A Second-Order Reachable Sets Computational Scheme for a Cauchy-type Variational Hamilton-Jacobi-Isaacs Equation.

Lekan Molu.

Abstract— **We consider an overapproximation scheme for computing high-dimensional reachable sets. Focusing on the continuous-time systems, we leverage the standard terminal value of the Hamilton-Jacobi-Isaacs (HJI) equation in devising a second-order variational method for computing the reachable sets. We numerically evaluate our methods on mechanical control systems and formal verification domains. Within the bounds here set, our analyses and results suggest that these overapproximated reachable sets computation scheme is applicable to numerous applications ranging from formal verification analysis, invariant sets determination, steady-state/asymptotic stability analysis, safety-critical control analysis, nonconvex robust control analysis, set-based observers and predictions, to state estimation theory.**

I. INTRODUCTION

At issue are *point sets* for dynamical systems' behavioral evolution (or trajectory) throughout a state space. Such sets must satisfy *input or state constraints* in a *least restrictive sense* and under a *worst-possible disturbance* [1]. Because of *decidability issues*[1], we restrict our computation scheme *over a finite time period*. This is a classic reachability theory problem. Because the reachable sets are computed in a finite time horizon, we resort to approximation methods. To allow formal reasoning and guarantees on the system, we require the approximation to contain the reachable set i.e. our scheme must engender an over-approximation guarantee. Our overapproximated sets computation method is applicable to continuous systems automatons in formal verification analysis, invariant sets determination and applications to safety-critical controllers, nonconvex robust control analysis, set-based observers and predictions, as well as state estimation theory.

In emphasis, we present an overapproximation scheme for constructing the *discriminating kernel* for a reachable sets computational problem. Ours is an extension of differential dynamic programming methods in Bolza-like objective functions inspired by [2], [3], and [4]'s second-order variational methods. In emphasis, we focus on pursuit-evasion games [5], where in an iterative dynamic game fashion [6], each agent's strategy depends on its opponent's control law as agents locally approximate successive trajectories' *values* that emanate from a state space. Our evaluation is on a variant of Isaacs' [5] *dolichobrachistochrone* optimization problem where two players are tasked with finding opposing extremals for the time of quickest or slowest descent on a *capture surface*.

Reachable sets can be analyzed in a (i) *forward* sense, where system trajectories are examined to determine if they enter certain states from an *initial set*; (ii) *backward* sense, where 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 examined to determine that they reach a set *during a time interval*.

We focus on the backward construction. This consists in avoiding an unsafe state set under the worst-possible disturbance within a given time-bound. The state sets of a backward reachable problem constitute the *discriminating kernel* that is "safety-preserving" for

Microsoft Research, 300 Lafayette Street, New York, NY10012, USA.
`lekanmolu@microsoft.com`

[1]We say a reachability problem is decidable if we can compute all the states that can be reached from an initial condition in a finite number of steps.

a finite-time horizon control problem. This discriminating kernel's boundary satisfies a generalized Isaac's equation i.e. if the state $\boldsymbol{x}$ is within the set's boundary, but not the target's boundary, then for a co-state p that is an exterior proximal normal at $\boldsymbol{x}$, $\boldsymbol{H}(\boldsymbol{x}, p) \leq 0$, where $\boldsymbol{H}(\boldsymbol{x}, p)$ is the problem's Hamiltonian. The set associated with the kernel computed in a backward reachability framework is termed the *backward reachable set* or *BRS*.

Eulerian methods [7], [8] resolve the BRS as the zero-level set of an implicitly-defined value function on a state space. Constructed as an initial value problem for a *Cauchy-type* Hamilton-Jacobi-Isaacs (HJI) partial differential equation (P.D.E.) [9], [10], the BRS in one dimension is equivalent to the conservation of momentum equation [8]. For a multidimensional problem, by resolving the P.D.E. on a dimension-by-dimension basis in a consistent and monotone fashion [11], a numerically precise and accurate solution to the HJ P.D.E can be determined [7], [8]. However, as state dimensions increase, spacetime discretization methods become impractical owing to their exponential complexity.

Contrary to Eulerian methods, our approximation method does not require state space discretization and hence reduces the exponential complexity to polynomial time. While reachable sets computed as the zero-sublevel sets of the terminal HJI value function in a two-player game using Lax-Friedrichs schemes may be interpreted to contain the Taylor model we introduce, their computational cost increases exponentially with the state dimensions.

Contributions: We describe a computational scheme, provide a summary of the complexity and convergence behavior for the overapproximated BRS using [9]'s standard variational Hamilton-Jacobi equation. Iteratively approximating nonlinear trajectories about "near" local paths and under positive-definiteness requirement of the stagewise cost function's second-derivatives (in control terms), we compute the extrema of the nominal state and control-disturbance policy pair's cost to find a *cost improvement* per iteration. Trajectories are updated until we sweep all possible initial conditions into the space of *all trajectory costs*. Within the bounds here set, the zero isocontour of all "unionized costs" of all trajectories swept along paths that assure cost improvement per iteration of the algorithm constitute the zero level set. This work is the first to systematically provide an polynomial time complexity computational scheme for backward reachable sets to our knowledge.

The body of this letter is structured as follows: § II describes the notations used . In § III, we introduce methods that will enable us formulate our proposal in § IV. We describe the computational and complexity analysis in § V.

II. NOTATIONS, TERMINOLOGY, AND ASSUMPTIONS

We employ standard vector-matrix notations throughout. Conventions: lower-case Latin and Greek letters are scalars; in bold-font they are vectors. Upper-case calligraphic letters are sets. Exceptions: Players in a differential game e.g. $\boldsymbol{P}, \boldsymbol{E}$, are individual entities. All vectors are column-stacked. Capital Greek and Calligraphic letters are sets. The scalar product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is written as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. Arbitrary real variables e.g., t, t_0, t_f, τ, T denote *time*. For a scalar function $\boldsymbol{V}$, $\boldsymbol{V_x}$ is the gradient vector, and $\boldsymbol{V_{xx}}$ is the jacobian matrix.

Differential Games: We analyze conflicting objectives between a pursuer ($\boldsymbol{P}$) and an evader ($\boldsymbol{E}$) in a standard pursuit-evasion game. The set of all *strategies* executed by $\boldsymbol{P}$ (resp. $\boldsymbol{E}$) during a game (beginning at a time t) is denoted as $\mathscr{B}(t)$ (resp. $\mathscr{A}(t)$).

