A Second-Order Reachable Sets Computation Scheme via a Cauchy-Type Variational Hamilton-Jacobi-Isaacs Equation.

Lekan Molu*, Ian Abraham‡, and Sylvia Herbert†.

***Abstract*—Motivated by the scalability limitations of a popular Eulerian variational Hamilton-Jacobi-Isaacs (HJI) computational method for backward reachability analysis – essentially providing a least restrictive controller in problems that involve state or input constraints under a worst-possible disturbance, we introduce a second-order approximation algorithm for computing the zero sublevel sets of the variational HJI equation. By continuously approximating the value functions of all possible trajectories within a dynamical system's state space, under sufficient HJI partial differential equation regularity and continuity conditions throughout the state space, we show that with proper line search for step size adjustments and state feedback control under the worst-possible disturbance, we can compute the state set that are reachable within a prescribed verification time bound.**

I. Introduction

In this letter, we shall provide an approximation scheme for constructing the *discriminating kernel* for a controller's "constraints-satisfaction" design specifications problem. At issue is returning *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* under a *worst-possible disturbance* [1], and *over a time period*. This is a classic reachability theory problem. Our study is motivated by the exponential computational requirement of Eulerian methods [2], [3]. Ours is an extension of differential dynamic programming methods in Bolza-like objective functions inspired by [4], [5], and [6]'s second-order variational methods for solving general optimal control problems. In emphasis, we focus on pursuit-evasion games [7], where in an iterative dynamic game fashion [8], each agent's strategy depends on its opponent's control law as agents locally approximate successive trajectories and their values that emanate from a state space. We adopt a formal treatment, indicating how to resolve these problems computationally.

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 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 [3], [9] 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.) [2], [10], the BRS in one dimension is equivalent to the conservation of momentum equation [3]. 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 [3], [9]. 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.

Contributions: We describe a computational scheme, provide a summary of the complexity and convergence behavior for the overapproximated BRS using [2]'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: small Latin and Greek letters are scalars; in bold-font they are vectors. Exceptions: Players in a differential

*Microsoft Research, 300 Lafayette Street, New York, NY10012, USA. ‡Department of Mechanical Engineering, Yale University, New Haven, Conn, USA, †Department of Mechanical and Aerospace Engineering, UC San Diego, La Jolla, CA, USA. {lekanmolu@microsoft.com, ian.abraham@yale.edu, sherbert@eng.ucsd.edu }.

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: At issue are conflicting objectives between two players in pursuit-evasion games with each player being a pursuer ($\boldsymbol{P}$) or an evader ($\boldsymbol{E}$). 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

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

where the state $\boldsymbol{x}(t)$ evolves from some initial negative time $-T$ to a final time 0. **Assumptions**: 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[1]. 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

$$\bar{\mathscr{U}} \equiv \{\boldsymbol{u} : [-T, 0] \to \mathscr{U} \,|\, \boldsymbol{u} \text{ measurable}, \mathscr{U} \subset \mathbb{R}^{n_u}\}, \tag{2}$$
$$\bar{\mathscr{V}} \equiv \{\boldsymbol{v} : [-T, 0] \to \mathscr{V} \,|\, \boldsymbol{v} \text{ measurable}, \mathscr{V} \subset \mathbb{R}^{n_v}\}, \tag{3}$$

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 *non-anticipative 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,

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

satisfies (1) almost everywhere (a.e.) [2]. 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].

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

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.,

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

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

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

is the *target set* in the phase space $\Omega \times \mathbb{R}$ (proof in [2]). 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.,

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

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

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

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 [2], the states in the reachable set admit the following properties w.r.t the value function $\boldsymbol{V}$:

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

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-[2].

[2]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 $\boldsymbol{P}$.

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 $\beta : \mathscr{U} \to \mathscr{V}$. Suppose further that β is provided for each $-T \le \tau \le 0$ and $\boldsymbol{u}, \hat{\boldsymbol{u}} \in \mathscr{U}$. Then,

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

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] of the associated HJ-Isaacs (HJI) PDE given by

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

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

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

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 [2].

IV. Successive Approximation Scheme

Throughout this section, all feasible trajectories $\{\boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \cdots, \boldsymbol{x}_n\} \in \Omega$ are scheduled for iterative second-order expansion along the nonlinear state variations $\delta\boldsymbol{x}_i$ and about nominal states $\bar{\boldsymbol{x}}_i$; 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 $\boldsymbol{x}$.

