalently be written as a Gaussian expectation,

$$\boldsymbol{\omega}^\delta(t; \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \delta t)} \left[\exp\left(-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{y})\right)\right]. \tag{4}$$

Lemma 2.2 (Smoothed HJ Solution)**.** *Under the frozen-coefficient interpretation of* (2) *(or exactly when $\boldsymbol{H} = \frac{1}{2}|p|^2$), the solution to the smoothed Hamilton-Jacobi equation* (HJ-Visc) *is,*

$$\boldsymbol{v}^\delta(t; \boldsymbol{x}) = -\frac{1}{\boldsymbol{c}^{(k)}} \log\left\{\frac{1}{(\sqrt{2\pi\delta t})^n} \int_\Omega \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{2\delta t}\right) \exp\left(-\boldsymbol{c}^{(k)}\boldsymbol{g}(\boldsymbol{y})\right) d\boldsymbol{y}\right\}, \tag{5}$$

in $\Omega \times (0, T]$, where $\boldsymbol{c}^{(k)}$ is the frozen coefficient at iteration k. This can equivalently be written as,

$$\boldsymbol{v}^\delta(t; \boldsymbol{x}) = -\frac{1}{\boldsymbol{c}^{(k)}} \cdot \log \mathbb{E}_{\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \delta t)} \left[\exp\left(-\boldsymbol{c}^{(k)}\boldsymbol{g}(\boldsymbol{y})\right)\right]. \tag{6}$$

Lemma 2.3 (Spatial gradients of the HJ solution)**.** *Under the same frozen-coefficient assumption as Lemma 2.2, the spatial gradient of the value function $\boldsymbol{v}^\delta(t; \boldsymbol{x})$ admits the form,*

$$D\boldsymbol{v}^\delta = \frac{1}{t \cdot \delta \cdot \boldsymbol{c}^{(k)}} \left(\boldsymbol{x} - \frac{\mathbb{E}_{\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \delta t)} \left[\boldsymbol{y} \cdot \exp\left(-\boldsymbol{c}^{(k)}\boldsymbol{g}(\boldsymbol{y})\right)\right]}{\mathbb{E}_{\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \delta t)} \left[\exp\left(-\boldsymbol{c}^{(k)}\boldsymbol{g}(\boldsymbol{y})\right)\right]}\right), \tag{7}$$

where $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, \delta t)$, i.e. , with mean $\boldsymbol{x}$ and variance $\delta t I_n$.

Algorithm 1 Quasi-Linearization Algorithm Cole-Hopf

Fix: $\epsilon > 0$, $\boldsymbol{v}^{(0)}(t;\boldsymbol{x}) = g(\boldsymbol{x})$, $\boldsymbol{c}^{(0)} = \frac{2\,\boldsymbol{H}(t;\boldsymbol{x},D\boldsymbol{g})}{\delta\,|D\boldsymbol{g}|^2}$.

For $k = 0, 1, 2, \ldots$**:**

1. Freeze $\boldsymbol{c}^{(k)}$ at the current iterate.
2. Solve the heat equation $\boldsymbol{\omega}_t = \frac{\delta}{2}\Delta\boldsymbol{\omega}$ with initial data $\boldsymbol{\omega}^{(k)}(0;\boldsymbol{x}) = e^{-c^{(k)}g(\boldsymbol{x})}$.
3. Recover $\boldsymbol{v}^{(k+1)} = -(1/\boldsymbol{c}^{(k)})\log\boldsymbol{\omega}^{(k+1)}$.
4. Update $D\boldsymbol{v}^{(k+1)}$ and $\boldsymbol{c}^{(k+1)} = \frac{2\boldsymbol{H}(t;\boldsymbol{x},D\boldsymbol{v}^{(k+1)})}{\delta|D\boldsymbol{v}^{(k+1)}|^2}$.
5. Check convergence: $\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|/\|\boldsymbol{v}^{(k)}\| < \varepsilon$.

Remark 2.4 (Exact reduction vs. quasi-linearization). Lemmas 2.2 and 2.3 are exact when $\boldsymbol{H} = \frac{1}{2}|p|^2$, in which case $\boldsymbol{c} = 1/\delta$ is a constant and the residual $\boldsymbol{R}(t;\boldsymbol{x})$ in the linearized equation vanishes identically (see Appendix B). For general Hamiltonians, $\boldsymbol{c}^{(k)}$ depends on the unknown $D\boldsymbol{v}^{\delta}$; we resolve this by Algorithm 1, whereupon $\boldsymbol{c}^{(k)}$ is formed from the previous iterate, frozen, and updated after each linear solve. The formulas in both lemmas apply at each iteration with $\boldsymbol{c}^{(k)}$ in place of $\boldsymbol{c}$.

2.3 Relations to Backward Reachable Sets/Tubes

Since we now have the derivatives that constitute the viscous HJ equation,we can rewrite (HJI-RCBRT) as

$$\boldsymbol{v}_t^{\delta}(t;\boldsymbol{x}) + \min\{0, \boldsymbol{H}^{\delta} - \frac{\delta}{2}\Delta\boldsymbol{v}^{\delta}\} = 0, \quad \boldsymbol{v}^{\delta}(0;\boldsymbol{x}) = \boldsymbol{g}(0;\boldsymbol{x}). \tag{HJI-RCBRT-Visc}$$

so that the solution to (HJI-RCBRT-Visc) is from integrating the quantities given in Lemma 2.3. This is tantamount to sampling from the kernels of the given Gaussian densities. The invariant set $\mathcal{L}_0$ obtained at T is

$$\mathcal{L}_0(T) = \{\boldsymbol{x} \in \Omega \,|\, \boldsymbol{v}^{\delta}(0;\boldsymbol{x}) \leq 0\}, \tag{8}$$

and the RCBRT is as given in (HJI-RCBRT). Similarly, the viscous version of the *reach-avoid BRT* i.e. (HJI-RCBRAT) can be represented as

$$\min\{\boldsymbol{v}_t^{\delta}(t;\boldsymbol{x}) + \boldsymbol{H}^{\delta} - \frac{\delta}{2}\Delta\boldsymbol{v}^{\delta}, \boldsymbol{g}(t;\boldsymbol{x}) - \boldsymbol{\ell}(t;\boldsymbol{x})\} \leq 0, \quad \boldsymbol{v}^{\delta}(0;\boldsymbol{x}) = \boldsymbol{g}(0;\boldsymbol{x}).$$

(HJI-RCBRAT-Visc)

2.4 Weighted Importance Sampling of Reachable Sets

While we may not recover the exact value function system, the mean and variance of its derivatives can be evaluated with the kernel expectations of Lemma 2.2, and 2.3. Previously, self-supervised physics-informed neural network (PINN) approaches such as DeepReach [Bansal and Tomlin, 2021] train neural networks to minimize the HJ PDE residual directly without requiring external supervision or a reference grid solution. However, neural network-based approximation methods encounter inherent scaling limitations; DeepReach scales only to moderate dimensions (reported to 9D/10D in [Bansal and Tomlin, 2021]), and approximation accuracy degrades as dimension increases. ***Our sampling-based scheme addresses this bottleneck by decoupling the memory cost from dimensionality.***

We propose a sequential importance sampling (IS) scheme for a high-dimensional $\boldsymbol{x}$. Consequently, $\boldsymbol{v}^{\delta}$ can be devised for the target density $\mathbb{P}_{y\sim\mathcal{N}(\boldsymbol{x},\delta t)}$ to induce a chain-like decomposition $\boldsymbol{v}^{\delta} := (\boldsymbol{v}_1^{\delta}, \cdots, \boldsymbol{v}_d^{\delta}), 1 \leq d \leq T$ i.e. , $\boldsymbol{v}^{\delta}(t;\boldsymbol{x}) = \boldsymbol{v}^{\delta}(d_1;\boldsymbol{x}_1)\prod_{d=2}^{T}\boldsymbol{v}^{\delta}(d;\boldsymbol{x}_d)|\boldsymbol{v}^{\delta}(d;\boldsymbol{x}_{[1:d-1]})$ from which we can build an importance sampling scheme such as $q^{\delta}(t;\boldsymbol{x}) = q_1^{\delta}(d_1;\boldsymbol{x}_1)\prod_{d=2}^{T}q_d^{\delta}(d;\boldsymbol{x}_d)|q_d^{\delta}(d;\boldsymbol{x}_{[1:d-1]})$. Corollaries to Lemmas 2.2 and 2.3 in light of this importance sampling scheme are in the appendix. The quasi-linearization, essentially a Picard fixed-point iterative scheme, is stated in Algorithm 1. The heat-kernel expectation is estimated via Monte Carlo, while the resulting log-sum-exp estimator for the value function enjoys the sample guarantee given in §2.5.

2.5 Sampling Complexity and Convergence Guarantees

In this sub-section, we analyze the sample complexity of the scheme with coefficient $\boldsymbol{c}$ frozen and then generate a conditional linear-convergence guarantee as a fixed-point iteration on a finite collection of evaluation states.

Theorem 2.5 (Finite-sample concentration of the frozen-coefficient value estimator)**.** *Fix phase $(t;\boldsymbol{x}) \in [0,T) \times \Omega$ and a frozen coefficient $\boldsymbol{c} > 0$ (satisfied when the Hamiltonian is bounded away from zero in regions of smoothness). Let $\sigma \triangleq \sqrt{\delta(T-t)}, \boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{x}, \delta(T-t)I_n)$, and assume that the terminal cost is bounded, i.e. there exist constants $g_{\min} \leq g_{\max}$ such that $g_{\min} \leq \boldsymbol{g}(\boldsymbol{\zeta}) \leq g_{\max}$, $\forall\, \boldsymbol{\zeta} \in \Omega$.*

Let $Z \triangleq \exp(-\boldsymbol{c}\,\boldsymbol{g}(\boldsymbol{\zeta})), \mu \triangleq \mathbb{E}[Z], \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x}) \triangleq -\frac{1}{\boldsymbol{c}} \log \mu$. Given i.i.d. samples $\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_N \sim \mathcal{N}(\boldsymbol{x}, \delta(T-t)I_n)$, let $Z_i \triangleq \exp(-\boldsymbol{c}\,\boldsymbol{g}(\boldsymbol{\zeta}_i))$, $\bar{Z}_N \triangleq \frac{1}{N}\sum_{i=1}^N Z_i$, and $\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) \triangleq -\frac{1}{\boldsymbol{c}} \log \bar{Z}_N$. Let $\alpha \triangleq e^{-\boldsymbol{c} g_{\max}}$, $\beta \triangleq e^{-\boldsymbol{c} g_{\min}}$. Then $\alpha \leq Z_i \leq b$ almost surely, and for all $\varepsilon > 0$,

$$\mathbb{P}\left(|\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x})| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2N\mu^2(1-\exp(-\boldsymbol{c}\varepsilon))^2}{(\beta-\alpha)^2}\right). \tag{9}$$

Remark 2.6 (Tightness via Bernstein and Jensen bounds). The Hoeffding bound in (9) is asymptotically tight but ignores variance. When $\mathrm{Var}(Z) \ll (\beta-\alpha)^2/4$, a Bernstein tail bound yields strictly tighter concentration. Additionally, Corollary 2.7 uses the loose lower bound $\mu \geq \alpha = e^{-\boldsymbol{c} g_{\max}}$; when a sharper estimate (e.g., via Jensen, $\mu \geq e^{-\boldsymbol{c}\mathbb{E}[\boldsymbol{g}(\boldsymbol{\zeta})]}$) is available, the required sample size can be substantially reduced.

Corollary 2.7 (Explicit sample size independent of μ)**.** *In particular, since $\mu \geq \alpha$, it is sufficient to choose*

$$N \geq \frac{(\beta-\alpha)^2}{2\alpha^2(1-\exp(-\boldsymbol{c}\varepsilon))^2} \log\left(\frac{2}{\alpha}\right) \tag{10}$$

to guarantee $\mathbb{P}\left(|\widehat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x})| \geq \varepsilon\right) \leq \alpha$.

Assumption 2.8. Let $V = \mathbb{R}^M$ with the sup norm $\|v\|_\infty = \max_{1\leq m\leq M} |v_m|$, where $v_m \approx \boldsymbol{v}(t;\boldsymbol{x}_m)$. Assume the following hold on a closed admissible set $\mathcal{A} \subset V$.

1. There exists $G > 0$ such that $|\boldsymbol{g}(\boldsymbol{x})| \leq G$ for all $\boldsymbol{x} \in \mathbb{R}^n$.
2. There exist constants $0 < m_0 \leq P_*$ and $0 < c_{\min} \leq c_{\max} < \infty$ such that for every $v \in \mathcal{A}$ and each m,
$$m_0 \leq |p_m(v)| \leq P_*, \qquad c_{\min} \leq (\Gamma(v))_m \leq c_{\max}. \tag{11}$$
3. The Hamiltonian is Lipschitz in the co-state on the ball $|p| \leq P_*$: there exist constants $H_*, L_H > 0$ such that
$$|\boldsymbol{H}(t;\boldsymbol{x}_m,p)| \leq H_*, \qquad |\boldsymbol{H}(t;\boldsymbol{x}_m,p) - \boldsymbol{H}(t;\boldsymbol{x}_m,q)| \leq L_H|p-q| \tag{12}$$
for all $|p|, |q| \leq P_*$ and all m.
4. The reconstruction map is Lipschitz on $\mathcal{A}$: there exists $L_D > 0$ such that
$$\|\mathcal{G}(v) - \mathcal{G}(w)\|_\infty \leq L_D\|v-w\|_\infty, \qquad \forall v, w \in \mathcal{A}. \tag{13}$$
5. The admissible set is invariant under Λ, and the quantity
$$q \triangleq \frac{2G}{c_{\min}} \cdot \frac{2L_D}{\delta}\left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right) \tag{14}$$
satisfies $0 \leq q < 1$.

Theorem 2.9 (Contraction convergence of Algorithm Algorithm 1)**.** *Fix evaluation states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M \in \Omega$ and a time $t \in [0,T)$. Let $\mathcal{G} : V \to (\Omega)^M$ denote a stable deterministic gradient reconstruction operator, which may arise from smoothing, interpolation, kernel regression, or variance-controlled Monte Carlo estimation used by Algorithm 1, and write $p_m(v) \triangleq (\mathcal{G}(v))_m$. For $v \in V$,*

define the coefficient-update map by $(\Gamma(v))_m \triangleq 2\boldsymbol{H}(t;\boldsymbol{x}_m, p_m(v))/(\delta|p_m(v)|^2)$, $m = 1,\dots,M$, *and define the frozen-coefficient heat-kernel map by*

$$(\Phi(c))_m \triangleq -\frac{1}{c_m}\log \mathbb{E}_{\boldsymbol{\zeta}\sim\mathcal{N}(\boldsymbol{x}_m,\delta(T-t)I_n)}\left[\exp\left(-c_m \boldsymbol{g}(\boldsymbol{\zeta})\right)\right]. \tag{15}$$

Let $\Lambda \triangleq \Phi \circ \Gamma$ *so that Algorithm 1 is the iteration* $v^{(k+1)} = \Lambda(v^{(k)})$.