System Description: For the dynamical system

$$\dot{\boldsymbol{x}}(t) = f(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)), \; \boldsymbol{x}(0) = \boldsymbol{x}_0, \; -T \le t \le 0, \qquad (1)$$

where the state $\boldsymbol{x}(t)$ evolves from some initial negative time $-T$ to a final time 0, the flow field $f(t, \cdot, \cdot, \cdot)$ and $\boldsymbol{x}(t)$ are bounded and Lipschitz continuous for fixed controls $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$ with continuous second derivatives[2]. We take $\boldsymbol{x}(t)$ to belong in the open set $\Omega \subset \mathbb{R}^n$ and the pair $(\boldsymbol{x}, t)$ as the system's *phase*. The Cartesian product of Ω and the space $T = \mathbb{R}^1$ of all time values is termed the *phase space*, $\Omega \times T$. The closure of Ω is denoted $\bar{\Omega}$ and we let $\delta\Omega$ denote the boundary of Ω. Unless otherwise stated, vectors $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ and $\boldsymbol{v}(t) \in \mathbb{R}^{n_v}$ are reserved for admissible control (resp. disturbance) at time t for some $n_u, n_v > 0$. Controls $\boldsymbol{u}(t)$ (resp. $\boldsymbol{v}(t)$) are piecewise continuous in t, if for each t, $\boldsymbol{u} \in \mathscr{U}$ (resp. $\boldsymbol{v} \in \mathscr{V}$), $\mathscr{U}$ (resp. $\mathscr{V}$) is a Lebesgue measurable and compact set. At all times, any of $\boldsymbol{u}(t)$ or $\boldsymbol{v}(t)$ will be under the influence of a *player* such that the motion of $\boldsymbol{x}(t)$ will be influenced by the will of that player. When a control law or value function is optimal, it shall be signified by an asterisk superscript e.g. $\boldsymbol{u}^\star$. For the phase space $(\Omega \times T)$, the set of all controls for players $\boldsymbol{P}$ and $\boldsymbol{E}$ are respectively drawn from

$$\bar{\mathscr{U}} \equiv \{\boldsymbol{u} : [-T, 0] \to \mathscr{U} \,|\, \boldsymbol{u} \text{ measurable}, \mathscr{U} \subset \mathbb{R}^{n_u}\}, \qquad (2)$$

$$\bar{\mathscr{V}} \equiv \{\boldsymbol{v} : [-T, 0] \to \mathscr{V} \,|\, \boldsymbol{v} \text{ measurable}, \mathscr{V} \subset \mathbb{R}^{n_v}\}, \qquad (3)$$

with $\mathscr{U}$, $\mathscr{V}$ being compact.

Existence and Uniqueness of Value and Trajectories: For any admissible control-disturbance pair $(\boldsymbol{u}(t), \boldsymbol{v}(t))$ and initial phase $(\boldsymbol{x}, -T)$, given a pursuer's strategy, the game admits a value [13] and there exists a unique trajectory $\boldsymbol{\xi}(t)$ such that the motion of (1) passing through phase $(\boldsymbol{x}, -T)$ under the action of control $\boldsymbol{u}(t)$, and disturbance $\boldsymbol{v}(t)$, and observed at a time t afterwards, given by,

$$\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t; -T, \boldsymbol{x}, \boldsymbol{u}(\cdot), \boldsymbol{v}(\cdot)) \qquad (4)$$

satisfies (1) almost everywhere (a.e.) [9]. Note that under feedback strategies, the game may not admit a value necessarily [14]; however, if the target sets are initialized such that there is no overlap of states between players, and that the evader(s) are not captured by the pursuer(s) at the start of the game, then one can guarantee a pursuit winning strategy [15, Assumption 2.1].

III. BACKGROUND.

Reachable sets in the context of dynamic programming and two person games is here introduced. We restrict attention to computing the backward reachable sets of a dynamical system. We establish the viscosity solution P.D.E. to the terminal HJI P.D.E., and then describe the formulation of the BRS and backward reachable tube (BRT). Let us enquire.

A. The Backward Reachable (Target) Set

For any optimal control problem, a value function is constructed based on a user-defined optimal cost that is bounded and uniformly continuous for any input phase $(\boldsymbol{x}, -T)$ e.g. reach goal at the end of a time horizon i.e.,

$$|g(0; \boldsymbol{x})| \le k, \; |g(0; \boldsymbol{x}) - g(t; \hat{\boldsymbol{x}})| \le k|\boldsymbol{x} - \hat{\boldsymbol{x}}| \qquad (5)$$

for constant k and all $-T \le t \le 0$, $\hat{\boldsymbol{x}}, \boldsymbol{x} \in \mathbb{R}^n$. The set

$$\mathscr{L}_0 = \{\boldsymbol{x} \in \bar{\Omega} \,|\, g(0; \boldsymbol{x}) \le 0\}, \qquad (6)$$

is the *target set* in the phase space $\Omega \times \mathbb{R}$ (proof in [9]). This target set can represent the failure set (to avoid) or a goal set (to reach) in the state space.

B. The Backward Reachable Tube

Backward reachability analysis seeks to capture all conditions under which trajectories of the system may enter a user-defined target set cf. (6). This could be desirable in goal-regions of the state (safe sets) or undesirable state configurations (unsafe sets). For a target set construction problem, a differential game's (lower) value is equivalent to a solution of (1) for $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t) = \beta[\boldsymbol{u}](t)$ i.e.,

$$\boldsymbol{V}(\boldsymbol{x}, t) = \inf_{\beta \in \mathscr{B}(t)} \sup_{\boldsymbol{u} \in \mathscr{U}(t)} \min_{t \in [-T, 0]} g(0; t, \boldsymbol{x}, \boldsymbol{u}(\cdot), \boldsymbol{v}(\cdot)). \qquad (7)$$

Optimal trajectories emanating from an initial phase $(\boldsymbol{x}, -T)$ where the value function is non-negative will maintain non-negative cost over an entire time horizon, thereby avoiding the target set and vice versa. For the safety problem setup in (7), we can define the corresponding *robustly controlled backward reachable tube* as the closure of the open set

$$\mathscr{L}([\tau, 0], \mathscr{L}_0) = \{\boldsymbol{x} \in \Omega \,|\, \exists \beta \in \bar{\mathscr{V}}(t) \,\forall \boldsymbol{u} \in \mathscr{U}(t), \exists \tau \in [-T, 0],$$
$$\boldsymbol{\xi}(\bar{t}) \in \mathscr{L}_0\}. \qquad (8)$$

Read: The set of states from which the strategies of $\boldsymbol{P}$ and for all controls of $\boldsymbol{E}$ imply that we *reach and remain in the target set* in the interval $[-T, 0]$. Following Lemma 2 of [9], the states in the reachable set admit the following properties w.r.t the value function $\boldsymbol{V}$:

$$\boldsymbol{x}(t) \in \mathscr{L}(\cdot) \implies \boldsymbol{V}(\boldsymbol{x}, t) \le 0, \boldsymbol{V}(\boldsymbol{x}, t) \le 0 \implies \boldsymbol{x}(t) \in \mathscr{L}(\cdot). \qquad (9)$$

Player $\boldsymbol{P}$ is minimizing (the game's termination time c.f. (6)), seeking to drive system trajectories into the unsafe set; and $\boldsymbol{E}$ is maximizing (the game's termination time) i.e. is seeking to avoid the unsafe set-[3].

