Time-Dependent Surveillance-Evasion Games

Elliot Cartee[1,2], Lexiao Lai[3], Qianli Song[3], Alexander Vladimirsky[1]

***Abstract*— Surveillance-Evasion (SE) games form an important class of adversarial trajectory-planning problems. We consider time-dependent SE games, in which an Evader is trying to reach its target while minimizing the cumulative exposure to a moving enemy Observer. That Observer is simultaneously aiming to maximize the same exposure by choosing how often to use each of its predefined patrol trajectories. Following the framework introduced in [1], we develop efficient algorithms for finding Nash Equilibrium policies for both players by blending techniques from semi-infinite game theory, convex optimization, and multi-objective dynamic programming on continuous planning spaces. We illustrate our method on several examples with Observers using omnidirectional and angle-restricted sensors on a domain with occluding obstacles.**

I. INTRODUCTION

While optimal trajectory-planning is a common task in robotics, the notion of "optimality" can be based on many different criteria, such as minimizing time taken to reach the destination, energy required to traverse the path, or minimizing exposure to some sort of threat. In this paper, we focus on the latter in the context of Surveillance-Evasion (SE) games.

We model this adversarial trajectory-planning problem as a semi-infinite zero-sum game between two robotic players: an Observer (O) and an Evader (E) traveling through a domain with occluding obstacles. O chooses a probability distribution over a finite set of predefined surveillance plans. E chooses a probability distribution over an infinite set of trajectories that bring it to a target location before a specified deadline. O's position and orientation define a time-dependent *pointwise observability* function. E's goal is to minimize its expected *cumulative observability*, while O aims to maximize this same quantity. Our model is continuous in time and space, relying on numerical methods for partial differential equations (PDEs) to plan E's trajectories via dynamic programming. We also use techniques from multi-objective and convex optimization to find Nash equilibrium policies for both players.

Classical SE games were introduced in the 1970s and assumed that each player is fully aware of the opponent's state and can react to any changes in real time [2], [3]. In contrast, our version of the game is built on a different information structure, as each player makes their decisions based only on the probabilistic policy chosen by its opponent, rather than on that opponent's actual current state. In this sense, we are looking for optimal *open loop* controls, with a built-in uncertainty of the opponent's position. This information structure arises in applications where logistical considerations force the players to commit to strategies in advance.

This version of SE games was recently considered in [1] for a model that was rather simplified from the robotics point of view: it lacked any detailed kinematics for E, allowed a set of stationary locations (rather than general surveillance plans) for O, and used the pointwise observability based on omnidirectional sensors at O's location(s). Here we use the same algorithms for finding Nash equilibrium policies, but extend the approach introduced in [1] to more realistic settings. While we still use a simplified/isotropic model for E's dynamics, our observer might be non-stationary (choosing among patrol trajectories) and our sensor might be only effective in an angular sector relative to O's current heading.

In Section II, we review a PDE approach to E's deterministic trajectory-planning, assuming the pointwise observability is fixed or predetermined. Section III describes the model and algorithms for SE games, taking adversarial planning into account. Section IV focuses on numerical methods, while Section V examines the Nash equilibrium policies for several test problems. The limitations of our approach and directions for future work are discussed in Section VI.

II. EVASIVE PATH-PLANNING

We begin by considering a standard optimal control problem for the Evader under the assumption that the pointwise observability is fixed and fully known.

E starts at some position $\boldsymbol{x} \in \Omega$ at the time t, moves with a location-dependent speed $f(\boldsymbol{x}) > 0$, and can instantaneously change its direction of motion. Its dynamics are given by:

$$\boldsymbol{y}'(s) = f\big(\boldsymbol{y}(s)\big)\boldsymbol{a}(s), \quad \boldsymbol{y}(t) = \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{a} : \mathbb{R} \to S^1$ is a measurable *control function* – E's chosen (time-dependent) direction of motion. Its path is constrained to stay within $\Omega \subset \mathbb{R}^2$ (*e.g.,* avoiding any obstacles), and must reach the target $\boldsymbol{x}_{\mathrm{T}}$ by the time T.

Suppose that the pointwise observability function $K(\boldsymbol{x}, s)$ is given (i.e., the observer's strategy is fixed). Let $T_{\boldsymbol{a}} = \min\{s \geq 0 \mid \boldsymbol{y}(s) = \boldsymbol{x}_{\mathrm{T}}\}$ be the time it takes the Evader to reach its target using control $\boldsymbol{a}(\cdot)$. E's goal is to minimize its cumulative observability:

$$\mathcal{J}(\boldsymbol{x}, t, \boldsymbol{a}(\cdot)) = \begin{cases} \int_t^{T_{\boldsymbol{a}}} K(\boldsymbol{y}(s), s)ds, & T_{\boldsymbol{a}} \leq T \\ +\infty, & \text{otherwise} \end{cases} \tag{2}$$

*Much of this work was conducted in an REU program, partially supported by the NSF-RTG award (DMS-1645643). The 1st and 4th authors were also supported by the NSF ATD award (DMS-1738010). The last author's work is also supported by the Simons Foundation Fellowship.