Then Λ *is a contraction on* $\mathcal{A}$. *Consequently,* Λ *has a unique fixed point* $v^* \in \mathcal{A}$, *and for every initial iterate* $v^{(0)} \in \mathcal{A}$ *the sequence generated by Algorithm 1 converges linearly to* v^* *with*

$$\|v^{(k+1)} - v^*\|_\infty \le q\|v^{(k)} - v^*\|_\infty \le q^{k+1}\|v^{(0)} - v^*\|_\infty. \tag{16}$$

Moreover,

$$\begin{aligned}\|\Gamma(v^{(k)}) - \Gamma(v^*)\|_\infty &\le \frac{2L_D}{\delta}\left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right)\|v^{(k)} - v^*\|_\infty,\\ \|v^{(k+1)} - v^{(k)}\|_\infty &\le q^k\|v^{(1)} - v^{(0)}\|_\infty,\end{aligned} \tag{17}$$

and the a posteriori error estimate

$$\|v^{(k)} - v^*\|_\infty \le \frac{q}{1-q}\|v^{(k)} - v^{(k-1)}\|_\infty \tag{18}$$

holds for every $k \ge 1$.

Remark 2.10. Theorem 2.9 is a convergence result for the *frozen-coefficient numerical map* implemented by Algorithm Algorithm 1. It shows conditional linear convergence under explicit regularity, nondegeneracy, and contraction hypotheses. It does not claim unconditional global convergence for arbitrary Hamiltonians, nor does it identify the fixed point with the exact solution of the original nonlinear HJ PDE without further assumptions.

2.6 Notes on Monte Carlo and Viscosity Approximation Errors

The standard MC estimator for (HJ-Visc) converges at rate $O(1/\sqrt{N})$ where N is the number of samples. The variance depends on $\boldsymbol{c}$ and the range of $\boldsymbol{g}$: when $\boldsymbol{c}\,\|\boldsymbol{g}\|_\infty \gg 1$ (small δ), the exponential weights become concentrated and the effective sample size shrinks. Importance sampling with a tilted proposal (e.g., Laplace approximation around the mode of $e^{-\boldsymbol{cg}}$) can reduce variance. By [Crandall et al., 1984], $\|\boldsymbol{v}^\delta - \boldsymbol{v}\|_\infty \le k\sqrt{\delta}$ where $\boldsymbol{v}$ is the inviscid viscosity solution. Thus, smaller δ improves the approximation quality but increases the MC variance (the weights become overly peaked). This creates a fundamental bias-variance trade-off controlled by δ.

3 Numerical Results

All experiments run on a single CPU core of an Intel Core i7-14700K processor (20 physical cores, 33.8 MiB L3 cache per core, base clock 5.6 GHz) with 31 GiB RAM running Ubuntu 22.04. The processor supports SIMD vectorization (AVX2, SSE4.2) which JAX and PyTorch leverage for batch operations. All reported metrics are averaged across $N_{\text{runs}} = 5$ independent trials with different random seeds; tables and figures report mean values with standard deviation error bars where applicable.

3.1 Territorial Defense: A Game of Two Rockets on a Plane

We adopt the rockets launch problem (see Fig. 7) of Dreyfus [Dreyfus, 1966] and cast it as a terminal differential game between two identical rockets, $\boldsymbol{P}$ and $\boldsymbol{E}$, on an $(x-z)$ cross-section of a Cartesian plane in [Molu, 2024a, 2025]. The game terminates when *capture* occurs, i.e. , the distance $\|\boldsymbol{PE}\|$ becomes less than a certain prespecified (scalar) quantity. The states of $\boldsymbol{P}$ and $\boldsymbol{E}$ are denoted as (x_p, x_e) respectively, driven by thrusts (u_p, u_e) in the (x,z)-plane. The motion of $\boldsymbol{P}$ relative to $\boldsymbol{E}$'s along the $(\boldsymbol{x}-\boldsymbol{z})$ plane includes the relative orientation, the control input, shown in Fig. 7 as $\theta = u_p - u_e$. Full derivation of the scheme is provided in Appendix A.9. **Scope.** The proposed method is not intended to outperform structured grid solvers in low-dimensional settings ($n \le 4$), where highly-optimized implementations exist; rather, its advantage emerges as dimensionality increases and grid storage becomes prohibitive.

Table 1: 15-agent multi-pursuer single-evader game with three speed regimes on $n = 45$.

Case	a_{evader}	a_{pursuers}	Iterations	$\varepsilon(k)$	Wall-clock (s)
1. (Evader faster)	2.0	1.0	15	0.003121	27.6
2. (Equal speed)	1.0	1.0	15	0.002454	30.0
3. (Pursuers faster)	1.0	2.0	15	0.003462	27.6

3.2 Dimensional Scalability: A 15 Rockets System in a Pursuit-Evasion Game

To validate the scalability benefits of Algorithm 1, we consider a 15-rocket multi-pursuer single-evader game with state dimension $n = 45$; this dimension is intractable for grid-based solvers. State is $\boldsymbol{x} = (x_1, y_1, \theta_1, \ldots, x_{15}, y_{15}, \theta_{15}) \in \mathbb{R}^{45}$, where each agent $i \in \{1, \ldots, 15\}$ has position (x_i, y_i), heading θ_i, and control input $u_i \in [-1, +1]$ regulating turn rate. Dynamics are:

$$\dot{x}_i = a_i \cos(\theta_i), \quad \dot{y}_i = a_i \sin(\theta_i), \quad \dot{\theta}_i = u_i, \tag{19}$$

with forward speeds $a_i \in \mathbb{R}_{>0}$. The evader (agent 15) seeks escape; capture occurs when any of 14 pursuers reach distance $r_{\text{capture}} \leq 1.5$ ft. Target set: $\phi(\boldsymbol{x}) = \min_{i \leq 14} \|\boldsymbol{x}_i^{\text{pos}} - \boldsymbol{x}_{15}^{\text{pos}}\|_2 - r_{\text{capture}}$.

We test three speed regimes to examine relative agent capabilities on the reachable set, (1.) evader speed $a_{\text{evader}} = 2.0$ ft/s, pursuer speeds $a_{\text{pursuers}} = 1.0$ ft/s; evader advantage; (2.) all agents equal speed $a = 1.0$ ft/s; balanced game; (3.) evader speed $a_{\text{evader}} = 1.0$ ft/s, pursuer speeds $a_{\text{pursuers}} = 2.0$ ft/s; pursuer advantage.

This $n = 45$ problem is computationally inaccessible to grid-based reachability solvers. A 101-point grid would require $101^{45} \approx 10^{90}$ cells, exceeding all conceivable storage. Our method operates at $O(N \cdot n) \approx 7.2$ MB per iteration, where the memory cost is independent of the state dimension exponent. The quasi-linear Picard iteration (Algorithm 1) computes the relative change in value function between successive iterations; Table 1 reports the final relative residual, $\varepsilon = \|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|/\|\boldsymbol{v}^{(k)}\|$ as a measure of iteration convergence.

Table 1 shows the results for 15-rocket pursuit-evasion across three speed regimes. Wall-clock times are reported on a 16-core Intel i7-14700K. The residual is the final relative L^2 change between Picard iterates. All three cases converge in 15 iterations with residuals below 10^{-2}, demonstrating stability across different game parameters. Memory consumption remains constant at 7.2 MB per iteration; wall-clock variance reflects Monte Carlo sampling fluctuations. The results demonstrate that the $O(N \cdot n)$ scaling becomes practical in 45D, where grid-based methods are intractable; the achievable fidelity (sub-1

3.3 Experimental Setup and Quantitative Results: Pursuit-Evasion Games

Viscosity parameter selection. The viscosity parameter δ controls the degree of smoothing in the Cole-Hopf transformation. By Crandall-Lions theory, the approximation error scales as $O(\sqrt{\delta})$, so smaller δ yields more accurate solutions but requires higher Monte Carlo sample counts to maintain accuracy. In our experiments, we empirically set $\delta = 0.08$–0.1, which balances approximation error $O(\sqrt{\delta}) \approx 0.28$ with manageable variance in the log-sum-exp estimator. The choice $\delta \in [0.05, 0.2]$ is robust across all benchmark problems; practitioners should adjust based on the required accuracy-to-computation trade-off.

Frozen-coefficient bias. The quasi-linearization approximates $|D\boldsymbol{v}^\delta|$ by $|D\boldsymbol{g}|$ at each Picard iterate, introducing a systematic bias that is largest near the zero level set where $|D\boldsymbol{v}^\delta|$ varies most rapidly. Theorem 2.9 guarantees convergence to a fixed point of the frozen-coefficient map, not to the true viscous solution; the gap between these two objects depends on how well $|D\boldsymbol{g}|$ tracks $|D\boldsymbol{v}^\delta|$ along the optimal trajectory, a quantity we do not bound here. Empirically, this bias manifests as larger L^∞ errors near the reachability boundary where the gradient is most active; the method mitigates this through the viscosity smoothing and iterative refinement of the frozen coefficient.

Numerical realization: We discretize the spatial and temporal domains as,

$$(x, z) \in (-100, +100], \quad \theta \in (-\pi/2, \pi/2], \quad t \in (0, 1], \tag{20}$$

with 1,000 temporal grid points and spatial step $\boldsymbol{x}_k - \boldsymbol{x}_{k-1} = 10^{-2}$.

Table 2: Pursuit-evasion games: Monte Carlo vs.
textttLevelSetPy error metrics on 2D (x, z) slices. Both CPU-based; MC uses $N = 14{,}000$ samples per quasi-linear iteration. Crandall-Lions theory gives approximation error bound $O(\sqrt{\delta})$ where $\delta = 0.08 \Rightarrow O(\sqrt{\delta}) \approx 0.283$ (viscosity-induced smoothing).

System	θ (rad)	L^∞	L^2_{rel}	MC time (s)	Iters
Rockets	$-\pi/2$	0.844	0.147	14.7	12
	0	1.044	0.134	13.9	12
	$\pi/2$	0.740	0.099	14.0	12
Dubins	$-\pi/2$	1.139	0.197	25.5	15
	0	0.821	0.032	24.9	15
	$\pi/2$	1.163	0.196	24.7	15

We set the target set as the ℓ_2-ball $\ell(\boldsymbol{x}) = \sqrt{x^2 + z^2}$ and run Algorithm 1 over the time range $(0, 1]$ with Dirichlet boundary conditions,

$$\boldsymbol{v}^\delta(0; \boldsymbol{x}) = \boldsymbol{g}(0; \boldsymbol{x}) \triangleq 0. \tag{21}$$

Spatial gradients (co-states) are computed via (6) and (7); the Hamiltonian is evaluated iteratively using $N = 14{,}000$ Monte Carlo samples per iteration according to Algorithm 1. The results are seen in Fig. 1.

Convergence. Algorithm 1 converges within 20 iterations across all three benchmark systems, with final relative residuals in the range 0.14–0.22. This is consistent with the linear convergence guarantee of Theorem 2.9: the contraction factor $q < 1$ under the stated regularity conditions implies that the residual sequence $\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|/\|\boldsymbol{v}^{(k)}\|$ decays geometrically, as observed empirically.

BRT geometry. Errors concentrate near the zero level-set boundary where $|\nabla v^\delta|$ is maximal, as expected in frozen-coefficient approximations [Crandall et al., 1992].

Table 2 reports point-wise error metrics (L^∞ and L^2_{rel}) for 2D slices on the 3D Dubins and Rockets pursuit-evasion benchmarks across three representative heading angles, evaluated on a uniform 40×40 grid with reference from `LevelSetPy`'s 45^3 solution interpolated to the same points. The L^∞ (point-wise maximum) error represents worst-case divergence, useful where outlier errors matter. The relative RMS error, L^2_{rel} weights by solution magnitude and captures overall accuracy. Notably, the empirical L^2_{rel} errors substantially undercut the worst-case viscosity bound $O(\sqrt{\delta}) \approx 0.283$ [Crandall et al., 1992]: at $\theta = 0$, both systems achieve $L^2_{\text{rel}} < 0.04$, indicating that the Monte Carlo sampling error is well-controlled relative to the Cole-Hopf approximation error. In contrast, L^∞ errors ($\sim 0.8 - 1.2$) are larger and driven by pointwise deviations near the zero level-set boundary where $|\nabla v^\delta|$ is maximal; this is expected in frozen-coefficient approximations [Crandall et al., 1992]. The 3D isosurface computation over $15{,}625$ sample points completes in 125 seconds (rockets) and 75.1 seconds (Dubins) on a single CPU, compared to 1.1 seconds (rockets) and 3.2 seconds (Dubins) for the dense $45^3 = 91{,}125$-point grids. The zero level-set boundary is extracted via marching cubes [Lorensen and Cline, 1987]. The wall-clock overhead of the Monte Carlo method reflects the sample count ($N = 14{,}000$ per iteration) necessary to balance variance in the Cole-Hopf estimator against quasi-linearization residuals, a tradeoff that becomes favorable in higher dimensions where grid storage becomes prohibitive.

Scalability. Algorithm 1 requires a memory footprint per iteration of $O(N \cdot n)$ for N samples per n state dimensions. With $N = 14{,}000$ and $n = 3$, each iteration allocates approximately 0.6 MB for samples and intermediate values. In contrast, the dense 45^3
textttLevelSetPy grid requires approximately 1.5 MB for the value function and gradient storage. Each 2D θ-slice completes in $14 - 26$ seconds on a single core, fully parallelizable across the CPU's 20 cores. This reduces aggregate wall-clock time proportionally with available core count and makes the method well-suited to parallel architectures (GPUs, HPC clusters) where grid-based methods become memory-prohibitive in dimensions $n \geq 5$.

4 Conclusion

We have presented a quasi-linearized, frozen-coefficient sampling scheme for the viscous Hamilton-Jacobi PDE arising in safety analysis of dynamical systems. By applying a generalized Cole-Hopf-type transformation — exact when $\boldsymbol{H} = \frac{1}{2}|p|^2$ and iteratively approximated via Picard quasi-linearization for general Hamiltonians — the nonlinear HJ equation is reduced to a sequence of linear heat equations whose solutions are Gaussian heat-kernel expectations. The value function and its spatial gradient are recovered from these expectations via Monte Carlo sampling, yielding a storage-free, grid-free algorithm with memory cost $O(N \cdot n)$ rather than the $O(M^n)$ cost of classical grid-based solvers.

Theorem 2.5 provides a finite-sample concentration bound for the frozen-coefficient Monte Carlo estimator, establishing a standard $O(N^{-1/2})$ error rate under explicit boundedness conditions on the terminal cost. Theorem 2.9 establishes conditional linear convergence of the Picard fixed-point iteration under Lipschitz and nondegeneracy assumptions, with an explicit contraction constant and a posteriori error estimate. Neither result claims unconditional global convergence for arbitrary Hamiltonians. Numerical experiments on three benchmark reachability problems i.e. a 3D rocket pursuit-evasion game, a 3D Dubins two-car game, and a double integrator plant confirm convergence with residuals of $0.14 - 0.22$.

The connection to safe reinforcement learning, policy certification, and model-based control with learned dynamics suggests several directions for future work. These include adaptive importance sampling to reduce variance in high dimensions, extension to systems with stochastic dynamics, and tighter integration with deep learning pipelines for scalable safety certification of learned policies.

References

S. Bansal and C. J. Tomlin. Deepreach: A Deep Learning Approach to High-dimensional Reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2021.

R. Bellman. *Dynamic programming*. Princeton University Press, 1957. ISBN 0-486-42809-5.

F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems 30*, pages 908–918. Curran Associates, Inc., 2017. URL `https://papers.nips.cc/paper/6692-safe-model-based-reinforcement-learning-with-stability-guarantees`.

S. P. Bhat and D. S. Bernstein. Continuous finite-time stabilization of the translational and rotational double integrators. *IEEE Transactions on automatic control*, 43(5):678–682, 1998.

P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5:1–30, 2018.

M. G. Crandall and P.-L. Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

M. G. Crandall and P.-L. Lions. Two Approximations of Solutions of Hamilton-Jacobi Equations. *Mathematics of Computation*, 43(167):1–19, 1984.