C. The Terminal HJI Value Function for Reachability

The *non-anticipative strategy* used in level set methods follows from [12] viz., suppose that the pursuer's strategy starting at a time $-T$, is $\beta : \bar{\mathscr{U}} \to \bar{\mathscr{V}}$. Suppose further that β is provided for each $-T \le \tau \le 0$ and $\boldsymbol{u}, \hat{\boldsymbol{u}} \in \bar{\mathscr{U}}$. Then,

$$\begin{cases} \boldsymbol{u}(\bar{t}) = \hat{\boldsymbol{u}}(\bar{t}) \text{ a.e. on } -T \le \bar{t} \le \tau, \\ \text{implies } \beta[\boldsymbol{u}](\bar{t}) = \beta[\hat{\boldsymbol{u}}](\bar{t}) \text{ a.e. on } -T \le \bar{t} \le \tau. \end{cases} \qquad (10)$$

In our work, however, we use feedback strategies in an iterative dynamic game framework while initializing the target sets to ensure the existence of a value as discussed in §II. It is well-known that a differential game's value function admits a "viscosity" (generalized) solution [16], [17] for the associated HJ-Isaacs (HJI) PDE

$$\frac{\partial \boldsymbol{V}}{\partial t}(\boldsymbol{x}, t) + \min\{0, \boldsymbol{H}(t; \boldsymbol{x}, p)\} = 0, \; V(\boldsymbol{x}, 0) = g(\boldsymbol{x}) \qquad (11)$$

where the vector field $p \, (\equiv \boldsymbol{V}_{\boldsymbol{x}})$ is known in terms of the game's terminal conditions so that the overall game is akin to a two-point boundary-value problem; and the Hamiltonian $\boldsymbol{H}(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{V}_{\boldsymbol{x}})$ is defined as

$$\boldsymbol{H}(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{V}_{\boldsymbol{x}}) = \max_{u \in \mathscr{U}} \min_{v \in \mathscr{V}} \langle f(t; \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}), \boldsymbol{V}_{\boldsymbol{x}} \rangle. \qquad (12)$$

---

[2]This bounded Lipschitz continuity property assures uniqueness of the system response $\boldsymbol{x}(t)$ to controls $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$ [12].

[3]For the goal-satisfaction (or *liveness*) problem setups, the strategies are reversed and the backward reachable tube are the states from which the evader $\boldsymbol{E}$ can successfully reach the target set despite worst-case efforts of the pursuer $\boldsymbol{P}$.

Equation (7) only allows the game to determine that a trajectory belongs in the target set at exactly time zero. That is, the evader can chase a trajectory into the target set and escape it before reaching the final time, 0. The min operation between the scalar 0 and the Hamiltonian allows the pursuer to "freeze" trajectories that may be under the evader's willpower when the evader tries to evolve such trajectories outside of the target set. For more details on the construction of this P.D.E., see [9].

IV. SUCCESSIVE APPROXIMATION SCHEME

Throughout this section, all feasible trajectories $\{x_i, x_{i+1}, \cdots, x_n\} \in \Omega$ are scheduled for iterative second-order expansion along the nonlinear state variations δx_i and about nominal states $\bar{x}_i$; we derive the variational linear differential equation form of (11) and describe the integration scheme of the backward and forward steps of standard DDP on the variational HJI equation. In what follows, for ease of notation we drop the subscripts that denote the respective trajectories. For clarity's sake, we shall drop the subscripts and derive the optimal value for a generic trajectory x.

A. Local Approximations to Nonlinear Trajectories

Suppose that system (1) is controllable everywhere on an interval $[-T, 0]$ along a local trajectory x_r generated by admissible control disturbance pair u_r, v_r starting from an initial phase $(x_r, -T)$ so that

$$\dot{x}_r(\tau) = f(t; x_r(\tau), u_r(\tau), v_r(\tau)). \tag{13}$$

First, we apply local controls $u_r(t)$ and $v_r(t)$ on (1) so that the nominal value is $V(t; \cdot, \cdot, \cdot)$ for a resulting nominal state $x_r(\tau); \tau \in [-T, 0]$. System dynamics that describe variations from the nonlinear system cf. (1) with state and control pairs $\delta x(t)$, $\delta u(t)$, $\delta v(t)$ respectively[4] can then be written as

$$x(t) = x_r(t) + \delta x(t), \quad u(t) = u_r(t) + \delta u(t), \tag{14a}$$
$$v(t) = v_r(t) + \delta v(t), \quad t \in [-T, 0]. \tag{14b}$$

Abusing notation, we drop the templated time arguments in the variations (14) so that the canonical problem is now

$$\frac{d}{dt}(x_r + \delta x) = f(t; x_r + \delta x, u_r + \delta u, v_r + \delta v), \tag{15}$$

$$x_r(-T) + \delta x(-T) = x(-T), \tag{16}$$

whose terminal value admits the optimal form (see (11)):

$$-\frac{\partial V^*}{\partial t}(x_r + \delta x, t) = \min \left\{ 0, \max_{\delta u \in \mathcal{U}} \min_{\delta v \in \mathcal{V}} \langle f(t; x_r + \delta x, \right.$$
$$\left. u_r + \delta u, v_r + \delta v), \frac{\partial V^*}{\partial x}(x_r + \delta x, t) \rangle \right\},$$

$$V^*(x_r + \delta x, 0) = g(0; x_r(0) + \delta x(0)); \tag{17}$$

and state trajectory

$$\xi(t) = \xi(t; -T, x_r + \delta x, u + \delta u, v + \delta v), \quad t \in [-T, 0]. \tag{18}$$

For the inequality $-T < t \leq 0$ and a $\tau \in [t, 0]$, let the *optimal cost* for using the optimal control $u^\star(\tau) = u_r(\tau) + \delta u^\star(\tau)$ and disturbance $v^\star(\tau) = v_r(\tau) + \delta v^\star(\tau)$ be $V^\star(x_r, \tau)$. In addition, let the *nominal cost* for using the pair $(u_r(\tau), v_r(\tau))$ be $V_r(x_r, t)$, so that on the phase (x_r, t), we have

$$\tilde{V}^\star(x_r, t) = V^\star(x_r, t) - V_r(x_r, t). \tag{19}$$

Next, we expand V^* in (17) under sufficient regularity assumptions.

B. Power Series Expansion Scheme

Suppose that the optimal terminal cost, $V^*(\cdot)$, is sufficiently smooth to allow a power series expansion in the state variation δx about the nominal state, x_r, then we must have

$$V^\star(x_r + \delta x, t) = \tilde{V}^\star(x_r, t) + V_r(x_r, t) + \langle V_x^*(x_r, t), \delta x \rangle$$
$$+ \frac{1}{2} \langle \delta x, V_{xx}^\star(x_r, t) \delta x \rangle + \text{h.o.t in } \delta x, \tag{20}$$

where h.o.t. signifies higher order terms. The smoothness assumption is necessary for admitting the linear differential equation expansion of (20). This expansion is consistent with **differential dynamic programming** schemes [2], [4], [18].