[1]Department of Mathematics, Cornell University
[2]evc34@cornell.edu
[3]University of Hong Kong

The value function $u(\boldsymbol{x}, t)$ is then defined by

$$u(\boldsymbol{x}, t) = \inf_{\boldsymbol{a}(\cdot)} \mathcal{J}(\boldsymbol{x}, t, \boldsymbol{a}(\cdot)). \tag{3}$$

Standard arguments in optimal control theory (e.g., [4, Chapters 3,4]) show that u is the domain-constrained viscosity solution of a time-dependent Hamilton-Jacobi-Bellman (HJB) equation:

$$\begin{aligned} \frac{\partial u}{\partial t} + \min_{|\boldsymbol{a}|=1} \{f(\boldsymbol{x}) \nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{a} + K(\boldsymbol{x}, t)\} = 0; \\ u(\boldsymbol{x}, T) = +\infty, \qquad \forall \boldsymbol{x} \neq \boldsymbol{x}_{\mathrm{T}}; \\ u(\boldsymbol{x}_{\mathrm{T}}, t) = 0, \qquad \forall t \leq T. \end{aligned} \tag{4}$$

with an additional condition at $\partial\Omega \setminus \{\boldsymbol{x}_T\}$ that the minimum is taken over the subset of control values that ensure staying inside $\overline{\Omega}$. Wherever the value function u is smooth, the optimal direction of motion is $\boldsymbol{a}_* = -\nabla u / |\nabla u|$ and the above equation simplifies to a time-dependent Eikonal PDE

$$\frac{\partial u}{\partial t} - f(\boldsymbol{x}) |\nabla u(\boldsymbol{x}, t)| + K(\boldsymbol{x}, t) = 0. \tag{5}$$

In general, (5) often does not have a classical solution, but always has a unique viscosity solution, coinciding with the value function of the optimal control problem [4]. The optimal direction of motion $\boldsymbol{a}_*$ is then uniquely determined almost everywhere on a set $\{(\boldsymbol{x}, t) \,|\, u(\boldsymbol{x}, t) < +\infty\}$ since u is Lipschitz and thus differentiable except on a set of measure zero. However, for those rare starting positions where ∇u is discontinuous, the optimal trajectory is not unique: there can be multiple optimal initial directions $\boldsymbol{a}_*$, each of them producing a different optimal trajectory to $\boldsymbol{x}_{\mathrm{T}}$. The numerical methods for solving this PDE and tracing the optimal trajectories will be covered in section IV.

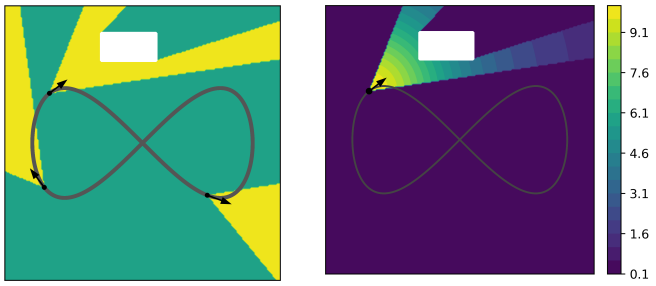


Fig. 1. LEFT: Visibility field (yellow) for 3 different Observer positions along a patrol trajectory (gray) in a domain with an occluding obstacle (white). RIGHT: The contour plot of the pointwise observability $K(\boldsymbol{x}, t_1)$, corresponding to O's position $\boldsymbol{z}(t_1)$.

Before switching to a game-theoretic version of the problem, we remark on the definition of pointwise observability K. Given O's current position $\boldsymbol{z}(t)$, one can define its *visibility field* $\mathcal{V}(t)$ restricted by obstacles and distance/angle limitations of the sensor. Throughout this paper, we assume that all obstacles are fully occluding and the sensors are only effective in an angular sector of width α centered on observer's current heading $\boldsymbol{h}(t)$. I.e., $\boldsymbol{x} \in \mathcal{V}(t)$ if the line segment $(\boldsymbol{z}(t), \boldsymbol{x})$ stays in Ω without intersecting any obstacles and the observation angle (i.e., the angle that $(\boldsymbol{z}(t), \boldsymbol{x})$ makes with $\boldsymbol{h}(t)$) is at most $\alpha/2$. Within $\mathcal{V}(t)$, the pointwise observability should be a decreasing function of the distance $|\boldsymbol{x} - \boldsymbol{z}(t)|$. Assuming $\boldsymbol{z}(t)$ is known, we use

$$K(\boldsymbol{x}, t) = \begin{cases} \frac{K_0}{|\boldsymbol{x} - \boldsymbol{z}(t)|^2 + 0.1} + \sigma, & \boldsymbol{x} \in \mathcal{V}(t); \\ \sigma, & \boldsymbol{x} \notin \mathcal{V}(t); \end{cases} \tag{6}$$

where K_0 and σ are positive constants. This sector-restricted observability is illustrated in Fig. 1.