M. G. Crandall and A. Majda. Monotone Difference Approximations For Scalar Conservation Laws. *Mathematics of Computation*, 34(149):1–21, 1980.

M. G. Crandall, L. C. Evans, and P. L. Lions. Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations. *Transactions of the American Mathematical Society*, 282(2):487, 1984. ISSN 00029947.

M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

S. E. Dreyfus. Control Problems With Linear Dynamics, Quadratic Criterion, and Linear Terminal Constraints. Technical report, Rand Corp, Santa Monica Calif, 1966.

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.

L. Evans and P. E. Souganidis. Differential Games And Representation Formulas For Solutions Of Hamilton-Jacobi-Isaacs Equations. *Indiana Univ. Math. J*, 33(5):773–797, 1984. ISSN 0022-2518.

L. C. Evans. *Partial Differential Equations*, volume 19. American Mathematical Society, 2022.

H. Heaton, S. Wu Fung, and S. Osher. Global solutions to nonconvex problems by evolution of hamilton-jacobi pdes. *Communications on Applied Mathematics and Computation*, 6(2):790–810, 2024.

R. Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization.* Kreiger, Huntigton, NY, 1999.

D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. American Elsevier Publishing Company, Inc., New York, NY, 1970.

Y. Li, N. Li, H. E. Tseng, A. Girard, D. Filev, and I. Kolmanovsky. Safe reinforcement learning using robust action governor. In A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, and M. N. Zeilinger, editors, *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 1093–1104. PMLR, 07–08 Jun 2021. URL `https://proceedings.mlr.press/v144/li21b.html`.

P.-L. Lions. *Generalized solutions of Hamilton-Jacobi equations*, volume 69. London Pitman, 1982.

W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

J. Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.

A. Merz. The game of two identical cars. *Journal of Optimization Theory and Applications*, 9(5): 324–343, 1972.

I. Mitchell. A toolbox of level set methods, version 1.0. *The University of British Columbia, UBC CS TR-2004-09*, pages 1–94, July 2004.

I. Mitchell. A Robust Controlled Backward Reach Tube with (Almost) Analytic Solution for Two Dubins Cars. *EPiC Series in Computing*, 74:242–258, 2020.

I. M. 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. ISSN 00189286.

L. Molu. The Python LevelSet Toolbox (LevelSetPy). In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 8938–8945, 2024a. doi: 10.1109/CDC56724.2024.10886640.

L. Molu. The python levelset toolbox (levelsetpy). In *IEEE 63rd Conference on Decision and Control (CDC)*, pages 8938–8945, 2024b. doi: 10.1109/CDC56724.2024.10886640.

L. Molu. LevelSetPy: A GPU-Accelerated Package for Hyperbolic Hamilton-Jacobi Partial Differential Equations' Solubility. *The ACM Transactions on Mathematical Software*, 2025.

S. Osher and R. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. *Applied Mechanics Reviews*, 57(3):B15–B15, 2004. ISSN 0003-6900.

S. Osher and J. A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. ISSN 10902716.

W. M. Wonham. Linear multivariable control: A geometric approach. *Applications of Mathematics*, 10, 1985.

A Background and Preliminaries (Unabridged).

This section provides a context for the contribution of this work.

We first introduce the notations that are commonly used throughout the rest of this article. Reachable sets within the context of two person games [Isaacs, 1999] and their accompanying "viscous" terminal HJ PDE [Evans and Souganidis, 1984] are then introduced. This is followed by the HJ-Isaacs (HJI) PDEs commonly used to characterize reachable sets.

A.1 Notations and Terminologies

Conventions: Upper-case and lower-case bold font Roman letters are matrices and vectors, respectively; calligraphic letters are sets. Time variables e.g. t_0, t, τ, T will always be real positive numbers. The n-dimensional Euclidean space is $\mathbb{R}^n$; the set of natural numbers is $\mathbb{N}$. The natural log of T is denoted $\log T$. The state $\boldsymbol{x}$ belongs to the open set $\Omega \subseteq \mathbb{R}^n$. The HJ value function has a terminal value defined on the boundary of Ω, denoted $\partial\Omega$. The closure of Ω is $\bar{\Omega}$. The dot product is signified by $\langle \cdot, \cdot \rangle$. We will use D to denote the gradient of a function $\boldsymbol{x}$ of n variables and $\nabla \boldsymbol{w} = (\boldsymbol{w}_{x_1}, \cdots, \boldsymbol{w}_{x_n}, \boldsymbol{w}_t) := (D\boldsymbol{w}, \boldsymbol{w}_t)$ as the full gradient of a function of $n+1$ variables, $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x}, t)$. Where clarity is necessary, we may write $D_y w$ to mean the spatial derivative of w with respect to the vector $\boldsymbol{y}$.

The unique solution $\xi \in \mathbb{R}^n$ of the dynamical system $\dot{\boldsymbol{x}}(\tau) = f(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})$, influenced by a pursuing player $\boldsymbol{P}$ (with control $\boldsymbol{w} \in \mathcal{W} \subseteq \mathbb{R}^p$) and its evading pair $\boldsymbol{E}$ (with control $u \in \mathcal{U} \subseteq \mathbb{R}^m$), results in system trajectories $\xi_{\boldsymbol{x},\tau}^{\boldsymbol{u},\boldsymbol{w}}$ that are obtained by optimizing a scalar-valued payoff functional, $\boldsymbol{\ell}(\xi_{\boldsymbol{x},t}^{\boldsymbol{u},\boldsymbol{w}}(\tau))$ i.e.

$$\mathcal{J}(\tau; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) = \min_{\tau \in [t,T]} \boldsymbol{\ell}(\xi_{\boldsymbol{x},t}^{\boldsymbol{u},\boldsymbol{w}}(\tau)) \qquad \text{(Payoff function)}$$

within a two-person differential game for a fixed $T \geq \tau > t$, where $\boldsymbol{\ell}$ can be a signed distance to the (boundary of the) target set/region, for example. The controllers belong in compact sets which are measurable functions i.e. $\bar{\mathcal{U}} \equiv \boldsymbol{u} : [t, T] \to \mathcal{U}$, $\bar{\mathcal{W}} \equiv \boldsymbol{w} : [t, T] \to \mathcal{W}$.

A.2 Robustly Controlled Backward Reachable Set and Tube

Suppose that the goal of $\boldsymbol{P}$ is to drive the system into a user-specified target region $\mathcal{L}_0(\tau)$ within $\tau \leq T$ time steps of playing the game; while $\boldsymbol{E}$ simultaneously seeks to prevent this from happening. The invariant set $\mathcal{L}_0$ obtained at T, i.e.

$$\mathcal{L}_0(T) = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \boldsymbol{v}(0; \boldsymbol{x}) \leq 0\} \qquad \text{(Target-Set)}$$

is "*robustly controlled*" for the "*distance-to-target-set*" cost $\boldsymbol{g}(0; \boldsymbol{x})$, with constraints $|\boldsymbol{g}(0; \boldsymbol{x})| \leq k,\ |\boldsymbol{v}(0; \boldsymbol{x}) - \boldsymbol{v}(t; \hat{\boldsymbol{x}})| \leq k|\boldsymbol{x} - \hat{\boldsymbol{x}}|$ where $k > 0$ is a constant, and all $-T \leq t \leq 0$, $\{\hat{\boldsymbol{x}},\ \boldsymbol{x} \in \mathbb{R}^n\}$[1]. The distance to $\mathcal{L}_0(\tau)$ is typically found by optimizing $\boldsymbol{\ell}(\boldsymbol{x}, t)$ as in (Payoff function). The HJI PDE (A.15) and Hamiltonian (A.16) for this set admit the form

$$\boldsymbol{v}_t(t; \boldsymbol{x}) + \min\{0, \boldsymbol{H}\,(t; \boldsymbol{x}, D\boldsymbol{v}(t; \boldsymbol{x}))\} = 0, \quad \boldsymbol{v}(0; \boldsymbol{x}) = \boldsymbol{g}(0; \boldsymbol{x}). \qquad \text{(HJI-RCBRT)}$$

For the safety problem setup in (A.15), the corresponding *robustly controlled backward reachable* ***tube*** (RCBRT) [Mitchell, 2020] on $(0, T]$ is the closure of the open set,

$$\mathcal{L}([-T, 0], \mathcal{L}_0) = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \exists\, \beta \in \mathcal{B}(t) \forall\, \boldsymbol{u} \in \mathcal{U}(t), \qquad \exists\, \bar{\tau} \in [-T, 0], \boldsymbol{\xi}(\bar{\tau}) \in \mathcal{L}_0\}, \qquad \text{(Target-Tube)}$$

Read: The set of states from which there exists a strategy of $\boldsymbol{P}$ such that, for all controls of $\boldsymbol{E}$, the resulting trajectory *reaches and remains in the target set* within the interval $[-T, 0]$. Following Lemma 2 of [Mitchell et al., 2005], the states in the reachable set admit the following properties w.r.t the value function $\boldsymbol{v}$,

$$\boldsymbol{x}(t) \in \mathcal{L}(\cdot) \implies \boldsymbol{v}(t; \boldsymbol{x}) \leq 0,\ \boldsymbol{v}(t; \boldsymbol{x}) \leq 0 \implies \boldsymbol{x}(t) \in \mathcal{L}(\cdot). \qquad \text{(Zero Levelset)}$$

In *backward reach avoid tubes*, the agent must avoid the unsafe region at all times. We can write (HJI-RCBRT) a *robustly controlled backward reach-avoid tube* (RCBRAT) as,

$$\min\{\boldsymbol{v}_t(t; \boldsymbol{x}) + \boldsymbol{H}\,(t; \boldsymbol{x}, D\boldsymbol{v})\,, \boldsymbol{g}(t; \boldsymbol{x}) - \boldsymbol{\ell}(t; \boldsymbol{x})\} \leq 0, \quad \boldsymbol{v}(0; \boldsymbol{x}) = \boldsymbol{g}(0; \boldsymbol{x}). \qquad \text{(HJI-RCBRAT)}$$

[1] Time is reversed in BRT computational scenarios.

A.3 Dynamic Programming and Two-Person Games.

The formal relationships between the dynamic programming (DP) optimality condition for the *value* in differential two-person zero-sum games, and the solutions to PDEs that solve "min-max" or "max-min" type nonlinearity (the Isaacs' equation) were presented in [Isaacs, 1999]. Essentially, Isaacs' claim was that if the *value* functions are smooth enough, then they solve certain first-order partial differential equations (PDE) problems with "max-min" or "min-max"-type nonlinearity. However, the DP value functions are seldom regular enough to admit a solution in the classical sense. "Weaker" solutions, on the other hand [Lions, 1982, Evans and Souganidis, 1984, Crandall et al., 1984, Crandall and Majda, 1980], provide generalized "viscosity" solutions to HJ PDEs under relaxed regularity conditions; these viscosity solutions are not necessarily differentiable anywhere in the state space, and the only regularity prerequisite in the definition is continuity [Crandall and Lions, 1983]. However, wherever they are differentiable, they satisfy the upper and lower values of HJ PDEs (discussed in § A.4) in a classical sense. Thus, they lend themselves well to many real-world problems existing at the interface of discrete, continuous, and hybrid systems [Lygeros, 2004, Osher and Sethian, 1988, Mitchell, 2020, Evans and Souganidis, 1984, Mitchell et al., 2005].

Matter-of-factly, viscosity solutions to *Cauchy-type*[2] HJ Equations are highly useful in backward reachability analysis [Mitchell et al., 2005]. For a state $\boldsymbol{x} \in \Omega$ and a fixed time t: $0 \leq t < T$, suppose that the set of all controls for players $\boldsymbol{P}$ and $\boldsymbol{E}$ are respectively

$$\bar{\mathcal{U}} \equiv \{\boldsymbol{u} : [t,T] \to \mathcal{U} | \boldsymbol{u} \text{ measurable}, \mathcal{U} \in \mathbb{R}^m\}, \quad \bar{\mathcal{W}} \equiv \{\boldsymbol{w} : [t,T] \to \mathcal{V} | \boldsymbol{v} \text{ measurable}, \mathcal{W} \subset \mathbb{R}^p\}. \tag{A.2}$$

We are concerned with the differential equation,

$$\dot{\boldsymbol{x}}(\tau) = f(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{w}(\tau)), \ \boldsymbol{x}(t) = \boldsymbol{x}, \ T \leq \tau \leq t \tag{A.3}$$

where $f(\tau, \cdot, \cdot, \cdot)$ and $\boldsymbol{x}(\cdot)$ are bounded and Lipschitz continuous. This bounded Lipschitz continuity property assures uniqueness of the system response $\boldsymbol{x}(\cdot)$ to controls $\boldsymbol{u}(\cdot)$ and $\boldsymbol{v}(\cdot)$ [Evans and Souganidis, 1984]. Associated with (A.3) is the payoff functional

$$\boldsymbol{\ell}(\boldsymbol{u}, \boldsymbol{w}) = \boldsymbol{\ell}(t; \boldsymbol{x}, \boldsymbol{u}(\cdot), \boldsymbol{w}(\cdot)) \equiv \int_t^T l(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{w}(\tau)) d\tau + \boldsymbol{g}(\boldsymbol{x}(T)), \tag{A.4}$$

where $\boldsymbol{g}(\cdot) : \mathbb{R}^n \to \mathbb{R}$ satisfies

$$|\boldsymbol{g}(\boldsymbol{x})| \leq k_1, \quad |\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\hat{\boldsymbol{x}}) | \leq k_1 |\boldsymbol{x} - \hat{\boldsymbol{x}}| \tag{A.5}$$

and $l : [0,T] \times \mathbb{R}^n \times \mathcal{U} \times \mathcal{W} \to \mathbb{R}$ is bounded and uniformly continuous, with

$$| l(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) | \leq k_2, \ | l(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) - l(t; \hat{\boldsymbol{x}}, \boldsymbol{u}, \boldsymbol{w}) | \leq k_2 | \boldsymbol{x} - \hat{\boldsymbol{x}} | \tag{A.6}$$

for constants k_1, k_2 and all $0 \leq t \leq T$, $\hat{\boldsymbol{x}}$, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{w} \in \mathcal{W}$. We call T the *terminal time* (it may be infinity!) and the integral, when it does not depend on the control laws, is the *performance index*. The evader's goal is to maximize the payoff (A.4) and pursuer's goal is to minimize it.

A.4 Upper and Lower Values of the Differential Game.

Suppose that the pursuer's mapping strategy (starting at t) is $\beta : \bar{\mathcal{U}}(t) \to \bar{\mathcal{W}}(t)$ provided for each $t \leq \tau \leq T$ and $\boldsymbol{u}, \hat{\boldsymbol{u}} \in \bar{\mathcal{U}}(t)$; then $\boldsymbol{u}(\bar{t}) = \hat{\boldsymbol{u}}(\bar{t})$ a.e. on $t \leq \bar{t} \leq \tau$ implies $\beta[\boldsymbol{u}](\bar{t}) = \beta[\hat{\boldsymbol{u}}](\bar{t})$ a.e. on $t \leq \bar{t} \leq \tau$. The differential game's lower value for a solution $\boldsymbol{x}(t)$ that solves (A.3) for $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t) = \beta[\boldsymbol{u}](\cdot)$ is

$$\boldsymbol{v}^-(t; \boldsymbol{x}) = \inf_{\beta \in \mathcal{B}(t)} \sup_{\boldsymbol{u} \in \mathcal{U}(t)} \boldsymbol{\ell}(\boldsymbol{u}, \beta[\boldsymbol{u}]) \triangleq \inf_{\beta \in \mathcal{B}(t)} \sup_{\boldsymbol{u} \in \mathcal{U}(t)} \int_t^T l(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \beta[\boldsymbol{u}](\tau)) d\tau + \boldsymbol{g}\left(\boldsymbol{x}(T)\right). \tag{A.7}$$

Similarly, suppose that the evader's mapping strategy (starting at t) is $\alpha : \bar{\mathcal{W}}(t) \to \bar{\mathcal{U}}(t)$ provided for each $t \leq \tau \leq T$ and $\boldsymbol{w}, \hat{\boldsymbol{w}} \in \bar{\mathcal{W}}(t)$; then $\boldsymbol{w}(\bar{t}) = \hat{\boldsymbol{w}}(\bar{t})$ a.e. on $t \leq \bar{t} \leq \tau$ implies $\alpha[\boldsymbol{w}](\bar{t}) =$

[2] Cauchy-type HJ equations are time-dependent versions of the HJ PDE.