A. Local Approximations to Nonlinear Trajectories

Suppose that system (1) is controllable everywhere on an interval $[-T,0]$ along a local trajectory $\boldsymbol{x}_r$ generated by admissible control disturbance pair $\boldsymbol{u}_r$, $\boldsymbol{v}_r$ starting from an initial phase $(\boldsymbol{x}_r, -T)$ so that

$$\dot{\boldsymbol{x}}_r(\tau) = f(t;\boldsymbol{x}_r(\tau), \boldsymbol{u}_r(\tau), \boldsymbol{v}_r(\tau)). \tag{13}$$

First, we apply local controls $\boldsymbol{u}_r(t)$ and $\boldsymbol{v}_r(t)$ on (1) so that the nominal value is $\boldsymbol{V}(t;\cdot,\cdot,\cdot)$ for a resulting nominal state $\boldsymbol{x}_r(\tau)$; $\tau \in [-T,0]$. System dynamics that describe variations from the nonlinear system cf. (1) with state and control pairs $\delta\boldsymbol{x}(t)$, $\delta\boldsymbol{u}(t)$, $\delta\boldsymbol{v}(t)$ respectively[3] can then be written as

$$\boldsymbol{x}(t) = \boldsymbol{x}_r(t) + \delta\boldsymbol{x}(t), \ \boldsymbol{u}(t) = \boldsymbol{u}_r(t) + \delta\boldsymbol{u}(t), \tag{14a}$$

$$\boldsymbol{v}(t) = \boldsymbol{v}_r(t) + \delta\boldsymbol{v}(t), \ t \in [-T,0]. \tag{14b}$$

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

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

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

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

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

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

and state trajectory

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

For $-T < t \le 0$ and a $\tau \in [t,0]$, let the *optimal cost* for using the optimal control $\boldsymbol{u}^\star(\tau) = \boldsymbol{u}_r(\tau) + \delta\boldsymbol{u}^\star(\tau)$ and optimal disturbance $\boldsymbol{v}^\star(\tau) = \boldsymbol{v}_r(\tau) + \delta\boldsymbol{v}^\star(\tau)$ be $\boldsymbol{V}^\star(\boldsymbol{x}_r, \tau)$; and the *nominal cost* for using $\boldsymbol{u}_r(\tau)$ be $\boldsymbol{V}_r(\boldsymbol{x}_r, t)$, so that on the phase $(\boldsymbol{x}_r, t)$, we have

$$\tilde{\boldsymbol{V}}^\star(\boldsymbol{x}_r, t) = \boldsymbol{V}^\star(\boldsymbol{x}_r, t) - \boldsymbol{V}_r(\boldsymbol{x}_r, t). \tag{19}$$

Next, we expand $\boldsymbol{V}^*$ in (17) under sufficient regularity assumptions.

B. Power Series Expansion Scheme

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

$$\boldsymbol{V}^\star(\boldsymbol{x}_r + \delta\boldsymbol{x}, t) = \tilde{\boldsymbol{V}}^\star(\boldsymbol{x}_r, t) + \boldsymbol{V}_r(\boldsymbol{x}_r, t) + \langle \boldsymbol{V}^*_{\boldsymbol{x}}(\boldsymbol{x}_r, t), \delta\boldsymbol{x} \rangle + \frac{1}{2}\langle \delta\boldsymbol{x}, \boldsymbol{V}^\star_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{x}_r, t)\delta\boldsymbol{x} \rangle + \text{h.o.t in } \delta\boldsymbol{x}, \tag{20}$$

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 [4], [6], [18].

Observe: If $\delta\boldsymbol{x}$ 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 $\boldsymbol{x}_r$ is constrained to be sufficiently close to $\boldsymbol{x}$, the state variation $\delta\boldsymbol{x}$ will be small, resulting in $\boldsymbol{x} \approx \boldsymbol{x}_r$ cf. (14)[4]; (ii) for small $\delta\boldsymbol{x}$, the expansion of (20) becomes consistent with second-order methods where quadratic terms in $\delta\boldsymbol{x}$ dominate higher-order terms. Hence, we can avoid the infinite data storage requirement by truncating the expansion in (20) at second-order terms in $\delta\boldsymbol{x}$. This will incur an $O(\delta\boldsymbol{x}^3)$ approximation

[3] Note that $\delta\boldsymbol{x}(t)$, $\delta\boldsymbol{u}(t)$, and $\delta\boldsymbol{v}(t)$ are respectively measured with respect to $\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)$ and are not necessarily small.