Observe: If δx is not constrained to be small, (20) may require huge memory for storage owing to the large dimensionality of terms beyond second order. However, (i) if x_r is constrained to be sufficiently close to x, the state variation δx will be small, resulting in $x \approx x_r$ cf. (14)[5]; (ii) for small δx, the expansion of (20) becomes consistent with second-order methods where quadratic terms in δx dominate higher-order terms. Hence, we can avoid infinite data storage requirement by truncating the expansion in (20) at second-order terms in δx. This will incur an $O(\delta x^3)$ approximation error, affording us realizable control laws that can be executed on the system (1). Thus, we rewrite (20) as

$$V^\star(x_r + \delta x, t) = \tilde{V}^\star(x_r, t) + V_r(x_r, t) + \langle V_x^*(x_r, t), \delta x \rangle$$
$$+ \frac{1}{2} \langle \delta x, V_{xx}^\star(x_r, t) \delta x \rangle. \tag{21}$$

Denote by $V_x^\star(x_r + \delta x, t)$ the optimal value of the co-state on the r.h.s of (17). Then, expanding up to second order we find that

$$V_x^\star(x_r + \delta x, t) = V_x^\star(x_r, t) + \langle V_{xx}^\star(x_r, t), \delta x \rangle, \tag{22}$$

where we have again omitted the quadratic term $V_{xxx}^\star \delta x \delta x$ owing to the foregoing reason. Substituting (21) and (22) into (17), abusing notation by dropping the templated phase arguments, we find that

$$-\frac{\partial \tilde{V}^*}{\partial t} - \frac{\partial V_r}{\partial t} - \left\langle \frac{\partial V_x^*}{\partial t}, \delta x \right\rangle - \frac{1}{2} \left\langle \delta x, \frac{\partial V_{xx}^*}{\partial t} \delta x \right\rangle =$$
$$\min \left\{ 0, \max_{\delta u \in \mathcal{U}} \min_{\delta v \in \mathcal{V}} \left\langle f^T(t; x_r + \delta x, u_r + \delta u, v_r + \delta v), \right.\right. \tag{23}$$
$$\left.\left. V_x^* + V_{xx}^* \delta x \right\rangle \right\}.$$

From (19), we may write

$$\frac{d}{dt}(V_r + \tilde{V}^*) = \frac{\partial}{\partial t}(V_r + \tilde{V}^*) + \left\langle f^T(t; x_r, u_r, v_r), V_x^* \right\rangle \tag{24a}$$

$$\dot{V}_x^* = \frac{\partial V_{xx}^*}{\partial t} + \langle f^T(t; x_r, u_r, v_r), V_{xx}^* \rangle, \quad \dot{V}_{xx}^* = \frac{\partial V_{xx}^*}{\partial t}. \tag{24b}$$

Therefore, (23) in light of (24) becomes

$$-\frac{\partial \tilde{V}^*}{\partial t} - \frac{\partial V_r}{\partial t} - \left\langle \frac{\partial V_x^*}{\partial t}, \delta x \right\rangle - \frac{1}{2} \left\langle \delta x, \frac{\partial V_{xx}^*}{\partial t} \delta x \right\rangle =$$
$$\min \left\{ 0, \max_{\delta u \in \mathcal{U}} \min_{\delta v \in \mathcal{V}} [H(t; x_r + \delta x, u_r + \delta u, v_r + \delta v, V_x^*) + \right. \tag{25}$$
$$\left. \langle V_{xx}^* \delta x, f(t; x_r + \delta x, u_r + \delta u, v_r + \delta v) \rangle] \right\}$$

where $H(t; x, u, v, V_x^*) = \langle V_x^*, f(t; x, u, v) \rangle$.

C. Variational HJI Linear Differential Equation

Let the respective maximizing and minimizing control-disturbance pair when $x = x_r$ at time t be

$$u^* = u_r + \delta u^*, \quad v^* = v_r + \delta v^*. \tag{26}$$

Given the variation δx on the r.h.s. of (25), let us write the respective maximizing evader's and minimizing pursuer's controls for state $x = x_r + \delta x$ as

$$u = u_r + \delta u^* + \delta u, \quad v = v_r + \delta v^* + \delta v. \quad (27)$$

Observe that δu and δv are yet to be determined. Henceforward, we adopt the abbreviations $H = H(t; x_r + \delta x, u_r + \delta u, v_r + \delta v, V_x^*)$ and $f = f(t; x_r + \delta x, u_r + \delta u, v_r + \delta v)$. Expanding the r.h.s. of (25) about x_r, u^*, v^* at time t up to second-order as before, we have

$$\min \left\{ 0, \max_{\delta u \in \mathcal{U}} \min_{\delta v \in \mathcal{V}} [H + \langle H_x + V_{xx}^* f, \delta x \rangle + \langle H_u, \delta u \rangle + \right.$$
$$\langle H_v, \delta v \rangle + \langle \delta u, (H_{ux} + f_u^T V_{xx}^*) \delta x \rangle + \left\langle \delta v, (H_{vx} + f_v^T V_{xx}^*) \delta x \right\rangle$$
$$+ \frac{1}{2} \left\langle \delta u, H_{uv} \delta v + \frac{1}{2} \langle \delta v, H_{vu} \delta u \rangle \right\rangle + \frac{1}{2} \langle \delta u, H_{uu} \delta u \rangle$$
$$\left. + \frac{1}{2} \langle \delta v, H_{vv} \delta v \rangle + \frac{1}{2} \left\langle \delta x, \left(H_{xx} + f_x^T V_{xx}^* + V_{xx}^* f_x \right) \delta x \right\rangle \right] \right\}. \quad (28)$$

When capture occurs i.e. when E's separation from P becomes less than a pre-specified (capture) radius, we must have

$$H_u(t; x_r, u^*, v, V_x) = 0; H_v(t; x_r, u, v^*, V_x) = 0 \quad (29)$$

for $t \in (-T, 0]$. Seeking variational feedback controllers of the form: $\delta u = k_u \delta x, \delta v = k_v \delta x$ and putting (29) into (28) we may write

$$H_{uu} \delta u + \left(H_{ux} + f_u^T V_{xx}^* \right) \delta x + \frac{1}{2} \left(H_{uv} + H_{vu}^T \right) \delta v = 0, \quad (30a)$$

$$H_{vv} \delta v + \left(H_{vx} + f_v^T V_{xx}^* \right) \delta x + \frac{1}{2} \left(H_{vu} + H_{uv}^T \right) \delta u = 0 \quad (30b)$$

so that evaluating (30) at x_r, u^*, and v^*, we have

$$k_u = -H_{uu}^{-1} \left[H_{uv} k_v + \left(H_{ux} + f_u^T V_{xx}^* \right) \right], \text{ and that}$$

$$k_v = -H_{vv}^{-1} \left[H_{vu} k_u + \left(H_{vx} + f_v^T V_{xx}^* \right) \right]. \quad (31)$$