$\alpha[\hat{\boldsymbol{w}}](\bar{t})$ a.e. on $t \leq \bar{t} \leq \tau$. The differential game's upper value for a solution $\boldsymbol{x}(t)$ that solves (A.3) for $\boldsymbol{u}(t) = \alpha[\boldsymbol{w}](\cdot)$ and $\boldsymbol{w}(t)$ is

$$\boldsymbol{v}^+(t;\boldsymbol{x}) = \sup_{\alpha\in\mathcal{A}(t)} \inf_{\boldsymbol{w}\in\mathcal{W}(t)} \boldsymbol{\ell}(\alpha[\boldsymbol{w}],\boldsymbol{w}) \triangleq \sup_{\alpha\in\mathcal{A}(t)} \inf_{w\in\mathcal{W}(t)} \int_t^T l(\tau,\boldsymbol{x}(\tau),\alpha[\boldsymbol{w}](\tau),\boldsymbol{w}(\tau))d\tau + \boldsymbol{g}\left(\boldsymbol{x}(T)\right). \tag{A.8}$$

These non-local PDEs ((A.7) and (A.8)) are hardly smooth throughout the state space so that they lack classical solutions even for smooth Hamiltonian and boundary conditions. However, these two values are "viscosity" (generalized) solutions [Lions, 1982, Crandall and Lions, 1983] of the associated HJ-Isaacs (HJI) PDE, i.e. solutions which are *locally Lipschitz* in $\Omega \times [0,T]$, and with at most first-order partial derivatives in the Hamiltonian.

A.5 Viscosity Solution of HJ-Isaac's Equations.

For any optimal control problem a value function is constructed based on the optimal cost (or payoff) of any input phase $(\boldsymbol{x}, T)$. In reachability analysis, typically this is defined using a terminal cost function $g(\cdot) : \mathbb{R}^n \to \mathbb{R}$ that satisfies

$$|\boldsymbol{g}(\boldsymbol{x})| \leq k,\ |\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\hat{\boldsymbol{x}})\ | \leq k|\boldsymbol{x} - \hat{\boldsymbol{x}}\ | \tag{A.9}$$

for constant k and all $T \leq t \leq 0$, $\hat{\boldsymbol{x}},\ \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{w} \in \mathcal{W}$. The zero sublevel set of $\boldsymbol{g}(\boldsymbol{x})$ i.e.

$$\mathcal{L}_0 = \{\boldsymbol{x} \in \bar{\Omega}\,|\,\boldsymbol{g}(\boldsymbol{x}) \leq 0\}, \tag{A.10}$$

Lemma A.1. *The lower value $\boldsymbol{v}^-$ in* (A.15) *is the viscosity solution to the lower Isaac's equation*

$$\boldsymbol{v}_t^- + \boldsymbol{H}^-(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},D\boldsymbol{v}^-) = 0,\ t \in [0,T]\ \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{v}^-(T;\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}(T)), \quad \boldsymbol{x} \in \mathbb{R}^m \tag{A.11}$$

with lower Hamiltonian,

$$\boldsymbol{H}^-(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},p) = \max_{u\in\mathcal{U}} \min_{w\in\mathcal{W}} \langle \boldsymbol{f}(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w}),p\rangle. \tag{A.12}$$

where p, the co-state, is the spatial derivative of $\boldsymbol{v}^-$ w.r.t $\boldsymbol{x}$.

Lemma A.2. *The upper value $\boldsymbol{v}^+$ in* (A.8) *is the viscosity solution of the upper Isaac's equation*

$$\boldsymbol{v}_t^+ + \boldsymbol{H}^+(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},D\boldsymbol{v}^+) = 0,\ t \in [0,T]\,,\ \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{v}^+(T;\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}(T)), \quad \boldsymbol{x} \in \mathbb{R}^n \tag{A.13a}$$

with upper Hamiltonian,

$$\boldsymbol{H}^+(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},p) = \min_{\boldsymbol{w}\in\mathcal{W}} \max_{\boldsymbol{u}\in\mathcal{U}} \langle \boldsymbol{f}(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w}),p\rangle, \tag{A.14}$$

with p being appropriately defined.

Corollary A.3. *(i) $\boldsymbol{v}^- \leq \boldsymbol{v}^+$ over $(t \in [0,T]\ \boldsymbol{x} \in \mathbb{R}^n)$ (ii) if for all $t \in [0,T]\,, (\boldsymbol{x},p)\ \in \mathbb{R}^n$, the minimax condition is satisfied i.e. $\boldsymbol{H}^+(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},p) = \boldsymbol{H}^-(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},p)$, then $\boldsymbol{v}^- \equiv \boldsymbol{v}^+$.*

A.6 Reachability for Systems Verification.

Reachability analysis is one of many verification methods that allows us to reason about (control-affine) dynamical systems. The verification problem may consist in finding a *set of reachable states* that lie along the trajectory of the solution to a first order nonlinear partial differential equation that originates from some initial state $\boldsymbol{x}_0 = \boldsymbol{x}(0)$ up to a specified time bound, $t = t_f$. *From a set of initial and unsafe state sets, the time-bounded safety verification problem is to determine if there is an initial state and a particular time within the bound that the solution to HJI PDE enters the unsafe set.*

Reachability could be analyzed in a (i) *forward* sense, whereupon system trajectories are examined to determine if they enter certain states from an *initial set*; (ii) *backward* sense, whereupon system trajectories are examined to determine if they enter certain *target sets*; (iii) *reach set* sense, in which they are examined to see if states reach a set at a *particular time*; or (iv) *reach tube* sense, in which they are evaluated that they reach a set at a point *during a time interval*.

Backward reachability consists in avoiding an unsafe set of states under the worst-possible disturbance at all times; relying on nonanticipative control strategies. Backward reachable sets (BRS) and backward reachable tubes (BRTs) are popularly analyzed in a game of two vehicles with non-stochastic dynamics [Merz, 1972]. Such BRTs possess discontinuity at cross-over points (which exist at edges) on the surface of the tube, and may be non-convex. Therefore, treating the end-point constraints under these discontinuity characterizations need careful consideration and analysis when switching control laws if the underlying PDE does not have continuous partial derivatives (we discuss this further in § 2).

A.7 A Differential Game with Two Inputs

This section provides a cursory background for a minimal problem construction. Readers interested in a more detailed background can consult the thorough treatment in appendix A.

For a (possibly infinite) *terminal time* T, a constant k, controls $\boldsymbol{u} \in \bar{\mathcal{U}}$ (of $\boldsymbol{E}$), $\boldsymbol{w} \in \bar{\mathcal{W}}$ (of $\boldsymbol{P}$), and all $\hat{\boldsymbol{x}},\ \boldsymbol{x} \in \mathbb{R}^n$, let the system possess a bounded and uniformly continuous (BUC) payoff functional $\boldsymbol{g}(\boldsymbol{x}(T))$ that satisfies $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}$. The differential game's *lower value* [Evans and Souganidis, 1984],

$$\boldsymbol{v}(t;\boldsymbol{x}) = \inf_{\beta\in\mathcal{B}(t)} \sup_{\boldsymbol{u}\in\mathcal{U}(t)} \mathcal{J}(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w}) \triangleq \inf_{\beta\in\mathcal{B}(t)} \sup_{\boldsymbol{u}\in\mathcal{U}(t)} \min_{\tau\in[t,T]} \boldsymbol{\ell}(\xi^{\boldsymbol{u},\boldsymbol{w}}_{\boldsymbol{x},t}(\tau)) \tag{A.15}$$

is the viscosity solution to (HJ-IVP) [Evans and Souganidis, 1984, Lemma 2.1] with Hamiltonian,

$$\boldsymbol{H}(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w},p) = \max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{w}\in\mathcal{W}} \langle f(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{w}), p\rangle. \tag{A.16}$$

Here, p is the co-state and $\beta \in \mathcal{B}(t)$ is a *non-anticipative strategy* of the pursuer $\boldsymbol{P}$, i.e. a map $\beta : \bar{\mathcal{U}}(t) \to \bar{\mathcal{W}}(t)$ such that for every $t \le \tau \le T$ and every $\boldsymbol{u}, \hat{\boldsymbol{u}} \in \bar{\mathcal{U}}(t)$, the condition $\boldsymbol{u}(\bar{t}) = \hat{\boldsymbol{u}}(\bar{t})$ a.e. on $[t,\tau)$ implies $\beta[\boldsymbol{u}](\bar{t}) = \beta[\hat{\boldsymbol{u}}](\bar{t})$ a.e. on $[t,\tau)$ [Evans and Souganidis, 1984] (see § A.4 for a detailed treatment). *Note that the $O(N \cdot n)$ scaling applies to the evaluation of this* $\max\min$ *Hamiltonian, which is often the most expensive part of grid-based solvers.*

A.8 Reachability from Differential Games Optimal Control

For any admissible control-disturbance pair $(\boldsymbol{u}(\cdot), \boldsymbol{w}(\cdot))$ and initial phase $(\boldsymbol{x}_0, t_0)$, Crandall [Crandall and Lions, 1983] and Evan's [Evans and Souganidis, 1984] claim is that there exists a unique function

$$\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t; t_0, \boldsymbol{x}_0, \boldsymbol{u}(\cdot), \boldsymbol{w}(\cdot)) \tag{A.17}$$

that satisfies (A.3) a.e. with the property that

$$\boldsymbol{\xi}(t_0) = \boldsymbol{\xi}(t_0; t_0, \boldsymbol{x}_0, \boldsymbol{u}(\cdot), \boldsymbol{w}(\cdot)) = \boldsymbol{x}_0. \tag{A.18}$$

Read (A.17): the motion of (A.3) passing through phase $(\boldsymbol{x}_0, t_0)$ under the action of control $\boldsymbol{u}$, and disturbance $\boldsymbol{w}$, and observed at a time t afterwards. One way to design a system verification problem is compute the reachable set of states that lie along the trajectory (A.17) such that we evade the unsafe sets up to a time e.g. t_f within a given time bound e.g. $[t_0, t_f]$. In this regard, we discard the *cost-to-go*, $l(t; \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{w}(\tau))$ in (A.4), (A.15), or (A.8) and certify safety as resolving the terminal value, $g(\boldsymbol{x}(T))$.

In backward reachability analysis, the lower value of the differential game is used in constructing an analysis of the backward reachable set (or tube). Therefore, we can cast a target set as the time-resolved terminal value $\boldsymbol{v}^-(\boldsymbol{x}, T) = \boldsymbol{g}(\boldsymbol{x}(T))$ so that given a time bound, and an unsafe set of states, the time-bounded safety verification problem consists in certifying that there is no phase within the target set (Target-Set) such that the solution to (A.3) enters the unsafe set. Following the backward reachability formulation of [Mitchell et al., 2005], we say the zero sublevel set of $g(\cdot)$ in (A.11) i.e.

$$\mathcal{L}_0 = \{\boldsymbol{x} \in \bar{\Omega} \,|\, \boldsymbol{g}(\boldsymbol{x}) \le 0\}, \tag{A.19}$$

is the *target set* in the phase space $\Omega \times \mathbb{R}$ for a backward reachability problem (proof in [Mitchell et al., 2005]). This target set can represent the failure set, regions of danger, or obstacles to be avoided

etc in the state space. Note that the target set, $\mathcal{L}_0$, is a closed subset of $\mathbb{R}^n$ and is in the closure of Ω. And the *robustly controlled backward reachable tube* for $\tau \in [-T, 0]$[3] is the closure of the open set

$$\mathcal{L}([\tau, 0], \mathcal{L}_0) = \{\boldsymbol{x} \in \Omega \,|\, \exists\, \beta \in \bar{\mathcal{W}}(t) \,\forall\, \boldsymbol{u} \in \mathcal{U}(t), \exists\, \bar{t} \in [-T, 0], \boldsymbol{\xi}(\bar{t}) \in \mathcal{L}_0\},\ \bar{t} \in [-T, 0]\,. \quad \text{(A.20)}$$

Read: the set of states from which the strategies of $\boldsymbol{P}$, and for all controls of $\boldsymbol{E}$ imply that we reach the target set within the interval $[T, 0]$. More specifically, following Lemma 2 of [Mitchell et al., 2005], the states in the reachable set admit the following properties w.r.t the value function $\boldsymbol{v}$

$$\boldsymbol{x} \in \mathcal{L}_0 \implies \boldsymbol{v}^-(\boldsymbol{x}, t) \leq 0, \quad \boldsymbol{v}^-(\boldsymbol{x}, t) \leq 0 \implies \boldsymbol{x} \in \mathcal{L}_0. \quad \text{(A.21a)}$$

Observe: (a) The goal of the pursuer, or $\boldsymbol{P}$, is to drive the system's trajectories into the unsafe set i.e., $\boldsymbol{P}$ has $\boldsymbol{w}$ at will and aims to minimize the termination time of the game (c.f. (Target-Set)); (b) the evader, or $\boldsymbol{E}$, seeks to avoid the unsafe set i.e., $\boldsymbol{E}$ has controls $\boldsymbol{u}$ at will and seeks to maximize the termination time of the game (c.f. (Target-Set)); (c) $\boldsymbol{E}$ has regular controls, $\boldsymbol{u}$, drawn from a Lebesgue measurable set, $\mathcal{U}$ (c.f. (A.15)). (d) $\boldsymbol{P}$ possesses *nonanticipative strategies* (c.f. (A.15)) i.e. $\beta[\boldsymbol{u}](\cdot)$ such that for any of the ordinary controls, $\boldsymbol{u}(\cdot) \in \mathcal{U}$ of $\boldsymbol{E}$, $\boldsymbol{P}$ knows how to optimally respond to $\boldsymbol{E}$'s inputs. This is a classic reachability problem on the resolution of the infimum-supremum over the *strategies* of $\boldsymbol{P}$ and *controls* of $\boldsymbol{E}$ with the time of capture resolved as an extremum of a cost functional) over a time interval.

The terminal value $\boldsymbol{v}^-(t; \boldsymbol{x})$, that characterizes the target set $\mathcal{L}_0$ is the viscosity solution to the HJI PDE

$$\boldsymbol{v}_t^-(t; \boldsymbol{x}) + \min\{0, \boldsymbol{H}^-(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, D\boldsymbol{v}^-)\} = 0, \quad \boldsymbol{v}^-(0; \boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}), \quad \text{(A.22)}$$

where the vector field $\boldsymbol{v}_{\boldsymbol{x}}^-$ is known in terms of the game's terminal conditions so that the overall game is akin to a two-point boundary-value problem. For ease of readability, we will remove the negative superscript on the lower value and Hamiltonian (A.16).