[4] A scheme to keep the variation $\delta\boldsymbol{x}$ small is discussed in section V. Ultimately, the $\delta\boldsymbol{u}, \delta\boldsymbol{v}$ terms will be quadratic in $\delta\boldsymbol{x}$ if we neglect h.o.t.

error, affording us realizable control laws that can be executed on the system (1). Thus, we rewrite (20) as

$$\boldsymbol{V}^{\star}(\boldsymbol{x}_r+\delta\boldsymbol{x},t)=\tilde{\boldsymbol{V}}^{\star}(\boldsymbol{x}_r,t)+\boldsymbol{V}_r(\boldsymbol{x}_r,t)+\langle\boldsymbol{V}_{\boldsymbol{x}}^{*}(\boldsymbol{x}_r,t),\delta\boldsymbol{x}\rangle + \frac{1}{2}\langle\delta\boldsymbol{x},\boldsymbol{V}_{\boldsymbol{xx}}^{\star}(\boldsymbol{x}_r,t)\delta\boldsymbol{x}\rangle. \tag{21}$$

Denote by $\boldsymbol{V}_{\boldsymbol{x}}^{\star}(\boldsymbol{x}_r+\delta\boldsymbol{x},t)$ the optimal value of the co-state $\frac{\partial\boldsymbol{V}^{\star}}{\partial\boldsymbol{x}}(\boldsymbol{x}_r+\delta\boldsymbol{x},t)$ on the r.h.s of (17). Then, expanding up to second order we find that

$$\boldsymbol{V}_{\boldsymbol{x}}^{\star}(\boldsymbol{x}_r+\delta\boldsymbol{x},t)=\boldsymbol{V}_{\boldsymbol{x}}^{\star}(\boldsymbol{x}_r,t)+\langle\boldsymbol{V}_{\boldsymbol{xx}}^{\star}(\boldsymbol{x}_r,t),\delta\boldsymbol{x}\rangle, \tag{22}$$

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

$$-\frac{\partial\tilde{\boldsymbol{V}}^{*}}{\partial t}-\frac{\partial\boldsymbol{V}_r}{\partial t}-\left\langle\frac{\partial\boldsymbol{V}_{\boldsymbol{x}}^{*}}{\partial t},\delta\boldsymbol{x}\right\rangle-\frac{1}{2}\left\langle\delta\boldsymbol{x},\frac{\partial\boldsymbol{V}_{\boldsymbol{xx}}^{*}}{\partial t}\delta\boldsymbol{x}\right\rangle= \min\left\{0,\max_{\delta\boldsymbol{u}\in\mathscr{U}}\min_{\delta\boldsymbol{v}\in\mathscr{V}}\left\langle f^{T}(t;\boldsymbol{x}_r+\delta\boldsymbol{x},\boldsymbol{u}_r+\delta\boldsymbol{u},\boldsymbol{v}_r+\delta\boldsymbol{v}),\boldsymbol{V}_{\boldsymbol{x}}^{*}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}\delta\boldsymbol{x}\right\rangle\right\}. \tag{23}$$

From (19), we may write

$$\frac{d}{dt}\left(\boldsymbol{V}_r+\tilde{\boldsymbol{V}}^{*}\right)=\frac{\partial}{\partial t}\left(\boldsymbol{V}_r+\tilde{\boldsymbol{V}}^{*}\right)+\left\langle f^{T}(t;\boldsymbol{x}_r,\boldsymbol{u}_r,\boldsymbol{v}_r),\boldsymbol{V}_{\boldsymbol{x}}^{*}\right\rangle \tag{24a}$$

$$\dot{\boldsymbol{V}}_{\boldsymbol{x}}^{*}=\frac{\partial\boldsymbol{V}_{\boldsymbol{xx}}^{*}}{\partial t}+\langle f^{T}(t;\boldsymbol{x}_r,\boldsymbol{u}_r,\boldsymbol{v}_r),\boldsymbol{V}_{\boldsymbol{xx}}^{*}\rangle,\ \dot{\boldsymbol{V}}_{\boldsymbol{xx}}^{*}=\frac{\partial\boldsymbol{V}_{\boldsymbol{xx}}^{*}}{\partial t}. \tag{24b}$$