In equation (31), the two players' control strategies are interdependent: the optimal control for the nominal agent is to choose a strategy that is an optimal response to the action choice of the adversary. This is akin to the Newton-Raphson scheme which "correctly" estimates the maximizing δu^* and and minimizing δv^* provided that $|u - u^*|, |v - v^*| < \varepsilon$ for $\varepsilon > 0$ and positive-definite H_{uu}^{-1} and H_{vv}^{-1}. Substituting (31) into (28), we have

$$\min \left\{ 0, H + \langle H_x + V_{xx}^* f, \delta x \rangle + \frac{1}{2} \langle \delta x, \right.$$
$$\left. \left(H_{xx} + f_x^T V_{xx}^* + V_{xx}^* f_x + k_u^T H_{uu} k_u + k_v^T H_{vv} k_v \right) \delta x \rangle \right\}. \quad (32)$$

Similar to [2], we discard terms involving the pairs $(\delta u, \delta x)$ and $(\delta v, \delta x)$ beyond a linear order because the l.h.s is expanded to second-order only. Comparing (32) to the l.h.s. of (25), we find that

$$-\frac{\partial \tilde{V}^*}{\partial t} - \frac{\partial V_r}{\partial t} = \min\{0, H\}, \quad (33a)$$

$$-\frac{\partial V_x^*}{\partial t} = \min\{0, H_x + V_{xx}^* f\}, \quad (33b)$$

$$-\frac{\partial V_{xx}^*}{\partial t} = \min\left\{0, H_{xx} + f_x^T V_{xx}^* + V_{xx}^* f_x + k_u^T H_{uu} k_u + k_v^T H_{vv} k_v\right\}. \quad (33c)$$

Furthermore, comparing (33) with (24), and using the first-order necessary condition of optimality cf. (29), we have

$$-\dot{V}^* = \min\{0, H - H(t; x_r, u_r, v_r, V_x^*)\} \quad (34a)$$

$$-\dot{V}_x^* = \min\{0, H_x + V_{xx}^*(f - f(t; x_r, u_r, v_r))\} \quad (34b)$$

$$-\dot{V}_{xx}^* = \min\left\{0, H_{xx} + f_x^T V_{xx}^* + V_{xx}^* f_x + k_u^T H_{uu} k_u + k_v^T H_{vv} k_v\right\}$$

where every quantity in (34) is evaluated at $t; x_r, u^*, v^{*6}$. Note that equation (34) signifies the terms to be computed in the typical *backward pass* of DDP-like algorithms namely, start at the final time $t = 0$ and proceed backwards until $-T$; whilst storing the locally linear control gains i.e. (31) at every step of the backward pass to keep the necessary conditions of optimality. In a *forward pass*, the following new controls

$$u(\tau) = u^*(\tau) + k_u \delta x(\tau), \quad (35a)$$

$$v(\tau) = v^*(\tau) + k_v \delta x(\tau), \tau \in [-t, 0], t > T \quad (35b)$$

are updated, proceeding forward in time from $-T$ to 0 [7].

The linear differential variational HJI equation for (11) thus becomes (for sufficiently small δx):

$$-\frac{\partial \hat{\tilde{V}}}{\partial t} - \frac{\partial V_r}{\partial t} - \left\langle \frac{\partial \hat{V}_x}{\partial t}, \delta x \right\rangle - \frac{1}{2} \left\langle \delta x, \frac{\partial \hat{V}_{xx}}{\partial t} \delta x \right\rangle = \min\{0,$$
$$[H(t; x_r, u^*, v^*, \hat{V}_x) + \langle H_x + \hat{V}_{xx} f, \delta x \rangle + \langle H_u, k_u \delta x \rangle$$
$$+ \langle H_v, k_v \delta x \rangle$$
$$+ \langle k_u \delta x, (H_{ux} + f_u^T \hat{V}_{xx}) \delta x \rangle + \left\langle k_v \delta x, (H_{vx} + f_v^T \hat{V}_{xx}) \delta x \right\rangle$$
$$+ \frac{1}{2} \left\langle k_u \delta x, H_{uv} \delta v + \frac{1}{2} \langle \delta v, H_{vu} k_u \delta x \rangle \right\rangle + \frac{1}{2} \langle k_u \delta x, H_{uu} k_u \delta x \rangle$$
$$\left. + \frac{1}{2} \langle k_v \delta x, H_{vv} k_v \delta x \rangle + \frac{1}{2} \left\langle \delta x, \left(H_{xx} + f_x^T \hat{V}_{xx} + \hat{V}_{xx} f_x \right) \delta x \right\rangle \right] \right\}. \quad (36)$$

where the hat terms indicate that (35) is used. *Note that we have discarded the V_{xxx} terms in our derivations. We also do away with the application of the penalizing $\varepsilon > 0$-term on the variational controls in (35) as proposed in [4] in lieu of standard step-size adjustment mechanisms to ensure a sufficiently small δx.*

In a similar spirit to (34), we have

$$-\hat{\dot{V}}^* = \min\{0, H - H(t; x_r, u_r, v_r, \hat{V}_x^*)\} \quad (37a)$$

$$-\hat{\dot{V}}_x^* = \min\left\{0, H_x + \hat{V}_{xx}^*(f - f(t; x_r, u_r, v_r))\right\} \quad (37b)$$

$$-\hat{\dot{V}}_{xx}^* = \min\left\{0, H_{xx} + f_x^T \hat{V}_{xx}^* + \hat{V}_{xx}^* f_x + k_u^T H_{uu} k_u + k_v^T H_{vv} k_v\right\}$$

with the caret symbols signifying predictions upon application of the policies (35).

V. COMPUTATIONAL PROCEDURE AND DISCUSSION.

Suppose that the position of points, $\{g.x_i\}_{i=1}^N$, where all trajectories emerge is known. In what follows, we describe the numerical scheme for carrying out the integrations (37) and computing the overapproximated level set of $V(x_r + \delta x, t)$ i.e. (36).

A. Computational Scheme

First, a schedule of controls $\{u_r(t), v_r(t)\}_{t=-T}^0$ needed for computing nominal states $\{x_r(t)\}_{t=-T}^0$ is initialized as a problem-dependent parameter and then used to run $\{x_r(t)\}_{t=-T}^0$; if the nominal control schedules are not available, they can be set from the system's passive dynamics. The predicted cost improvement starting from the final time and going backwards in time is (cf. (37)):

$$| \hat{V}^*(x, \tau) | = \int_0^\tau \min\{0, H - H(t; x_r, u_r, v_r, \hat{V}_x^*)\}, \tau \gg -T, \quad (38)$$

---

[6] The "*" sign implies that the optimal $u^* = u_r + \delta u^*$ and $v^* = v_r + \delta v^*$ are used – essentially the optimal control-disturbance pair for $x = x_r + \delta x$.

[7] Note that gains k_u, k_v are only used in computing feedback controllers for the discriminating kernel in a backward reachability setting.