A.9 Detailed Hamiltonian Derivation for Two-Rockets Game

For the two-rockets pursuit-evasion game in relative coordinates, we derive the simplified Hamiltonian from first principles. The general Hamiltonian for a two-player zero-sum game is

$$\boldsymbol{H}(t; \boldsymbol{x}, p) = -\max_{u_e \in [\underline{u}_e, \bar{u}_e]} \min_{u_p \in [\underline{u}_p, \bar{u}_p]} \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} a_p \cos\theta + u_e x \\ a_p \sin\theta + a_e + u_p x - g \\ u_p - u_e \end{bmatrix}, \quad \text{(A.23)}$$

where p_1, p_2, p_3 are the co-states (spatial derivatives of the value function), and $[\underline{u}_e, \bar{u}_e]$ and $[\underline{u}_p, \bar{u}_p]$ are the evader's and pursuer's control bounds, respectively. The game structure reflects the evader's ability to choose controls first (outer max over u_e) followed by the pursuer's response (inner min over u_p).

Setting symmetric control bounds $\underline{u}_e = \underline{u}_p = -1$ and $\bar{u}_e = \bar{u}_p = +1$, and using the symmetry assumption $a_e = a_p = a$, we optimize over the control variables:

$$\boldsymbol{H} = -\max_{u_e \in [-1,1]} \min_{u_p \in [-1,1]} \left[p_1(a\cos\theta + u_e x) + p_2(a\sin\theta + a + u_p x - g) + p_3(u_p - u_e)\right]. \quad \text{(A.24)}$$

Rearranging by control terms:

$$\boldsymbol{H} = -\max_{u_e \in [-1,1]} \min_{u_p \in [-1,1]} \left[p_1 a\cos\theta + p_2 a(1 + \sin\theta) - p_2 g + (p_1 x + p_3)u_p + (p_2 x - p_3)u_e\right]. \quad \text{(A.25)}$$

Since the pursuer minimizes over u_p, it chooses $u_p = -\text{sign}(p_1 x + p_3)$ to minimize the linear term. Similarly, the evader maximizes over u_e, choosing $u_e = \text{sign}(p_2 x - p_3)$. For bang-bang optimal controls, these become:

$$\min_{u_p}(p_1 x + p_3)u_p = -|p_1 x + p_3|, \quad \text{(A.26)}$$

$$\max_{u_e}(p_2 x - p_3)u_e = |p_2 x - p_3|. \quad \text{(A.27)}$$

Thus the simplified Hamiltonian becomes:

$$\boldsymbol{H}(t; \boldsymbol{x}, p) = -\left[a p_1 \cos\theta + p_2(a + a\sin\theta - g) - |p_1 x + p_3| - |p_2 x - p_3|\right], \quad \text{(A.28)}$$

which, when substituted into the HJ PDE yields the simplified rockets HJI equation used in the numerical experiments (cf. Section 3.3).

[3] The (backward) horizon, $-T$ is negative for $T > 0$.

B HJ PDE Linearization

In this section, we construct the Cole-Hopf-like linearization of the viscous Hamilton-Jacobi (HJ) PDE and propose a sampling machinery we use in building our results. We show that the transformation is *exact* only when the Hamiltonian is quadratic in the co-state i.e. $\boldsymbol{H} = \frac{1}{2}\langle \boldsymbol{p}^\top, \boldsymbol{p}\rangle$, and we derive the precise residual that for general Hamiltonians. We then formulate the quasi-linearization iterative algorithm for computing (HJI-RCBRT-Visc) and (HJI-RCBRAT-Visc). We finish this appendix with the correct Gaussian expectation formulas for the value function and its spatial gradient.

B.1 The viscous HJ Equation's solution

In these proofs that follow, we will express the solutions to the viscous HJ equation as the logarithm of the expectation of Gaussian kernels that parameterize the state space. Let us construct the spatial and time derivatives to the solution to the viscous intial-value-HJ problem (HJ-Visc).

Proof of Proposition 2.1. Recall from (HJ-Visc) that

$$\boldsymbol{v}^\delta_t + \boldsymbol{H}(t;\boldsymbol{x}, D\boldsymbol{v}^\delta) = \frac{\delta}{2}\Delta \boldsymbol{v}^\delta \quad \text{in} \quad \mathbb{R}^n \times (0,T], \quad \boldsymbol{v}^\delta(0;\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) \text{ on } \mathbb{R}^n \times \{t=0\}, \tag{B.1}$$

where $\delta > 0$ is the viscosity parameter, $\boldsymbol{H} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is the Hamiltonian, and $g : \mathbb{R}^n \to \mathbb{R}$ is the terminal/initial cost. Let $\boldsymbol{\omega}^\delta = \boldsymbol{\varphi}(\boldsymbol{v}^\delta(t;\boldsymbol{x}))$, where $\boldsymbol{\varphi} : \mathbb{R}^n \to \mathbb{R}$ is a smooth function. Our goal is to choose $\boldsymbol{\varphi}$ so that $\boldsymbol{\omega}^\delta$ solves a linear partial differential equation. Differentiating, we find that

$$\boldsymbol{\omega}^\delta_t = \boldsymbol{\varphi}'(\boldsymbol{v}^\delta)\boldsymbol{v}^\delta_t \text{ and } \Delta\boldsymbol{\omega}^\delta = \boldsymbol{\varphi}''(\boldsymbol{v}^\delta) \mid D\boldsymbol{v}^\delta \mid^2 + \boldsymbol{\varphi}'(\boldsymbol{v}^\delta)\Delta\boldsymbol{v}^\delta. \tag{B.2}$$

Thus,

$$\boldsymbol{\omega}^\delta_t = \boldsymbol{\varphi}'\boldsymbol{v}^\delta_t \triangleq \boldsymbol{\varphi}'(\boldsymbol{v}^\delta)\left(\frac{\delta}{2}\Delta\boldsymbol{v}^\delta - \boldsymbol{H}(t;\boldsymbol{x}, D\boldsymbol{v}^\delta)\right), \tag{B.3a}$$

$$= \frac{\delta}{2}\Delta\boldsymbol{\omega}^\delta - \left(\frac{\delta}{2}\boldsymbol{\varphi}''(\boldsymbol{v}^\delta) \mid D\boldsymbol{v}^\delta \mid^2 + \boldsymbol{\varphi}'(\boldsymbol{v}^\delta)\boldsymbol{H}(t;\boldsymbol{x}, D\boldsymbol{v}^\delta)\right). \tag{B.3b}$$

Suppose that we choose,

$$\boldsymbol{\varphi}^\delta = \exp\left(\frac{-2}{\delta}\cdot\frac{\boldsymbol{H}^\delta}{|D\boldsymbol{z}|^2}\boldsymbol{z}\right), \tag{B.4}$$

as the solution to the residual $\boldsymbol{R}(t;\boldsymbol{x}) := \frac{\delta}{2}\boldsymbol{\varphi}''(\boldsymbol{v}^\delta) \mid D\boldsymbol{v}^\delta \mid^2 + \boldsymbol{\varphi}'(\boldsymbol{v}^\delta)\boldsymbol{H}^\delta = 0$, it follows that

$$\boldsymbol{\omega}^\delta_t \triangleq \frac{\delta}{2}\Delta\boldsymbol{\omega}^\delta. \tag{B.5}$$

Set

$$\boldsymbol{c}(t;\boldsymbol{x}) = \tfrac{2}{\delta}\cdot\boldsymbol{H}^\delta/|D\boldsymbol{v}^\delta|^2 \tag{B.6}$$

and let $\boldsymbol{c}(t;\boldsymbol{x})$ be locally frozen as a spatially-varying coefficient rather than a functional of $\boldsymbol{v}^\delta$.

Corollary B.1. *Note that $\boldsymbol{R}(t;\boldsymbol{x})$ becomes zero when $\boldsymbol{\varphi}^\delta$ is a constant. This holds when $\boldsymbol{H}^\delta/|D\boldsymbol{v}^\delta|^2$ is independent of the phase $(\boldsymbol{x},t)$. In these situations, $\boldsymbol{H}^\delta = \frac{1}{2}\langle \boldsymbol{p}^\top, \boldsymbol{p}\rangle$, $\boldsymbol{c} = 1/\delta = $ const. thus recovering Proposition 2.1.*

Then, dropping the templated arguments for ease of readability, if $\boldsymbol{v}^\delta$ solves (HJ-Visc), then we have the following approximate local transformation

$$\boldsymbol{\omega}^\delta = \exp\left(-\boldsymbol{c}\boldsymbol{v}^\delta\right) \tag{B.7}$$

solves the initial value problem for the differential equation

$$\boldsymbol{\omega}^\delta_t - \frac{\delta}{2}\Delta\boldsymbol{\omega}^\delta = \boldsymbol{R}(t;\boldsymbol{x}) \equiv 0 \quad \text{in} \quad \Omega\times(0,T], \text{ and } \boldsymbol{\omega}^\delta(0;\boldsymbol{x}) = \exp\left(-\boldsymbol{c}(0;\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\right) \quad \text{on} \quad \Omega\times\{t=0\}, \tag{B.8}$$

where the **residual** is

$$\boldsymbol{R}(t;\boldsymbol{x}) = \left(\frac{\delta}{2}\boldsymbol{\varphi}''(\boldsymbol{v}^\delta) \mid D\boldsymbol{v}^\delta \mid^2 + \boldsymbol{\varphi}'(\boldsymbol{v}^\delta)\boldsymbol{H}(t;\boldsymbol{x},D\boldsymbol{v}^\delta)\right) \equiv 0. \tag{B.9}$$

Equation (B.7) is a heat-like equation, similar to the Cole-Hopf transformation [Evans, 2022] for the Eikonal version of (HJ-Visc)[4].

It follows that the unique bounded fundamental solution of (B.8) is given by the Green's function convolution,

$$\boldsymbol{\omega}^\delta(t;\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi\delta t})^n}\int_\Omega \exp\left(-\frac{1}{2\delta t} \mid \boldsymbol{x}-\boldsymbol{y} \mid^2\right)\exp\left(-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{y})\right)d\boldsymbol{y},\ \ (\boldsymbol{x}\in\Omega, t>0) \tag{B.10a}$$

$$\triangleq \mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[\exp\left(-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{y})\right)\right] := \mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[\exp\left(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y})\right)\right] \tag{B.10b}$$

where (B.10a) is the standard heat kernel convolution and the kernel has been rewritten as a Gaussian density with mean $\boldsymbol{x}$ and variance δt. Note that $\boldsymbol{\omega}^\delta(t;\boldsymbol{x}) = 0$ elsewhere. □

Proof of Lemma 2.2. Going by equation (B.7), we can write

$$\boldsymbol{v}^\delta = -(\delta/2)\frac{|D\boldsymbol{v}^\delta|^2}{\boldsymbol{H}(t;\boldsymbol{x},D\boldsymbol{v}^\delta)}\log\boldsymbol{\omega}^\delta := -\frac{1}{\boldsymbol{c}}\log\boldsymbol{\omega}^\delta, \tag{B.11}$$

so that the unique bounded solution to the initial-value (HJ-Visc) i.e. (5) becomes

$$\boldsymbol{v}^\delta(t;\boldsymbol{x}) = -\frac{1}{\boldsymbol{c}}\cdot\log\left\{\frac{1}{(\sqrt{2\pi\delta t})^n}\int_\Omega \exp\left(-\frac{1}{2\delta t}\cdot|\boldsymbol{x}-\boldsymbol{y}|^2\right)\exp\left(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y})\right)\right\}, \tag{B.12a}$$

$$\triangleq -\frac{1}{\boldsymbol{c}}\cdot\log\left\{\mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[\exp\left(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y})\right)\right]\right\}. \tag{B.12b}$$

Corollary B.2. *For the* (HJI-RCBRT-Visc) *and* (HJI-RCBRAT-Visc)*, $\boldsymbol{g}(\boldsymbol{y})$ the above equation transforms to $\boldsymbol{g}(\boldsymbol{x}+\sqrt{\delta(T-t)}\boldsymbol{y})$. For numerical stability, we use the log-sum-exp identity*

$$\boldsymbol{v}^\delta(t;\boldsymbol{x}) = \frac{-1}{\boldsymbol{c}}\log\frac{1}{N}\sum_{i=1}^N \exp(-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{x}+\sqrt{\delta(T-t)}\boldsymbol{y}_i)) \quad \textit{with}\ \ \boldsymbol{y}_i\sim\mathcal{N}(\boldsymbol{x},\delta t).$$

A fortiori, we have the solution to the viscous HJ equation as the log of the expectation of a Gaussian density with mean $\boldsymbol{x}$ and variance δt. □

B.2 Spatial Gradient of the HJ Payoff

Proof of Lemma 2.3. From (B.10a) observe,

$$D\boldsymbol{\omega}^\delta(t;\boldsymbol{x}) = -\frac{1}{(\sqrt{2\pi\delta t})^n}\int_\Omega \frac{(\boldsymbol{x}-\boldsymbol{y})}{\delta\cdot t}\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{y}|^2}{2\delta t}\right)\exp\left(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y})\right)d\boldsymbol{y}, \tag{B.13a}$$

$$= -\frac{1}{\delta t}\mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[(\boldsymbol{x}-\boldsymbol{y})\exp\left(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y})\right)\right]. \tag{B.13b}$$

Inspecting (B.11), we may write

$$D\boldsymbol{v}^\delta(t;\boldsymbol{x}) = -\frac{1}{\boldsymbol{c}}D[\log\boldsymbol{\omega}^\delta(t;\boldsymbol{x})] = -\frac{1}{\boldsymbol{c}}\frac{D\boldsymbol{\omega}^\delta(t;\boldsymbol{x})}{\boldsymbol{\omega}^\delta(t;\boldsymbol{x})} = \frac{1}{\delta\cdot t\cdot\boldsymbol{c}}\cdot\frac{\mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[(\boldsymbol{x}-\boldsymbol{y})\exp(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y}))\right]}{\mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[\exp(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y}))\right]}, \tag{B.14}$$

$$\triangleq \frac{1}{t\cdot\delta\cdot\boldsymbol{c}}\cdot\left(\boldsymbol{x}-\frac{\mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[\boldsymbol{y}\exp(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y}))\right]}{\mathbb{E}_{\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{x},\delta t)}\left[\exp(-\boldsymbol{c}\cdot\boldsymbol{g}(\boldsymbol{y}))\right]}\right). \tag{B.15}$$

□

[4]For the Eikonal version of (HJ-Visc), we set $\boldsymbol{H}(D\boldsymbol{v}(\boldsymbol{x})) = 0$ in Ω, and $\boldsymbol{v}(0;\boldsymbol{x}) = 0$ on $\partial\Omega$.