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

$$-\frac{\partial\tilde{\boldsymbol{V}}^{*}}{\partial t}-\frac{\partial\boldsymbol{V}_r}{\partial t}-\left\langle\frac{\partial\boldsymbol{V}_{\boldsymbol{x}}^{*}}{\partial t},\delta\boldsymbol{x}\right\rangle-\frac{1}{2}\left\langle\delta\boldsymbol{x},\frac{\partial\boldsymbol{V}_{\boldsymbol{xx}}^{*}}{\partial t}\delta\boldsymbol{x}\right\rangle= \min\left\{\boldsymbol{0},\max_{\delta\boldsymbol{u}\in\mathscr{U}}\min_{\delta\boldsymbol{v}\in\mathscr{V}}[\boldsymbol{H}(t;\boldsymbol{x}_r+\delta\boldsymbol{x},\boldsymbol{u}_r+\delta\boldsymbol{u},\boldsymbol{v}_r+\delta\boldsymbol{v},\boldsymbol{V}_{\boldsymbol{x}}^{*})+\langle\boldsymbol{V}_{\boldsymbol{xx}}^{*}\delta\boldsymbol{x},f(t;\boldsymbol{x}_r+\delta\boldsymbol{x},\boldsymbol{u}_r+\delta\boldsymbol{u},\boldsymbol{v}_r+\delta\boldsymbol{v})\rangle]\right\} \tag{25}$$

where $\boldsymbol{H}(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{V}_{\boldsymbol{x}}^{*})=\langle\boldsymbol{V}_{\boldsymbol{x}}^{*},f(t;\boldsymbol{x},\boldsymbol{u},\boldsymbol{v})\rangle$.

C. Variational HJI Linear Differential Equation

Let the respective maximizing and minimizing control-disturbance pair when $\boldsymbol{x}=\boldsymbol{x}_r$ at time t be

$$\boldsymbol{u}^{*}=\boldsymbol{u}_r+\delta\boldsymbol{u}^{*},\ \boldsymbol{v}^{*}=\boldsymbol{v}_r+\delta\boldsymbol{v}^{*}. \tag{26}$$

Given the variation $\delta\boldsymbol{x}$ on the r.h.s. of (25), write the respective maximizing evader's and minimizing pursuer's controls for state $\boldsymbol{x}=\boldsymbol{x}_r+\delta\boldsymbol{x}$

$$\boldsymbol{u}=\boldsymbol{u}_r+\delta\boldsymbol{u}^{*}+\delta\boldsymbol{u},\ \boldsymbol{v}=\boldsymbol{v}_r+\delta\boldsymbol{v}^{*}+\delta\boldsymbol{v}. \tag{27}$$

Note here that $\delta\boldsymbol{u}$ and $\delta\boldsymbol{v}$ are as yet to be determined. Henceforward, we adopt the abbreviations $\boldsymbol{H}=\boldsymbol{H}(t;\boldsymbol{x}_r+\delta\boldsymbol{x},\boldsymbol{u}_r+\delta\boldsymbol{u},\boldsymbol{v}_r+\delta\boldsymbol{v},\boldsymbol{V}_{\boldsymbol{x}}^{*})$ and $f=f(t;\boldsymbol{x}_r+\delta\boldsymbol{x},\boldsymbol{u}_r+\delta\boldsymbol{u},\boldsymbol{v}_r+\delta\boldsymbol{v})$.