Algorithm 1 Successive Approximation Scheme.

1: **procedure** VarHJIInt(η ρ)   ▷ Given $\eta > 0$, $\rho \in (0, 1]$.   ▷ Stop conditions.
2:    Initialize Buffer $V_{buf} = \emptyset$.
3:    **for all** initial trajectories $X = \{\boldsymbol{x}_i(t_0)\}_{i=1}^N \in \Omega$ **do**
4:       Generate schedule $\pi_{r_i} = \{\boldsymbol{u}_{r_i}(t), \boldsymbol{v}_{r_i}(t)\}_{t=0}^{t=-T}$;
5:       Initialize $|\hat{\boldsymbol{V}}_i(\boldsymbol{x}_{r_i}, t)| = \infty$;   ▷ cf. (39);
6:       $\boldsymbol{x}_{r_i}, \pi_i^* = $ BackwardPass$(\boldsymbol{x}_i, \pi_{r_i}, |\hat{\boldsymbol{V}}_i(\cdot)|)$;
7:       $\boldsymbol{V}_i^*(\boldsymbol{x}_{r_i} + \delta\boldsymbol{x}_i, t), \pi_{r_i}^* = $ ForwardPass$(\boldsymbol{x}_{r_i}, \pi_i^*)$;
8:       $\boldsymbol{V}_{buf} \leftarrow \boldsymbol{V}_{buf} \cup \boldsymbol{V}_i^*(\boldsymbol{x}_{r_i} + \delta\boldsymbol{x}_i, t)$;
9:    **end for**
10:    Compute zero-levelset of V_{buf} ▷ Using [19].
11: **end procedure**

1: **function** BackwardPass$(\boldsymbol{x}_i, \pi_{r_i}, |\hat{\boldsymbol{V}}_i^*|)$
2:    Initialize $t_{eff} = -T, k = 1 \cdots K$.
3:    **for** $t = 0, -k\Delta t, \cdots - T$ **do**   ▷ $\Delta t = T/(K-1)$;
4:       Unpack $\boldsymbol{u}_{r_i}(t), \boldsymbol{v}_{r_i}(t) := \pi_{r_i}(t)$;
5:       $\dot{\boldsymbol{x}}_{r_i}(t) \leftarrow f(\boldsymbol{x}_{r_i}, \boldsymbol{u}_{r_i}(t), \boldsymbol{v}_{r_i}(t))$ & compute $\boldsymbol{V}_{r_i}(\boldsymbol{x}_{r_i}, t)$;
6:       Compute $|\hat{\boldsymbol{V}}_i^*(\boldsymbol{x}, t)| \leftarrow \boldsymbol{V}_i^*(\boldsymbol{x}_{r_i}, t) - \boldsymbol{V}_{r_i}(\boldsymbol{x}_{r_i}, t)$;
7:       **if** $|\hat{\boldsymbol{V}}_i^*(\boldsymbol{x}_{r_i}, t)| < \eta$ **then**
8:          Update $\boldsymbol{x}_{r_i}$ ▷ e.g. via Runge-Kutta integration;
9:          Accept $\boldsymbol{x}_{r_i}$ only if condition (39) is satisfied;
10:          Set $t_{eff} = t$ and Terminate loop.
11:       **else** Line search for a smaller $\delta\boldsymbol{x}_{r_i}$; restart line 3.
12:       **end if**
13:       $\delta\pi_i^*(t) := (\delta\boldsymbol{u}_i^*, \delta\boldsymbol{v}_i^*) \leftarrow$ extrema of $\boldsymbol{H}_i(\boldsymbol{x}_{r_i}, \pi_{r_i}^*, p, t)$
14:       Update $\pi_i^*(t) := (\boldsymbol{u}_i^*(t), \boldsymbol{v}_i^*(t)) \leftarrow \pi_{r_i}(t) + \delta\pi_i^*(t)$;
15:    **end for**
16:    **return** $\boldsymbol{x}_{r_i}, \pi_i^* := \{\boldsymbol{u}_i^*(t), \boldsymbol{v}_i^*(t)\}_{t=0}^{t=t_{eff}}$.
17: **end function**

1: **function** ForwardPass$(\boldsymbol{x}_{r_i}, \pi_i^*)$
2:    **for** $t = -T, (K-1)\Delta t, \cdots, -2\Delta t, -\Delta t, 0$ **do**
3:       Unpack controllers $(\boldsymbol{u}_i^*(t), \boldsymbol{v}_i^*(t)) := \pi_i^*(t)$;
4:       Compute the extremizing $\delta\boldsymbol{u}$ & $\delta\boldsymbol{v}$ cf. (28), (31);
5:       Run $\delta\dot{\boldsymbol{x}}_i(t) \leftarrow f(\cdot, \delta\boldsymbol{u}_i(t), \delta\boldsymbol{v}_i(t))$;
6:       $\delta\boldsymbol{x}_i(t) \leftarrow \delta\dot{\boldsymbol{x}}_{r_i}(t)$ set $t_f = t_{ext}$ ▷ e.g. Euler/RK integration;
7:       Compute $\boldsymbol{V}_i^*(\boldsymbol{x}_{r_i} + \delta\boldsymbol{x}_i, t)$ cf. (20);
8:       Update $\pi_{r_i}^*(t) := (\boldsymbol{u}_r(t), \boldsymbol{v}_r(t))$ ▷ Using (26);
9:    **end for**
10:    **return** $\boldsymbol{V}_i^*(\boldsymbol{x}_{r_i} + \delta\boldsymbol{x}_i, 0), \pi_{r_i}^*$.
11: **end function**

---

while the actual cost improvement is given by (19). Similar to [2], define a cost improvement criterion $\rho > 0$ such that,

$$\tilde{\mathbf{V}}^\star(\boldsymbol{x}_r, t)/|\hat{\boldsymbol{V}}^*(\boldsymbol{x}_r, t)| > \rho \tag{39}$$

determines the "closeness" of the cost improvement to the cost prediction.

As seen in Algorithm 1, the procedure proceeds in two passes for all initial trajectories: (i) in a backward pass, costs (37) are estimated with the open loop sequence $\{\boldsymbol{u}_r(t), \boldsymbol{v}_r(t)\}_0^{t=-T}$. We generate $\boldsymbol{u}^*$ and $\boldsymbol{v}^*$ in (26) afterwards; (ii) in a forward pass, reversing the order of integration limits, controls $\boldsymbol{u}^*(t), \boldsymbol{v}^*(t), t \in [-T, 0]$ of (27) are then applied and the approximation in (20) is computed for every trajectory that emanates from the state space. For the line search procedure, an Armijo-Goldstein condition in a typical backtracking line search can be applied to iteratively keep $\delta\boldsymbol{x}$ small for a valid approximation of $\boldsymbol{V}(\cdot)$ i.e. (33). A regularization scheme similar to our previous work [6] can also be applied to keep the stagewise Hessians positive definite.