Corollary B.3 (Log-sum-exp estimator for the value function)**.** *For* (HJI-RCBRT-Visc) *and* (HJI-RCBRAT-Visc)*, the terminal cost argument transforms to* $\boldsymbol{g}(\boldsymbol{x} + \sqrt{\delta(T-t)}\boldsymbol{y}_i)$*. For numerical stability the* log-sum-exp *identity gives,*

$$\boldsymbol{v}^{\delta}(t;\boldsymbol{x}) = \frac{-1}{\boldsymbol{c}^{(k)}} \log \frac{1}{N} \sum_{i=1}^{N} \exp\Big(-\boldsymbol{c}^{(k)} \boldsymbol{g}\Big(\boldsymbol{x} + \sqrt{\delta(T-t)}\boldsymbol{y}_i\Big)\Big), \tag{B.16}$$

with $\boldsymbol{y}_i \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, I_n)$*.*

Corollary B.4 (Monte Carlo gradient estimator)**.** *With samples* $\boldsymbol{s}_i = \boldsymbol{x} + \sqrt{\delta(T-t)}\boldsymbol{y}_i$*,* $\boldsymbol{y}_i \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, I_n)$*, the importance-weighted estimator for* (7) *is,*

$$D\boldsymbol{v}^{\delta}(t;\boldsymbol{x}) = \frac{1}{t \cdot \delta \cdot \boldsymbol{c}^{(k)}} \left(\boldsymbol{x} - \frac{\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{s}_i \cdot \exp\big(-\boldsymbol{c}^{(k)} \boldsymbol{g}(\boldsymbol{s}_i)\big)}{\frac{1}{N}\sum_{i=1}^{N} \exp\big(-\boldsymbol{c}^{(k)} \boldsymbol{g}(\boldsymbol{s}_i)\big)} \right). \tag{B.17}$$

B.3 Sampling Complexity

Remark B.5 (Remark on Assumption 2.8). Note that item 2 of Assumption 2.8 excludes regions where the value-gradient degenerates, such as flat reachable interiors or singular shocks. Furthermore, under item 5, the contraction constant q may become unbounded in the inviscid limit, i.e. as $\delta \to 0$ and/or $m_0 \to 0$. In practice, numerical stabilization mechanisms such as viscosity regularization, coefficient clipping, or solution "smearing" [Osher and Fedkiw, 2004] may be introduced to maintain stability of the discrete integration scheme.

Proof of Theorem 2.5. Recall that,

$$Z \triangleq \exp(-\boldsymbol{c}\, \boldsymbol{g}(\boldsymbol{\zeta})), \quad \mu \triangleq \mathbb{E}[Z], \quad \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x}) \triangleq -\frac{1}{\boldsymbol{c}} \log \mu. \tag{B.18}$$

For i.i.d. samples $\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_N \sim \mathcal{N}(\boldsymbol{x}, \delta(T-t)I_n)$, $Z_i \triangleq \exp(-\boldsymbol{c}\, \boldsymbol{g}(\boldsymbol{\zeta}_i))$, so that $\bar{Z}_N \triangleq \frac{1}{N}\sum_{i=1}^{N} Z_i$, and $\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) \triangleq -\frac{1}{\boldsymbol{c}} \log \bar{Z}_N$.

Let $a \triangleq e^{-\boldsymbol{c} g_{\max}}$, $b \triangleq e^{-\boldsymbol{c} g_{\min}}$. Since $g_{\min} \le \boldsymbol{g}(\boldsymbol{\zeta}) \le g_{\max}$ and $\boldsymbol{c} > 0$, monotonicity of the exponential implies

$$e^{-\boldsymbol{c} g_{\max}} \le e^{-\boldsymbol{c}\boldsymbol{g}(\boldsymbol{\zeta})} \le e^{-\boldsymbol{c} g_{\min}} \tag{B.19}$$

almost surely. By definition of a and b, this is exactly $a \le Z \le b$ almost surely. Hence each Z_i is bounded in $[a, b]$.

Value residuals, $\bar{Z}_N$. Since $\boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x}) = -\frac{1}{\boldsymbol{c}} \log \mu$ and $\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) = -\frac{1}{\boldsymbol{c}} \log \bar{Z}_N$, we find that

$$\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x}) = -\frac{1}{\boldsymbol{c}} \log\left(\frac{\bar{Z}_N}{\mu}\right). \tag{B.20}$$

Hence, $|\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x})| \ge \varepsilon$ implies that

$$\left|\log\left(\frac{\bar{Z}_N}{\mu}\right)\right| \ge \boldsymbol{c}\varepsilon, \tag{B.21}$$

or that $\bar{Z}_N \ge \mu \exp(\boldsymbol{c}\varepsilon)$ or $\bar{Z}_N \le \mu \exp(-\boldsymbol{c}\varepsilon)$. Hence

$$\mathbb{P}\left(|\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x})| \ge \varepsilon\right) \le \mathbb{P}\left(\bar{Z}_N - \mu \ge \mu(\exp(\boldsymbol{c}\varepsilon) - 1)\right) + \mathbb{P}\left(\bar{Z}_N - \mu \le -\mu(1 - \exp(-\boldsymbol{c}\varepsilon)\right). \tag{B.22}$$

Hoeffding bounds for tails. Because $Z_1, \dots, Z_N$ are i.i.d. and each lies in $[a, b]$, Hoeffding's inequality implies that for every $s > 0$,

$$\mathbb{P}(\bar{Z}_N - \mu \geq s) \leq \exp\left(-\frac{2Ns^2}{(b-a)^2}\right), \qquad \mathbb{P}(\bar{Z}_N - \mu \leq -s) \leq \exp\left(-\frac{2Ns^2}{(b-a)^2}\right). \tag{B.23}$$

Apply the first inequality with $s_+ \triangleq \mu(\exp(\boldsymbol{c}\varepsilon) - 1)$ and the second with $s_- \triangleq \mu(1 - \exp(-\boldsymbol{c}\varepsilon))$. We obtain

$$\mathbb{P}\left(\bar{Z}_N - \mu \geq \mu(\exp(\boldsymbol{c}\varepsilon) - 1)\right) \leq \exp\left(-\frac{2N\mu^2(\exp(\boldsymbol{c}\varepsilon) - 1)^2}{(b-a)^2}\right), \tag{B.24}$$

$$\mathbb{P}\left(\bar{Z}_N - \mu \leq -\mu(1 - \exp(-\boldsymbol{c}\varepsilon))\right) \leq \exp\left(-\frac{2N\mu^2(1 - \exp(-\boldsymbol{c}\varepsilon))^2}{(b-a)^2}\right). \tag{B.25}$$

Union bound and simplification. Combining the previous bounds gives,

$$\mathbb{P}\left(|\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x})| \geq \varepsilon\right) \leq \exp\left(-\frac{2N\mu^2(\exp(\boldsymbol{c}\varepsilon) - 1)^2}{(b-a)^2}\right) + \exp\left(-\frac{2N\mu^2(1 - \exp(-\boldsymbol{c}\varepsilon))^2}{(b-a)^2}\right). \tag{B.26}$$

Since $\exp(\boldsymbol{c}\varepsilon) - 1 \geq 1 - \exp(-\boldsymbol{c}\varepsilon)$ for every $\varepsilon > 0$, the first exponential is no larger than the second. Therefore

$$\mathbb{P}\left(|\hat{\boldsymbol{v}}_{\boldsymbol{c},N}(t;\boldsymbol{x}) - \boldsymbol{v}_{\boldsymbol{c}}(t;\boldsymbol{x})| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2N\mu^2(1 - e^{-\boldsymbol{c}\varepsilon})^2}{(b-a)^2}\right). \tag{B.27}$$

This proves the claim of Theorem 2.5. □

Proof of Corollary 2.7. Since $a \leq Z$ almost surely, taking expectations yields $\mu = \mathbb{E}[Z] \geq a$. Hence it is enough to require

$$2\exp\left(-\frac{2Na^2(1 - e^{-\boldsymbol{c}\varepsilon})^2}{(b-a)^2}\right) \leq \alpha. \tag{B.28}$$

Taking logarithms and solving for N gives

$$N \geq \frac{(b-a)^2}{2a^2(1 - e^{-\boldsymbol{c}\varepsilon})^2} \log\frac{2}{\alpha}, \tag{B.29}$$

which proves the second claim. □

Proof of Theorem 2.9. We proceed in several steps.

Step 1: Lipschitz continuity of the frozen-coefficient solve map Φ. Fix an index m and define

$$\psi_m(s) \triangleq -\frac{1}{s}\log M_m(s), \qquad M_m(s) \triangleq \mathbb{E}_{\boldsymbol{\zeta}\sim\mathcal{N}(\boldsymbol{x}_m, \delta(T-t)I_n)}\left[\exp\left(-s\boldsymbol{g}(\boldsymbol{\zeta})\right)\right]. \tag{B.30}$$

Then $(\Phi(c))_m = \psi_m(c_m)$. Since $M_m(s) > 0$,

$$\psi_m'(s) = \frac{1}{s^2}\log M_m(s) - \frac{1}{s}\frac{M_m'(s)}{M_m(s)}. \tag{B.31}$$

Differentiating under the expectation gives

$$M_m'(s) = \mathbb{E}_{\boldsymbol{\zeta}\sim\mathcal{N}(\boldsymbol{x}_m, \delta(T-t)I_n)}\left[-\boldsymbol{g}(\boldsymbol{\zeta})\exp\left(-s\boldsymbol{g}(\boldsymbol{\zeta})\right)\right]. \tag{B.32}$$

Hence

$$-\frac{M_m'(s)}{M_m(s)} = \frac{\mathbb{E}\left[\boldsymbol{g}(\cdot)e^{-s\boldsymbol{g}(\cdot)}\right]}{\mathbb{E}\left[e^{-s\boldsymbol{g}(\cdot)}\right]}, \tag{B.33}$$

which is a weighted expectation of $\boldsymbol{g}$ which has absolute value of at most G by assumption (i). In addition, $|\boldsymbol{g}| \leq G$ implies

$$e^{-sG} \leq M_m(s) \leq e^{sG}. \tag{B.34}$$

We can therefore write $|\log M_m(s)| \le sG$. Substituting these two bounds into the derivative formula yields

$$|\psi'_m(s)| \le \frac{1}{s^2}(sG) + \frac{1}{s}G = \frac{2G}{s} \le \frac{2G}{c_{\min}} \tag{B.35}$$

for every $s \in [c_{\min}, c_{\max}]$. By the mean-value theorem,

$$|\psi_m(c_m) - \psi_m(\widetilde{c}_m)| \le \frac{2G}{c_{\min}}|c_m - \widetilde{c}_m|. \tag{B.36}$$

Taking the maximum over m, we must have

$$\|\Phi(c) - \Phi(\widetilde{c})\|_\infty \le \frac{2G}{c_{\max}}\|c - \widetilde{c}\|_\infty. \tag{B.37}$$

Step 2: Lipschitz continuity of the coefficient-update map Γ. Fix m and write $p \triangleq p_m(v)$, $q \triangleq p_m(w)$. Then

$$|(\Gamma(v))_m - (\Gamma(w))_m| = \frac{2}{\delta}\left|\frac{\boldsymbol{H}(t;\boldsymbol{x}_m,p)}{|p|^2} - \frac{\boldsymbol{H}(t;\boldsymbol{x}_m,q)}{|q|^2}\right|. \tag{B.38}$$

Add and subtract $\boldsymbol{H}(t;\boldsymbol{x}_m,q)/|p|^2$ to obtain

$$\left|\frac{\boldsymbol{H}(t;\boldsymbol{x}_m,p)}{|p|^2} - \frac{\boldsymbol{H}(t;\boldsymbol{x}_m,q)}{|q|^2}\right| \le \frac{|\boldsymbol{H}(t;\boldsymbol{x}_m,p) - \boldsymbol{H}(t;\boldsymbol{x}_m,q)|}{|p|^2} + |\boldsymbol{H}(t;\boldsymbol{x}_m,q)|\left|\frac{1}{|p|^2} - \frac{1}{|q|^2}\right|. \tag{B.39}$$

By assumption (ii), $|p|, |q| \ge m_0$, so that $|p|^2, |q|^2 \ge m_0^2$. Using assumption (iii), the first term is bounded by

$$\frac{|\boldsymbol{H}(t;\boldsymbol{x}_m,p) - \boldsymbol{H}(t;\boldsymbol{x}_m,q)|}{|p|^2} \le \frac{L_H}{m_0^2}|p-q|. \tag{B.40}$$

For the second term, note that

$$\left|\frac{1}{|p|^2} - \frac{1}{|q|^2}\right| = \frac{\big||q|^2 - |p|^2\big|}{|p|^2|q|^2} \le \frac{\big||q| - |p|\big|(|q| + |p|)}{m_0^4} \le \frac{2P_*}{m_0^4}|p-q|, \tag{B.41}$$

where the final inequality uses $|p|, |q| \le P_*$ from assumption (ii). Using $|\boldsymbol{H}(t;\boldsymbol{x}_m,q)| \le H_*$ from assumption (iii), we conclude that

$$\left|\frac{\boldsymbol{H}(t;\boldsymbol{x}_m,p)}{|p|^2} - \frac{\boldsymbol{H}(t;\boldsymbol{x}_m,q)}{|q|^2}\right| \le \left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right)|p-q|. \tag{B.42}$$

Therefore

$$|(\Gamma(v))_m - (\Gamma(w))_m| \le \frac{2}{\delta}\left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right)|p-q|. \tag{B.43}$$

Finally, by assumption (iv),

$$|p-q| \le \|\mathcal{G}(v) - \mathcal{G}(w)\|_\infty \le L_D\|v-w\|_\infty. \tag{B.44}$$

Taking the maximum over m yields

$$\|\Gamma(v) - \Gamma(w)\|_\infty \le \frac{2L_D}{\delta}\left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right)\|v-w\|_\infty. \tag{B.45}$$

Step 3: $\Lambda = \Phi \circ \Gamma$ is a contraction. Combining (B.37) and (B.45), we obtain

$$\|\Lambda(v) - \Lambda(w)\|_\infty = \|\Phi(\Gamma(v)) - \Phi(\Gamma(w))\|_\infty \le \frac{2G}{c_{\min}}\|\Gamma(v) - \Gamma(w)\|_\infty, \tag{B.46}$$

$$\le \frac{2G}{c_{\min}} \cdot \frac{2L_D}{\delta}\left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right)\|v-w\|_\infty = q\|v-w\|_\infty. \tag{B.47}$$

Assumption (v) states that $q < 1$, hence Λ is a contraction on $\mathcal{A}$.

Step 4: Existence, uniqueness, and linear convergence. Since $\mathcal{A}$ is closed in the Banach space $(V, \|\cdot\|_\infty)$ and invariant under Λ, Banach's fixed-point theorem applies. Therefore there exists a unique $v^* \in \mathcal{A}$ such that $\Lambda(v^*) = v^*$. Moreover, for every $v^{(0)} \in \mathcal{A}$, the sequence generated by $v^{(k+1)} = \Lambda(v^{(k)})$ converges to v^* and satisfies

$$\|v^{(k+1)} - v^*\|_\infty \leq q\|v^{(k)} - v^*\|_\infty \leq q^{k+1}\|v^{(0)} - v^*\|_\infty. \tag{B.48}$$

This proves the first convergence statement.