Expanding the r.h.s. of (25) about $\boldsymbol{x}_r,\boldsymbol{u}^{*},\boldsymbol{v}^{*}$ at time t up to second-order as before, we have

$$\min\left\{\boldsymbol{0},\max_{\delta\boldsymbol{u}\in\mathscr{U}}\min_{\delta\boldsymbol{v}\in\mathscr{V}}[\boldsymbol{H}+\langle\boldsymbol{H}_{\boldsymbol{x}}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f,\delta\boldsymbol{x}\rangle+\langle\boldsymbol{H}_{\boldsymbol{u}},\delta\boldsymbol{u}\rangle+\langle\boldsymbol{H}_{\boldsymbol{v}},\delta\boldsymbol{v}\rangle+\langle\delta\boldsymbol{u},(\boldsymbol{H}_{\boldsymbol{ux}}+f_{\boldsymbol{u}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*})\delta\boldsymbol{x}\rangle+\left\langle\delta\boldsymbol{v},(\boldsymbol{H}_{\boldsymbol{vx}}+f_{\boldsymbol{v}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*})\delta\boldsymbol{x}\right\rangle+\frac{1}{2}\left\langle\delta\boldsymbol{u},\boldsymbol{H}_{\boldsymbol{uv}}\delta\boldsymbol{v}+\frac{1}{2}\langle\delta\boldsymbol{v},\boldsymbol{H}_{\boldsymbol{vu}}\delta\boldsymbol{u}\rangle\right\rangle+\frac{1}{2}\langle\delta\boldsymbol{u},\boldsymbol{H}_{\boldsymbol{uu}}\delta\boldsymbol{u}\rangle+\frac{1}{2}\langle\delta\boldsymbol{v},\boldsymbol{H}_{\boldsymbol{vv}}\delta\boldsymbol{v}\rangle+\frac{1}{2}\left\langle\delta\boldsymbol{x},\left(\boldsymbol{H}_{\boldsymbol{xx}}+f_{\boldsymbol{x}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f_{\boldsymbol{x}}\right)\delta\boldsymbol{x}\right\rangle\right]\right\}. \tag{28}$$

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

$$\boldsymbol{H}_{\boldsymbol{u}}(t;\boldsymbol{x}_r,\boldsymbol{u}^{\star},\boldsymbol{v},\boldsymbol{V}_{\boldsymbol{x}})=0;\boldsymbol{H}_{\boldsymbol{v}}(t;\boldsymbol{x}_r,\boldsymbol{u},\boldsymbol{v}^{\star},\boldsymbol{V}_{\boldsymbol{x}})=0 \tag{29}$$

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

$$\boldsymbol{H}_{\boldsymbol{uu}}\delta\boldsymbol{u}+\left(\boldsymbol{H}_{\boldsymbol{ux}}+f_{\boldsymbol{u}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}\right)\delta\boldsymbol{x}+\frac{1}{2}\left(\boldsymbol{H}_{\boldsymbol{uv}}+\boldsymbol{H}_{\boldsymbol{vu}}^{T}\right)\delta\boldsymbol{v}=0, \tag{30a}$$

$$\boldsymbol{H}_{\boldsymbol{vv}}\delta\boldsymbol{v}+\left(\boldsymbol{H}_{\boldsymbol{vx}}+f_{\boldsymbol{v}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}\right)\delta\boldsymbol{x}+\frac{1}{2}\left(\boldsymbol{H}_{\boldsymbol{vu}}+\boldsymbol{H}_{\boldsymbol{uv}}^{T}\right)\delta\boldsymbol{u}=0 \tag{30b}$$

so that evaluating (30b) at $\boldsymbol{x}_r$, $\boldsymbol{u}^{*}$, and $\boldsymbol{v}^{*}$ we find that

$$\begin{aligned}\boldsymbol{k}_{\boldsymbol{u}}&=-\boldsymbol{H}_{\boldsymbol{uu}}^{-1}\left[\boldsymbol{H}_{\boldsymbol{uv}}\boldsymbol{k}_{\boldsymbol{v}}+\left(\boldsymbol{H}_{\boldsymbol{ux}}+f_{\boldsymbol{u}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}\right)\right],\ \text{and that}\\ \boldsymbol{k}_{\boldsymbol{v}}&=-\boldsymbol{H}_{\boldsymbol{vv}}^{-1}\left[\boldsymbol{H}_{\boldsymbol{vu}}\boldsymbol{k}_{\boldsymbol{u}}+\left(\boldsymbol{H}_{\boldsymbol{vx}}+f_{\boldsymbol{v}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}\right)\right].\end{aligned} \tag{31}$$