The union operator from line 8 of Algorithm 1 allows the recovery of the maximal BRS as proposed in [20]. Given the "safe" and "unsafe" sets that constitute $\boldsymbol{V}_{buf}$, the reachable sets can be over-approximated under a best-response strategy of the two-player game (See [6]). We have released code that computes the zero isocontour (levelset) of the (union of all trajectories') optimal value function stipulated on Line 10 of Algorithm 1. A method for obtaining this is available in [19], which is an implementation of [21].

B. Notes on Computational Complexity

The problem introduced in (33) can be solved with Newton's method to find the extremizing policies $\delta\pi$ for all t in Algorithm 1: with Hessian $\frac{\partial^2 V}{\partial \pi_u^2}$ for the maximizer and $\frac{\partial^2 V}{\partial u^2}, \frac{\partial^2 V}{\partial v^2}$ for the minimizer, the inversion of the Hessian matrix would constitute $O(T^3 m^3)$ CPU flops. Whereas the overall DDP-style computational scheme we have presented has a CPU cost per iteration of $O(N) + T \cdot (2n^3 + \frac{7}{2}n^2(n_u + n_v) + 2n(n_u + n_v)^2 + \frac{1}{3}(n_u + n_v)^3 + O(n^2) + O((n_u + n_v)^2))$ CPU flops (see [22, Appendix II] for details), where n_u and n_v are as given in section II. The polynomial time complexity of the presented scheme makes it more attractive compared to Newton's method or even level set methods which are well-known to scale exponentially. We refer readers to Pantoja [23] for a thorough differentiation between DDP and Newton's methods. In our opinion, recent first-order primal-dual algorithms such as Chambolle-Pock [24] may prove more computationally parsimonious for these problems.

A Note on Convergence: Conditions upon which the algorithm 1 converges is premised on the standard Hessians' positive-definiteness (PD) i.e. $\boldsymbol{H}_{uu}$ and $\boldsymbol{H}_{vv}$ of DDP algorithms. In addition, for linear dynamical equations (1), the cost function being PD convex is a sufficient requirement for PD stagewise Hessians. When 1 is nonlinear, stagewise PD of $\boldsymbol{H}_{uu}$ and $\boldsymbol{H}_{vv}$ is no longer guaranteed [25]. In such situations, one may explore (i) a Levenberg-Marquardt scheme, convert the scheme to steepest descent by turning the stagewise Hessian to an identity or using the *active shift* method of [25]'s Theorem IV.

A Note on Conservativeness of Our Approach: The computed level sets

A Note on Comparison with Lagrangian Methods: Lagrangian methods [26] resolve the flow field in (1) using support vector sets (such as polytopes and zonotopes) and are typically applied to linear systems. These sets have a preconceived notion on the geometric shape of the reachable set to be determined. In many real-world setting, however, polytopes and zonotopes are the exception rather than the norm as most safety sets can hardly be resolved using this naive geometric primitives. On the contrary, our algorithm computes the level sets as is without any assumption about the geometric curvature of the reachable set.

C. Notes on Convergence Well-Posedness

Throughout, we have assumed that the Hessian terms, $\boldsymbol{H}_{uu}$ and $\boldsymbol{H}_{vv}$ are positive-definite. This positive definiteness is critical in assuring the convergence properties of the proposed methods.

D. Hessian Perturbation Theory

- Conditions under which H may be positive definite under perturbations

VI. RESULTS AND DISCUSSION.

In what follows, we describe two experiments to demonstrate the veracity of our analysis. We are concerned with pursuit-evasion games on a vectogram. First, we solve Dreyfus' rocket problem [27] in a two-player differential game; TO-DO: we then follow-up with a Dubins' two vehicle differential game.
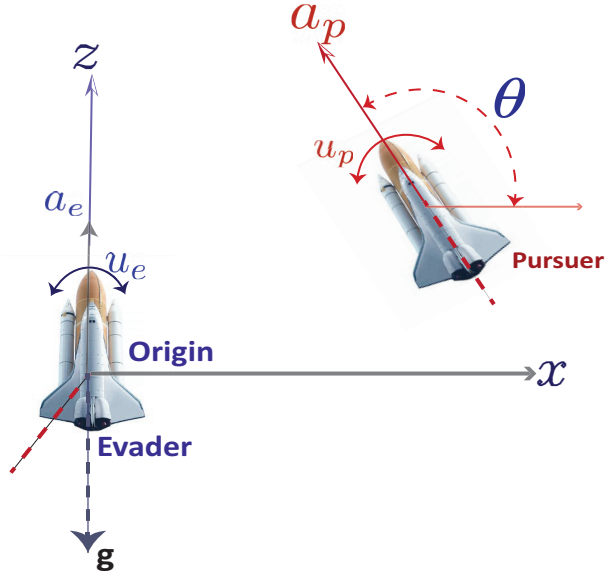
Fig. 1. Motion of two rockets on a Cartesian xz-plane with a thrust inclination in relative coordinates given by $\theta := u_p - u_e$.

A. The Rocket Launch Differential Game

We adopt the rocket launch problem of Dreyfus [28] 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)

$$\dot{x}_1 = x_3; \qquad x_1(t_0) = 0; \tag{40a}$$
$$\dot{x}_2 = x_4, \qquad x_2(t_0) = 0; \tag{40b}$$
$$\dot{x}_3 = a\cos u, \qquad x_3(t_0) = 0; \tag{40c}$$
$$\dot{x}_4 = a\sin u - g, \qquad x_4(t_0) = 0; \tag{40d}$$

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 (40) without the terminal time constraints as defined in [2]. The states of P and 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 1). The relevant kinematic equations are (40b) and (40d). We now make the problem amenable to a two-player differential game analysis so that every max and min operations are in the interior and no sudden changes from extremes are too aggravating in cost.

Therefore, we rewrite (40) with P's motion relative to E's along the (x, z) plane so that the relative orientation as shown in Fig. 1 is $\theta = u_p - u_e$. The coordinates of P are freely chosen; however, the coordinates of E are chosen a distance r away from (x, z) so that the EP vector's inclination measured counterclockwise from the $x-$axis is θ. Following the conventions in Fig. 1, 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

$$\dot{x} = a_p\cos\theta + u_e x, \tag{41a}$$
$$\dot{z} = a_p\sin\theta + a_e + u_e x - g, \tag{41b}$$
$$\dot{\theta} = u_p - u_e. \tag{41c}$$

The payoff, Φ, is the distance of P from E when capture occurs denoted as $\|PE\|_2$. Capture occurs when $\|PE\|_2 \le r$ for a pre-

specified capture radius, $r > 0$. In (41c), we say P controls u_p and is minimizing Φ, and E controls u_e and is maximizing P. The boundary of the *usable part* of the origin-centered circle of radius r^8 is $\|PE\|_2$ so that

$$r^2 = x^2 + z^2, \tag{42a}$$

and all capture points are specified by

$$\dot{r}(x, t) + \min\left[0, H\left(x, \frac{\partial r(x, t)}{\partial x}\right)\right] \le 0, \tag{43}$$

with the corresponding Hamiltonian

$$H(x, p) = -\max_{u_e \in \mathcal{U}_e} \min_{u_p \in \mathcal{U}_p} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^T \begin{bmatrix} a_p\cos\theta + u_e x \\ a_p\sin\theta + a_e + u_p x - g \\ u_p - u_e \end{bmatrix}. \tag{44}$$