Step 5: Convergence of the coefficients. Applying (B.45) with $w = v^*$ gives

$$\|\Gamma(v^{(k)}) - \Gamma(v^*)\|_\infty \leq \frac{2L_D}{\delta}\left(\frac{L_H}{m_0^2} + \frac{2H_*P_*}{m_0^4}\right)\|v^{(k)} - v^*\|_\infty. \tag{B.49}$$

Step 6: Residual decay. Since $v^{(k+1)} = \Lambda(v^{(k)})$ and Λ is a contraction,

$$\|v^{(k+1)} - v^{(k)}\|_\infty = \|\Lambda(v^{(k)}) - \Lambda(v^{(k-1)})\|_\infty \leq q\|v^{(k)} - v^{(k-1)}\|_\infty. \tag{B.50}$$

Applying this recursively yields

$$\|v^{(k+1)} - v^{(k)}\|_\infty \leq q^k\|v^{(1)} - v^{(0)}\|_\infty. \tag{B.51}$$

Step 7: A posteriori error estimate. Because $v^{(k)} \to v^*$, we may write

$$v^* - v^{(k)} = \sum_{j=k}^{\infty}(v^{(j+1)} - v^{(j)}). \tag{B.52}$$

Taking sup norms and using the triangle inequality,

$$\|v^* - v^{(k)}\|_\infty \leq \sum_{j=k}^{\infty}\|v^{(j+1)} - v^{(j)}\|_\infty \leq \sum_{j=k}^{\infty} q^{j-k}\|v^{(k+1)} - v^{(k)}\|_\infty \tag{B.53}$$

$$= \frac{1}{1-q}\|v^{(k+1)} - v^{(k)}\|_\infty. \tag{B.54}$$

Using once more that $\|v^{(k+1)} - v^{(k)}\|_\infty \leq q\|v^{(k)} - v^{(k-1)}\|_\infty$, we conclude that

$$\|v^{(k)} - v^*\|_\infty \leq \frac{q}{1-q}\|v^{(k)} - v^{(k-1)}\|_\infty. \tag{B.55}$$

A fortiori, this completes the proof. □

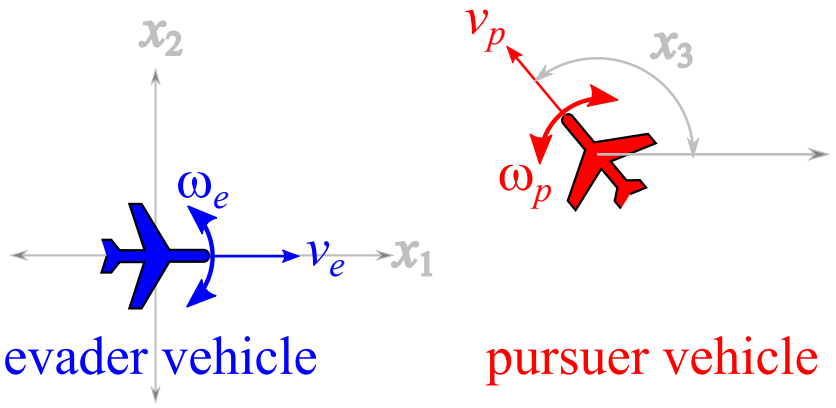

Figure 2: Two Dubins' vehicles in relative Cartesian coordinates. Reprinted from Mitchell [2020].

C Further Numerical Results

This example was originally proposed by Merz [1972] as an iteration upon Isaacs [1999]'s homicidal chauffeur game, whereupon a pursuit-evasion game between two players with similar speeds and minimum turn radii, is thoroughly analyzed. In Mitchell [2020], this problem was established as a benchmark for testing the solubility of capturable set of states (the backward reachable tube) in Merz's classical pursuit-evasion game. In this example, we solve the problem with our LevelSetPy toolbox and establish that the approximated barrier surface to the two-player game conforms with standard results.

The game is that of two cars sharing similar Dubins dynamics Dubins [1957]: $\boldsymbol{P}$ and $\boldsymbol{E}$ both have a positive minimum turn radii, w, and constant speeds v – with motion restricted to a plane as we have for the rocket launch differential game above. In relative coordinates, the diagrammatic structure of the motion is as depicted in Fig. 2. Choosing the Cartesian coordinate for motion representation, the state vector of the game with $\boldsymbol{E}$ at the origin can be characterized by its x_1, x_2 position relative to $\boldsymbol{P}$ and the angle θ between the two vehicles. Capture occurs when the distance $\|\boldsymbol{PE}\|_2$ between the pursuer and the evader becomes less than a specified radius.

The relative equations of motion, going by Fig. 2, is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} -v_e + v_p \cos x_3 + w_e x_2 \\ v_p \sin x_3 - w_e x_1 \\ w_p - w_e \end{pmatrix}. \tag{C.1}$$

We adopt specialization to a case where the two vehicles only possess a unit velocity and unit maximum turn rates. Here, as Merz notes, if the initial velocities are parallel such as $x_3 = 0$, then the equations of relative motion imply that $\boldsymbol{E}$ can be separated from $\boldsymbol{P}$ forever by the initial radial separation if it replicates $\boldsymbol{P}$'s strategy. Whence, the barrier surface is closed and we are presented with Isaacs [1999]'s game of kind where we must determine the nature of the surface. This terminal surface possesses a closed-form solution and we refer readers to the treatment by Merz [1972]. In this example, our chief concern is to judge the efficacy of our toolbox with respect to the analytical solution of the barrier surface.

The backward reachable tube that consists of the paths taken by the trajectories of either player is defined as in the rockets pursuit-evasion game so that we have

$$\ell(0, x) = \{x \in \mathcal{X} | x_1^2 + x_2^2 \leq r^2\}, \tag{C.2}$$

where again r is the capture radius. The target set is a cylinder as ℓ above excludes the heading, x_3. It is represented as shown in Fig. 3.

For a detailed treatment of the barrier surface, we refer readers to a proper analysis as elucidated in Mitchell [2020]. Here, we focus on the construction of the BUP. The set of states that constitute the useable part and its boundary are respectively a function of the implicit surface function representation

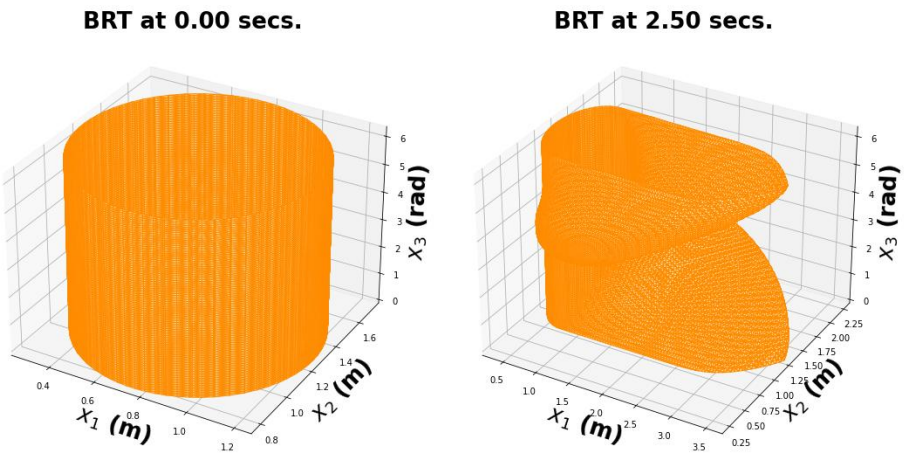

Figure 3: The target set (left) and the boundary of the useable part of the state space after the differential game between $\boldsymbol{P}$ and $\boldsymbol{E}$.

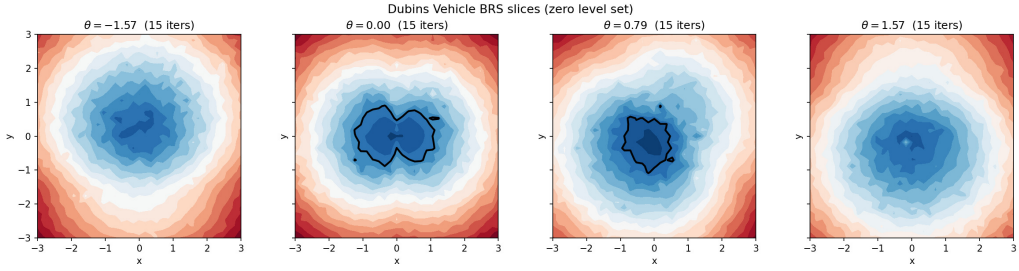

Figure 4: Dubins Vehicle BRT Slices

$\ell : [-T, 0] \times \mathcal{X} \to \mathbb{R}$ so that for a $t \in [0, T]$, where $T > 0$ is

$$\mathcal{T} = \{x \in \mathcal{X} | \ell(0, x) \leq 0\} \tag{C.3}$$

$$R([-t, 0], \mathcal{T}) = \{x \in \mathcal{X} | \ell(t, x) \leq 0\}, \tag{C.4}$$

When $t > 0$, the implicit surface representation is the following HJI PDE

$$\frac{\partial}{\partial t}\ell(t, x) + \min\left(0, \boldsymbol{H}(x, \nabla_x \ell(t, x))\right) = 0. \tag{C.5}$$

It is easy to verify that the Hamiltonian is

$$H(x, p) = p_1(v_p \cos x_3 - v_e) - p_2(v_p \sin x_3) + w|p_1 x_2 - p_2 x_1 - p_3| - w|p_3|. \tag{C.6}$$

Since we are concerned with the special case that the linear and angular speeds are equal, we set $v_e = v_p = w \triangleq +1$ in the foregoing so that the Hamiltonian, in the final analysis is

$$H(x, p) = p_1(\cos x_3 - 1) - p_2(\sin x_3) + |p_1 x_2 - p_2 x_1 - p_3| - |p_3|. \tag{C.7}$$

The results and comparisons are provided in Fig. 4 and Fig. 10.

C.1 The Double Integral Plant

Here, we analyze a time-optimal control problem to determine what admissible control[5] can "transport" the system under consideration to a desired "origin" in the shortest possible time. We consider the double integral plant Bhat and Bernstein [1998], Wonham [1985] as an illustrative example of our

[5]A control law is admissible when its range belongs in the admissible input set where it is bounded.

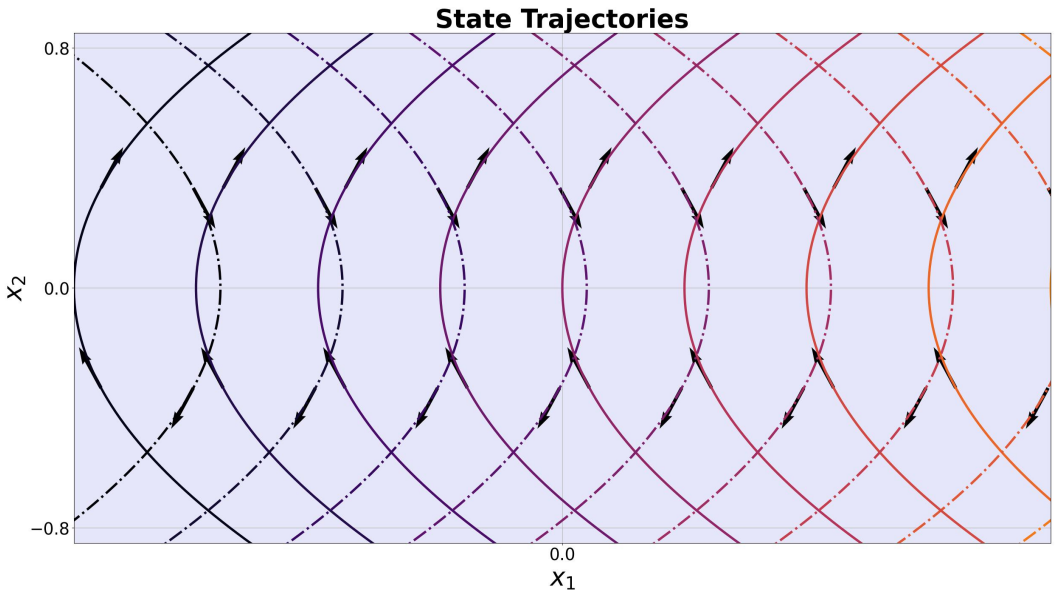

Figure 5: State trajectories of the double integral plant. The solid curves with upward-pointing arrows are trajectories generated under $\boldsymbol{u} = +1$ (switching curve $\boldsymbol{\gamma}_+$); dashed curves with downward-pointing arrows are trajectories under $\boldsymbol{u} = -1$ (switching curve $\boldsymbol{\gamma}_-$).

objective, which is to compute the points in the state space that can reach the origin in *finite-time* under the influence of a time-optimal controller.

We shall leverage standard necessary conditions from the principle of optimality Bellman [1957] to obtain a time-optimal feedback control design; introduce the notion of isochrones and switching surfaces; and discuss the analytic and approximate solutions (with our library) to the time-optimal control problem for a double integrator. We shall conclude the section by comparing the analytic and the overapproximated numerical solution (using the LevelSetPy toolbox) to the time to reach the origin problem.

C.1.1 Dynamics and Problem Setup

The double integrator is controllable, so that open-loop strategies may be employed in driving specific states to the origin in finite time Wonham [1985]. The plant has the following second-order dynamics

$$\ddot{\boldsymbol{x}}(t) = \boldsymbol{u}(t) \tag{C.8}$$

and admits bounded control signals $\mid \boldsymbol{u}(t) \mid \leq 1$ for all time t. After a change of variables,we have the following system of first-order differential equations

$$\begin{aligned} \dot{\boldsymbol{x}}_1(t) &= \boldsymbol{x}_2(t), \\ \dot{\boldsymbol{x}}_2(t) &= \boldsymbol{u}(t), \quad \mid \boldsymbol{u}(t) \mid \leq 1. \end{aligned} \tag{C.9}$$

The *reachability problem that we consider is to address the question of what states can reach a certain point (here, the origin) in a transient manner*. That is, we would like to find point sets on the state space, at a particular time step, such that we can bring the system to the equilibrium, $(0, 0)$.

C.1.2 Time-optimal control scheme

This is an $\boldsymbol{H}$-minimal control problem whereupon we must find the control law that minimizes the Hamiltonian

$$\boldsymbol{H}(\boldsymbol{x}, p) = p_1 \dot{\boldsymbol{x}}_1 + p_2 \dot{\boldsymbol{x}}_2. \tag{C.10}$$

The necessary optimality condition stipulates that the minimizing control law be

$$\boldsymbol{u}(t) = -\text{ sign }(p_2(t)) \triangleq \pm 1. \tag{C.11}$$

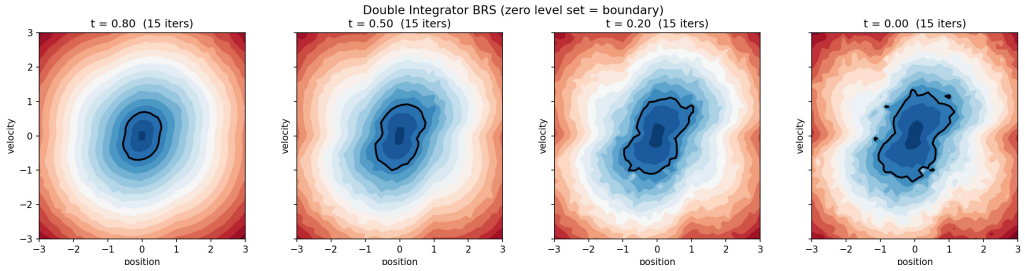

Figure 6: The time to reach the origin as 2D slice comparison for the double integratral plant

Table 3: Double integrator BRS; Algorithm 1 convergence at backward times $t \in \{0.8, 0.5, 0.2, 0.0\}$ with $N = 8{,}000$ and $\delta = 0.08$.

t	Iterations	Final Residual
0.80	15	0.0143
0.50	15	0.0334
0.20	15	0.0596
0.00	15	0.0828

For the co-states in question, suppose that their initial values (for constants k_1 and k_2) are $p_1(t_0) = k_1$ and $p_2(t_0) = k_2$, only four candidates can serve as time-optimal control sequences i.e. $\{[+1], [-1], [+1, -1], [-1, +1]\}$. On a finite time interval, $t \in [t_0, t_f]$, the time-optimal $\boldsymbol{u}(t)$ is a constant $k \equiv \pm 1$ so that for initial conditions $\boldsymbol{x}_1(t_0) = \boldsymbol{\xi}_1$ and $\boldsymbol{x}_2(t_0) = \boldsymbol{\xi}_2$, it can be verified that the state trajectories obey the relation

$$\boldsymbol{x}_1(t) = \boldsymbol{\xi}_1 + \frac{1}{2}k\left(\boldsymbol{x}_2^2 - \boldsymbol{\xi}_2^2\right), \text{ for } t = k\left(\boldsymbol{x}_2(t) - \boldsymbol{\xi}_2\right). \tag{C.12}$$

The trajectories of (C.12) traced out over a finite time horizon $t = [-1, 1]$ with *piecewise constant control laws*, $u = \pm 1$ on a state space and under the control laws $\boldsymbol{u}(t) = \pm 1$ is depicted in Fig. 5. Curves with arrows that point upwards denote trajectories under the control law $\boldsymbol{u} = +1$; call these trajectories $\boldsymbol{\gamma}_+$; while the trajectories marked by dashed arrows pointing downward on the curves were executed under $\boldsymbol{u} = -1$; call these trajectories $\boldsymbol{\gamma}_-$.