In equation (31), the two players' control strategies are interdependent: the optima; action for the nominal agent is to choose a control strategy that is an optimal response to the action choice of the adversary. is akin to the Newton-Raphson scheme which "correctly" estimates the maximizing $\delta\boldsymbol{u}^{*}$ and and minimizing $\delta\boldsymbol{v}^{*}$ provided that $|\boldsymbol{u}-\boldsymbol{u}^{*}|,|\boldsymbol{v}-\boldsymbol{v}^{*}|<\varepsilon$ for $\varepsilon>0$ and positive-definite $\boldsymbol{H}_{\boldsymbol{uu}}^{-1}$ and $\boldsymbol{H}_{\boldsymbol{vv}}^{-1}$. Upon substitution, (28) satisfies

$$\min\left\{\boldsymbol{0},\boldsymbol{H}+\langle\boldsymbol{H}_{\boldsymbol{x}}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f,\delta\boldsymbol{x}\rangle+\frac{1}{2}\langle\delta\boldsymbol{x},\left(\boldsymbol{H}_{\boldsymbol{xx}}+f_{\boldsymbol{x}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f_{\boldsymbol{x}}+\boldsymbol{k}_{\boldsymbol{u}}^{T}\boldsymbol{H}_{\boldsymbol{uu}}\boldsymbol{k}_{\boldsymbol{u}}+\boldsymbol{k}_{\boldsymbol{v}}^{T}\boldsymbol{H}_{\boldsymbol{vv}}\boldsymbol{k}_{\boldsymbol{v}}\right)\delta\boldsymbol{x}\right\rangle\right\}. \tag{32}$$

Similar to [4], we discard terms involving the pairs $(\delta\boldsymbol{u},\delta\boldsymbol{x})$ and $(\delta\boldsymbol{v},\delta\boldsymbol{x})$ 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

$$-\frac{\partial\tilde{\boldsymbol{V}}^{*}}{\partial t}-\frac{\partial\boldsymbol{V}_r}{\partial t}=\min\{\boldsymbol{0},\boldsymbol{H}\}, \tag{33a}$$

$$-\frac{\partial\boldsymbol{V}_{\boldsymbol{x}}^{*}}{\partial t}=\min\{\boldsymbol{0},\boldsymbol{H}_{\boldsymbol{x}}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f\}, \tag{33b}$$

$$-\dot{\boldsymbol{V}}_{\boldsymbol{xx}}^{*}=\min\left\{\boldsymbol{0},\boldsymbol{H}_{\boldsymbol{xx}}+f_{\boldsymbol{x}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f_{\boldsymbol{x}}+\boldsymbol{k}_{\boldsymbol{u}}^{T}\boldsymbol{H}_{\boldsymbol{uu}}\boldsymbol{k}_{\boldsymbol{u}}+\boldsymbol{k}_{\boldsymbol{v}}^{T}\boldsymbol{H}_{\boldsymbol{vv}}\boldsymbol{k}_{\boldsymbol{v}}\right\}.$$

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

$$-\dot{\tilde{\boldsymbol{V}}}^{*}=\min\{\boldsymbol{0},\boldsymbol{H}-\boldsymbol{H}(t;\boldsymbol{x}_r,\boldsymbol{u}_r,\boldsymbol{v}_r,\boldsymbol{V}_{\boldsymbol{x}}^{*})\} \tag{34a}$$

$$-\dot{\boldsymbol{V}}_{\boldsymbol{x}}^{*}=\min\{0,\boldsymbol{H}_{\boldsymbol{x}}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}(f-f(t;\boldsymbol{x}_r,\boldsymbol{u}_r,\boldsymbol{v}_r))\} \tag{34b}$$

$$-\dot{\boldsymbol{V}}_{\boldsymbol{xx}}^{*}=\min\left\{\boldsymbol{0},\boldsymbol{H}_{\boldsymbol{xx}}+f_{\boldsymbol{x}}^{T}\boldsymbol{V}_{\boldsymbol{xx}}^{*}+\boldsymbol{V}_{\boldsymbol{xx}}^{*}f_{\boldsymbol{x}}+\boldsymbol{k}_{\boldsymbol{u}}^{T}\boldsymbol{H}_{\boldsymbol{uu}}\boldsymbol{k}_{\boldsymbol{u}}+\boldsymbol{k}_{\boldsymbol{v}}^{T}\boldsymbol{H}_{\boldsymbol{vv}}\boldsymbol{k}_{\boldsymbol{v}}\right\}$$