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

$$\bar{u}_e = p_1 x - p_3, \tag{45a}$$
$$\bar{u}_p = p_3 - p_2 x. \tag{45b}$$

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

$$H(x, p) = -\cos(u)|ap_1| + \cos(u)|ap_1| - \sin(u)|ap_2| - \\ \sin(u)|ap_2| + u|p_3| - u|p_3|. \tag{46}$$

Using a distributed version of the levelset toolbox [19], the backward reachable tube of the game is depicted in Fig. 2. A game between the two players was run over 11 global optimization time steps. The initial value function (left inset of Fig. 2) 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. 2.
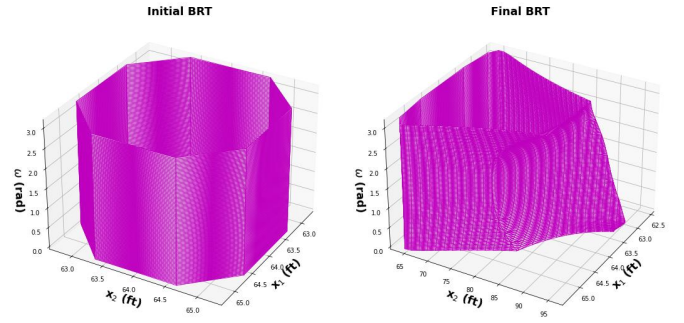

Fig. 2. Initial and final backward reachable tubes for the rocket system (cf. Fig. 1) computed using the method outlined in [10], [12], [29]. We set $a_e = a_p = 64 ft/sec^2$ and $g = 32 ft/sec^2$ as in Dreyfus' original example. We compute the reachable set by optimizing for the paths of slowest-quickest descent in equation (44).

REFERENCES

[1] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004. 1
[2] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. American Elsevier Publishing Company, Inc., New York, NY, 1970. 1, 3, 4, 5, 6
[3] A. E. Bryson and W. F. Denham, "A steepest-ascent method for solving optimum programming problems," 1962. 1

---

^{8}We set $r = 1.5 ft$ in our evaluations.

[4] D. H. Jacobson, "Differential Dynamic Programming Methods for Determining Optimal Control of Nonlinear Systems," *PhD Thesis, University of London*, vol. 103, no. 10, pp. 626–627, 1967. 1, 3, 4

[5] R. Isaacs, "Differential games: A mathematical theory with applications to warfare and pursuit, control and optimization." *Kreiger, Huntigton, NY*, 1965. 1

[6] O. Ogunmolu, N. Gans, and T. Summers, "Minimax iterative dynamic game: Application to nonlinear robot control tasks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6919–6925. 1, 5

[7] J. A. Sethian, "Level Set Methods And Fast Marching Methods: Evolving Interfaces In Computational Geometry, Fluid Mechanics, Computer Vision, And Materials Science," *Robotica*, vol. 18, no. 1, pp. 89–92, 2000. 1

[8] S. Osher and R. Fedkiw, "Level Set Methods and Dynamic Implicit Surfaces," *Applied Mechanics Reviews*, vol. 57, no. 3, pp. B15–B15, 2004. 1

[9] 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*, vol. 50, no. 7, pp. 947–957, 2005. 1, 2, 3

[10] I. M. Mitchell, "A Toolbox of Level Set Methods." *UBC Department of Computer Science Technical Report TR-2007-11*, 2007. 1, 6

[11] M. G. Crandall and P. L. Lions, "Two Approximations of Solutions of Hamilton-Jacobi Equations," *Mathematics of Computation*, vol. 43, no. 167, p. 1, 1984. 1

[12] L. Evans and P. E. Souganidis, "Differential Games And Representation Formulas For Solutions Of Hamilton-Jacobi-Isaacs Equations," *Indiana Univ. Math. J*, vol. 33, no. 5, pp. 773–797, 1984. 2, 6

[13] R. J. Elliott and N. J. Kalton, "Cauchy Problems for Certain Isaacs-Bellman Equations and Games of Survival," *Transactions of the American Mathematical Society*, vol. 198, no. 1970, p. 45, 1974. 2

[14] P. Cardaliaguet, "Nonsmooth Semipermeable Barriers, Isaacs' Equation, And Application to a Differential Game with One Target and Two Players," *Applied Mathematics and Optimization*, vol. 36, no. 2, pp. 125–146, 1997. 2

[15] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, "Matching-based Capture Strategies for 3D Heterogeneous Multiplayer Reach-Avoid Differential Games," *Automatica*, vol. 140, p. 110207, 2019. 2

[16] P.-L. Lions, *Generalized solutions of Hamilton-Jacobi equations*. London Pitman, 1982, vol. 69. 2

[17] M. G. Crandall and P.-L. Lions, "Viscosity solutions of hamilton-jacobi equations," *Transactions of the American mathematical society*, vol. 277, no. 1, pp. 1–42, 1983. 2

[18] W. Denham, "Book Reviews: Differential Dynamic Programming," *Review of Social Economy*, vol. 36, no. 2, pp. 228–229, 1978. 3

[19] O. Ogunmolu, "A Numerical Toolbox for the Scalable Analysis of Hamilton-Jacobi PDEs." https://github.com/robotsorcerer/LevelSetPy, 2022, accessed March 11, 2022. 5, 6

[20] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, "Decomposition of Reachable Sets and Tubes for a Class of Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3675–3688, 2018. 5

[21] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987. 5

[22] L.-Z. Liao and C. A. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," *IEEE Transactions on Automatic Control*, vol. 36, no. 6, pp. 692–706, 1991. 5

[23] J. D. O. Pantoja, "Differential Dynamic Programming and Newton's Method," *International Journal of Control*, vol. 47, no. 5, pp. 1539–1553, 1988. 5

[24] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems With Applications to Imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011. 5

[25] L.-Z. Liao and C. A. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," *IEEE Transactions on Automatic Control*, vol. 36, no. 6, pp. 692–706, 1991. 5

[26] M. Althoff, G. Frehse, and A. Girard, "Set Propagation Techniques for Reachability Analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 369–395, 2021. 5

[27] S. K. Mitter, "Successive Approximation Methods for the Solution of Optimal Control Problems," *Automatica*, vol. 3, no. 3-4, pp. 135–149, 1966. 5

[28] S. E. Dreyfus, "Control Problems With Linear Dynamics, Quadratic Criterion, and Linear Terminal Constraints," Rand Corp, Santa Monica Calif, Tech. Rep., 1966. 6

[29] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988. 6