Table 3 reports convergence at four backward times; all cases achieve 15 iterations with residuals well below the $O(\sqrt{\delta}) \approx 0.28$ bound.

C.2 The Game of Two rockets on a Plane

We adopt the rocket launch problem of Dreyfus Dreyfus [1966] which is to launch a rocket in fixed time to a desired altitude, given a final vertical velocity component and a maximum final horizontal component as constraints. The rocket's motion is dictated by the following differential equations (under Dreyfus' assumptions)

$$\dot{x}_1 = x_3; \qquad x_1(t_0) = 0; \tag{C.13a}$$
$$\dot{x}_2 = x_4, \qquad x_2(t_0) = 0; \tag{C.13b}$$
$$\dot{x}_3 = a\cos u, \qquad x_3(t_0) = 0; \tag{C.13c}$$
$$\dot{x}_4 = a\sin u - g, \qquad x_4(t_0) = 0; \tag{C.13d}$$

where, (x_1, x_2) are respectively the horizontal and vertical range of the rockets (in feet), (x_3, x_4) are respectively the horizontal and vertical velocities of the rockets (in feet per second), while a and g are respectively the acceleration and gravitational accelerations (in feet per square second). Being a free endpoint problem, we transform it into a game between two players (C.13) without the terminal time constraints as defined in Jacobson and Mayne [1970]. The states of $\boldsymbol{P}$ and $\boldsymbol{E}$ are now denoted as (x_p, x_e) respectively which are driven by their thrusts (u_p, u_e) respectively in the xz-plane (see

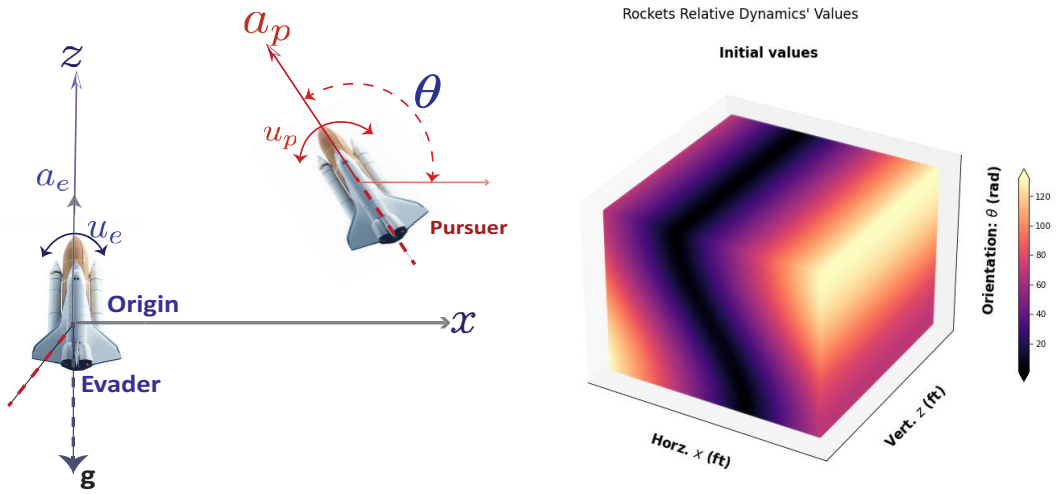

Figure 7: *Left:* Motion of two rockets on a Cartesian $\boldsymbol{xz}$-plane with a thrust inclination in relative coordinates given by $\theta := u_p - u_e$. *Right:* Representation of the initial values as a heat map for the two rockets described in (20).

Figure 7). The relevant kinematic equations are (C.13b) and (C.13d).

$$\dot{x}_{2e} = x_{4e}; \qquad \dot{x}_{2p} = x_{4p}, \tag{C.14a}$$

$$\dot{x}_{4e} = a \sin u_e - g, \qquad \dot{x}_{4p} = a \sin u_p - g \tag{C.14b}$$

where a and g are respectively the acceleration and gravitational accelerations (in feet per square second) i.e. $a = 64 ft/sec^2$ and $g = 32 ft/sec^2$. We reformulate the problem as a two-player differential game where the optimization operations remain in the interior, avoiding discontinuous switches in control. This formulation implicitly bounds each rocket's speed through the gravitational dynamics; the natural speed limit is $a \sin u/g$, which corresponds to each rocket's asymptotic speed when launched vertically.

Therefore, we rewrite (C.13) with $\boldsymbol{P}$'s motion relative to $\boldsymbol{E}$'s along the (x, z) plane so that the relative orientation as shown in Fig. 7 is $\theta = u_p - u_e$. The coordinates of $\boldsymbol{P}$ are freely chosen; however, the coordinates of $\boldsymbol{E}$ are chosen a distance r away from (x, z) so that the $\boldsymbol{EP}$ vector's inclination measured counterclockwise from the $x-$axis is θ. Following the conventions in Fig. 7, the game's relative equations of motion in reduced space is $\mathcal{X} = (x, z, \theta)$ where $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ and $(x, z) \in \mathbb{R}^2$ are

$$\dot{x} = a_p \cos\theta + u_e x, \tag{C.15a}$$

$$\dot{z} = a_p \sin\theta + a_e + u_e x - g, \tag{C.15b}$$

$$\dot{\theta} = u_p - u_e. \tag{C.15c}$$

The payoff, Φ, is the distance of $\boldsymbol{P}$ from $\boldsymbol{E}$ when capture occurs denoted as $\|\boldsymbol{PE}\|_2$. Capture occurs when $\|\boldsymbol{PE}\|_2 \leq r$ for a pre-specified capture radius, $r > 0$. In (**??**), we say $\boldsymbol{P}$ controls u_p and is minimizing Φ, and $\boldsymbol{E}$ controls u_e and is maximizing P. The boundary of the *usable part* of the origin-centered circle of radius r[6] is $\|\boldsymbol{PE}\|_2$ so that

$$r^2 = x^2 + z^2, \tag{C.16a}$$

and all capture points are specified by

$$\dot{r}(x, t) + \min\left[0, \boldsymbol{H}(\boldsymbol{x}, \frac{\partial r(x, t)}{\partial x})\right] \leq 0, \tag{C.17}$$

with the corresponding Hamiltonian

$$\boldsymbol{H}(\boldsymbol{x}, p) = -\max_{u_e \in \mathcal{U}_e} \min_{u_p \in \mathcal{U}_p} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^T \begin{bmatrix} a_p \cos\theta + u_e x \\ a_p \sin\theta + a_e + u_p x - g \\ u_p - u_e \end{bmatrix}. \tag{C.18}$$

[6]We set $r = 1.5 ft$ in our evaluations.

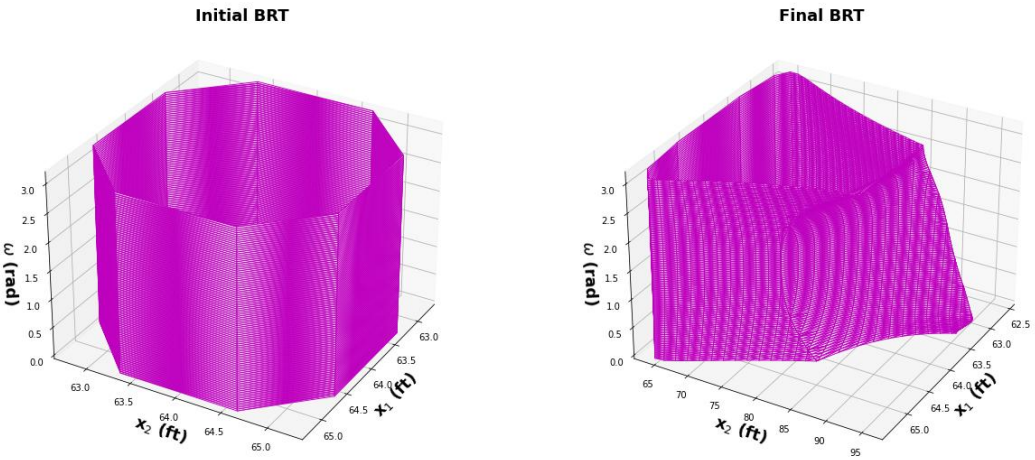

Figure 8: Initial and final backward reachable tubes for the rocket system (cf. Fig. 7) computed using the method outlined in Evans and Souganidis [1984], Osher and Sethian [1988], Mitchell [2004]. We set $a_e = a_p = 64ft/sec^2$ and $g = 32ft/sec^2$ as in Dreyfus' original example. We compute the reachable set by optimizing for the paths of slowest-quickest descent in equation (**??**).

Suppose that the maximizing u_e is $\bar{u}_e$ and the minimizing u_p is $\bar{u}_p$. We have at the point of slowest-quickest descent on the capture surface, that

$$\bar{u}_e = p_1 x - p_3, \tag{C.19a}$$
$$\bar{u}_p = p_3 - p_2 x. \tag{C.19b}$$

We set the linear velocities and accelerations equal to one another i.e. $u_e = u_p$ and $a_e = a_p$. Thus, the Hamiltonian takes the form

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{x}, p) = &- \cos(u)|ap_1| + \cos(u)|ap_1| - \sin(u)|ap_2| - \\ &\sin(u)|ap_2| + u|p_3| - u|p_3|. \end{aligned} \tag{C.20}$$

Using a distributed version of the levelset toolbox Molu [2024a], the backward reachable tube of the game is depicted in Fig. 8. A game between the two players was run over 11 global optimization time steps. The initial value function (left inset of Fig. 8) is represented as a dynamic implicit surface over all point sets in the state space with a signed distance function. We made the third coordinate axis of the state space (here the common heading of the two rockets) to align with the third cylinder axis. The final BRT at the end of the optimization run is shown in the right inset of Fig. 8.

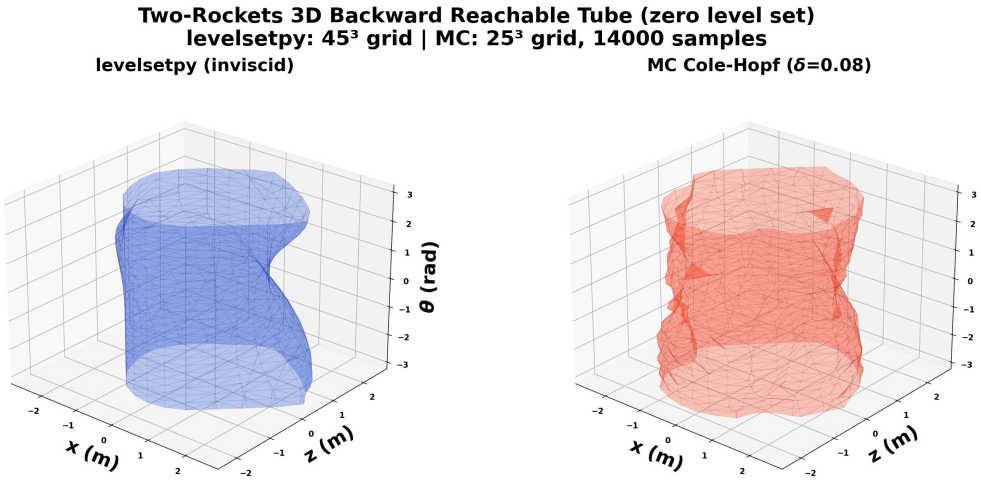

Figure 9: Initial and final backward reachable tubes for the rocket system (cf. Fig. 7) computed using the method outlined in Evans and Souganidis [1984], Osher and Sethian [1988], Mitchell [2004]. We set $a_e = a_p = 64ft/sec^2$ and $g = 32ft/sec^2$ as in Dreyfus' original example. We compute the reachable set by optimizing for the paths of slowest-quickest descent in equation (**??**).

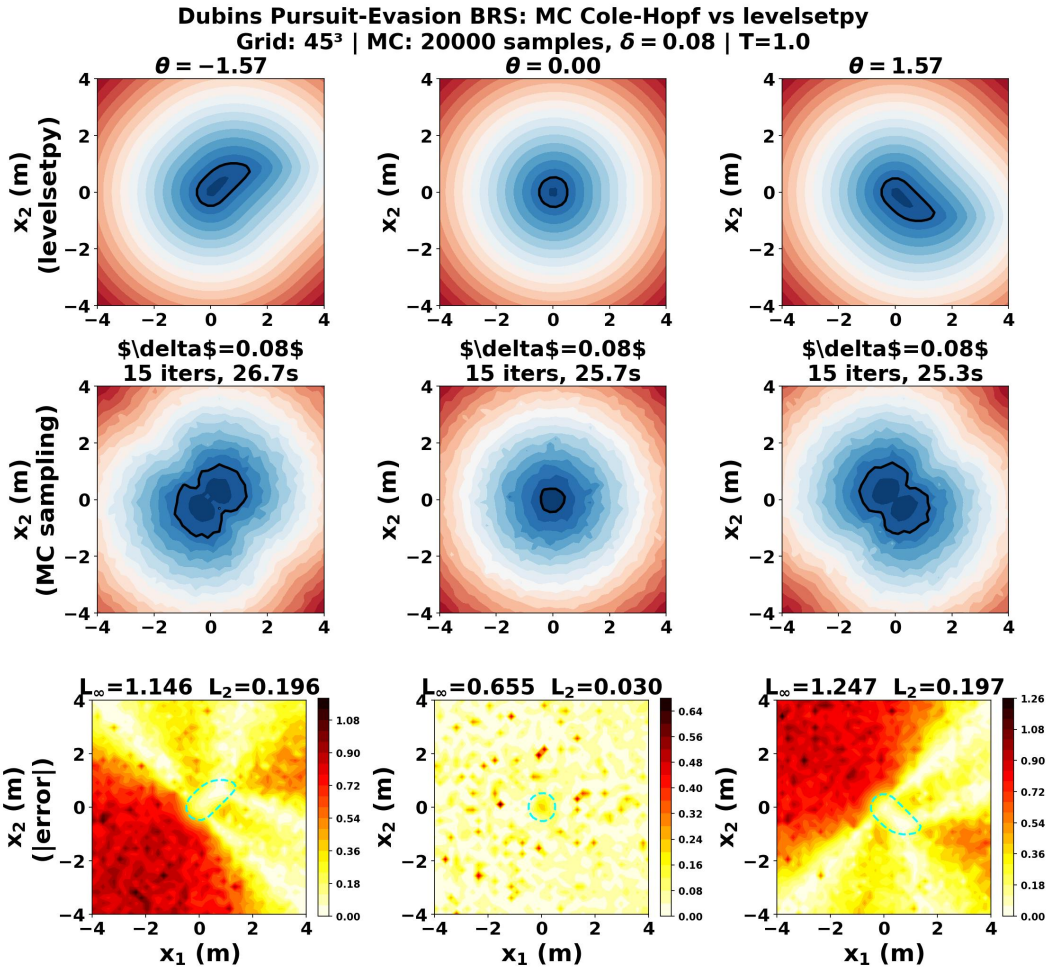

Figure 10: Dubins Vehicles' BRT Slices: Levelsetpy vs Monte-Carlo (Ours). The bottom row shows a heat map of the errors at various relative vehicle orientations (the control input) between the Monte Carlo sampling scheme and the
textttLevelSetPy scheme.