where every quantity in (34) is evaluated at $t; \boldsymbol{x}_r, \boldsymbol{u}^*, \boldsymbol{v}^*$[5]. 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

$$\boldsymbol{u}(\tau) = \boldsymbol{u}^*(\tau) + \boldsymbol{k_u}\delta\boldsymbol{x}(\tau), \tag{35a}$$
$$\boldsymbol{v}(\tau) = \boldsymbol{v}^*(\tau) + \boldsymbol{k_v}\delta\boldsymbol{x}(\tau),\ \tau \in [-t, 0],\ t > T \tag{35b}$$

are updated proceeding forward in time from $-T$ to 0 [6].

The linear differential variational HJI equation for (11) thus becomes (for sufficiently small $\delta\boldsymbol{x}$):

$$\begin{aligned}
-\frac{\partial \hat{\bar{V}}}{\partial t} - \frac{\partial \boldsymbol{V}_r}{\partial t} - \left\langle \frac{\partial \hat{\boldsymbol{V}}_{\boldsymbol{x}}}{\partial t}, \delta\boldsymbol{x} \right\rangle - \frac{1}{2}\left\langle \delta\boldsymbol{x}, \frac{\partial \hat{\boldsymbol{V}}_{\boldsymbol{xx}}}{\partial t}\delta\boldsymbol{x} \right\rangle = \min\{\boldsymbol{0}, \\
\Big[\boldsymbol{H}(t; \boldsymbol{x}_r, \boldsymbol{u}^*, \boldsymbol{v}^*, \hat{\boldsymbol{V}}_{\boldsymbol{x}}) + \langle \boldsymbol{H}_{\boldsymbol{x}} + \hat{\boldsymbol{V}}_{\boldsymbol{xx}} f, \delta\boldsymbol{x}\rangle + \langle \boldsymbol{H}_{\boldsymbol{u}}, \boldsymbol{k_u}\delta\boldsymbol{x}\rangle \\
+ \langle \boldsymbol{H}_{\boldsymbol{v}}, \boldsymbol{k_v}\delta\boldsymbol{x}\rangle \\
+ \langle \boldsymbol{k_u}\delta\boldsymbol{x}, (\boldsymbol{H}_{\boldsymbol{ux}} + f_{\boldsymbol{u}}^T \hat{\boldsymbol{V}}_{\boldsymbol{xx}})\delta\boldsymbol{x}\rangle + \langle \boldsymbol{k_v}\delta\boldsymbol{x}, (\boldsymbol{H}_{\boldsymbol{vx}} + f_{\boldsymbol{v}}^T \hat{\boldsymbol{V}}_{\boldsymbol{xx}})\delta\boldsymbol{x}\rangle \\
+ \frac{1}{2}\left\langle \boldsymbol{k_u}\delta\boldsymbol{x}, \boldsymbol{H}_{\boldsymbol{uv}}\delta\boldsymbol{v} + \frac{1}{2}\langle \delta\boldsymbol{v}, \boldsymbol{H}_{\boldsymbol{vu}}\boldsymbol{k_u}\delta\boldsymbol{x}\rangle \right\rangle + \frac{1}{2}\langle \boldsymbol{k_u}\delta\boldsymbol{x}, \boldsymbol{H}_{\boldsymbol{uu}}\boldsymbol{k_u}\delta\boldsymbol{x}\rangle \\
+ \frac{1}{2}\langle \boldsymbol{k_v}\delta\boldsymbol{x}, \boldsymbol{H}_{\boldsymbol{vv}}\boldsymbol{k_v}\delta\boldsymbol{x}\rangle + \frac{1}{2}\langle \delta\boldsymbol{x}, \left(\boldsymbol{H}_{\boldsymbol{xx}} + f_{\boldsymbol{x}}^T \hat{\boldsymbol{V}}_{\boldsymbol{xx}} + \hat{\boldsymbol{V}}_{\boldsymbol{xx}} f_{\boldsymbol{x}}\right)\delta\boldsymbol{x}\rangle \Big]\Big\}.
\end{aligned} \tag{36}$$

where the hat terms indicate that (35) is used. *Note that we have discarded the $\boldsymbol{V}_{xxx}$ 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 [6] in lieu of standard step-size adjustment mechanisms to ensure a sufficiently small $\delta\boldsymbol{x}$.*

In a similar spirit to (34), we have

$$-\dot{\hat{\boldsymbol{V}}}^* = \min\{\boldsymbol{0}, \boldsymbol{H} - \boldsymbol{H}(t; \boldsymbol{x}_r, \boldsymbol{u}_r, \boldsymbol{v}_r, \hat{\boldsymbol{V}}_{\boldsymbol{x}}^*)\} \tag{37a}$$
$$-\dot{\hat{\boldsymbol{V}}}_{\boldsymbol{x}}^* = \min\left\{0, \boldsymbol{H}_{\boldsymbol{x}} + \hat{\boldsymbol{V}}_{\boldsymbol{xx}}^*(f - f(t; \boldsymbol{x}_r, \boldsymbol{u}_r, \boldsymbol{v}_r))\right\} \tag{37b}$$
$$-\dot{\hat{\boldsymbol{V}}}_{\boldsymbol{xx}}^* = \min\left\{\boldsymbol{0}, \boldsymbol{H}_{\boldsymbol{xx}} + f_{\boldsymbol{x}}^T \hat{\boldsymbol{V}}_{\boldsymbol{xx}}^* + \hat{\boldsymbol{V}}_{\boldsymbol{xx}}^* f_{\boldsymbol{x}} + \boldsymbol{k_u}^T \boldsymbol{H}_{\boldsymbol{uu}}\boldsymbol{k_u} + \boldsymbol{k_v}^T \boldsymbol{H}_{\boldsymbol{vv}}\boldsymbol{k_v}\right\}$$

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

V. Computational Procedure and Discussion.

Suppose that the position of points, $\{g.\boldsymbol{x}_i\}_{i=1}^N$, 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 $\boldsymbol{V}(\boldsymbol{x}_r + \delta\boldsymbol{x}, t)$ i.e. (36).

A. Computational Scheme

First, a schedule of controls $\{\boldsymbol{u}_r(t), \boldsymbol{v}_r(t)\}_{t=-T}^0$ needed for computing nominal states $\{\boldsymbol{x}_r(t)\}_{t=-T}^0$ is initialized as a problem-dependent parameter and then used to run

Algorithm 1 Successive Approximation Scheme.

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

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

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

$\{\boldsymbol{x}_r(t)\}_{t=-T}^0$; 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)):

$$|\ \hat{\boldsymbol{V}}^*(\boldsymbol{x}, \tau)\ | = \int_0^\tau \min\{\boldsymbol{0}, \boldsymbol{H} - \boldsymbol{H}(t; \boldsymbol{x}_r, \boldsymbol{u}_r, \boldsymbol{v}_r, \hat{\boldsymbol{V}}_{\boldsymbol{x}}^*)\},\ \tau \gg -T, \tag{38}$$

while the actual cost improvement is given by (19). Similar to [4], define a cost improvement criterion $\rho > 0$ such that,

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

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

[5]The "*" sign implies that the optimal $\boldsymbol{u}^* = \boldsymbol{u}_r + \delta\boldsymbol{u}^*$ and $\boldsymbol{v}^* = \boldsymbol{v}_r + \delta\boldsymbol{v}^*$ are used – essentially the optimal control-disturbance pair for $\boldsymbol{x} = \boldsymbol{x}_r + \delta\boldsymbol{x}$.

[6]Note that gains $\boldsymbol{k_u}, \boldsymbol{k_v}$ are only used in computing feedback controllers for the discriminating kernel in a backward reachability setting.

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 $\boldsymbol{\delta x}$ small for a valid approximation of $\boldsymbol{V}(\cdot)$ i.e. (33). A regularization scheme similar to our previous work [8] 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 [8]). 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. Computational Complexity and Convergence Requirement

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 $\frac{\partial^2 V}{\partial \pi_\mu^2}$ 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.

REFERENCES

[1] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

[2] 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.

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

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

[5] A. E. Bryson and W. F. Denham, "A steepest-ascent method for solving optimum programming problems," 1962.

[6] 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.

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

[8] 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.

[9] 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.

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

[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.

[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.

[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.

[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.

[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.

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

[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.

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

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

[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.

[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.

[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.

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

[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.

[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.