III. Surveillance-Evasion Games

Here we describe the zero-sum game between O and E, following the general framework developed in [1]. We assume that

- O can use any patrol trajectory from the set $\mathcal{Z} = \{\boldsymbol{z}_1(t), \ldots \boldsymbol{z}_r(t)\}$ known to both players;
- E can use any of the infinitely many admissible trajectories that lead from $\boldsymbol{x}_{\mathrm{S}}$ to $\boldsymbol{x}_{\mathrm{T}}$ by the time T while staying in Ω and avoiding the obstacles.

The payoff of our game is the (expected) cumulative observability, with E as a minimizer and O as a maximizer. Both players are assumed to make their decisions ahead of time and cannot change their mind based on any information gained while traveling through Ω.

A. The "Evader-goes-second" problem

If O chooses the i-th patrol path, we can simply use $\boldsymbol{z}_i$ instead of $\boldsymbol{z}$ in (6) to define the corresponding pointwise observability K_i, cumulative observability $\mathcal{J}_i$, and value function u_i. The value of $u_i(\boldsymbol{x}_{\mathrm{S}}, 0)$ yields the cumulative observability along the $\boldsymbol{z}_i$-optimal trajectory (i.e., E's best response to O's choice of $\boldsymbol{z}_i$.) In Fig. 2 we show two circular patrol trajectories (traversed by O with the same angular velocity) and E's optimal trajectory in response to each of them. If O were forced to make its decision deterministically before E, it would simply choose whichever $\boldsymbol{z}_i$ yields the highest $u_i(\boldsymbol{x}_{\mathrm{S}}, 0)$.

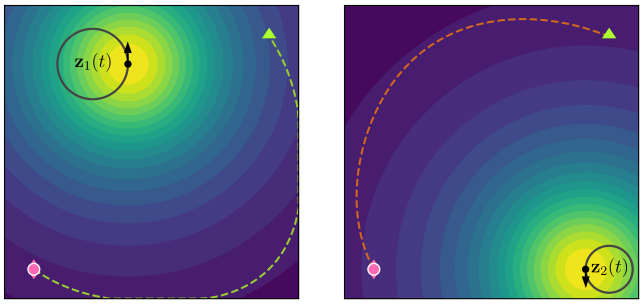
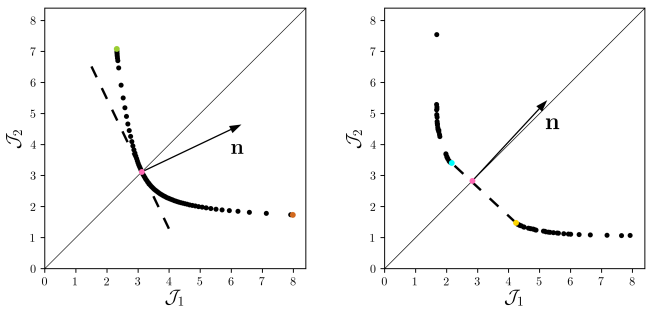


Fig. 2. E's optimal responses (shown in green and orange) corresponding to O's respective patrol trajectories. The sensor is omnidirectional (i.e., $\alpha = 2\pi$). The pointwise observability is time-dependent, and we show the contour plots for its *initial* version (i.e., $K_i(\boldsymbol{x}, 0)$) in both cases.

If O prefers to use a probability distribution $\lambda = (\lambda_1, \ldots, \lambda_r)$ over the set $\mathcal{Z}$ instead of choosing a specific $\boldsymbol{z}_i$, we can similarly define the expected pointwise observability

$K^\lambda = \sum_{i=1}^r \lambda_i K_i$ and the corresponding expected cumulative observability $\mathcal{J}^\lambda = \sum_{i=1}^r \lambda_i \mathcal{J}_i$. To find E's best response, the corresponding value function u^λ can be computed by solving (5) with K^λ used in place of K. We note that u^λ is *not* a linear combination of u_i's since this PDE is nonlinear.

This approach is also related to methods for *multi-objective* optimal control. It is generally impossible to minimize the observability with respect to all patrol trajectories simultaneously. Instead, one can consider the task of finding *Pareto-optimal* controls. A control $\tilde{\boldsymbol{a}}(\cdot)$ dominates $\boldsymbol{a}(\cdot)$ if $\mathcal{J}_i(\boldsymbol{x}_\mathrm{s}, 0, \tilde{\boldsymbol{a}}(\cdot)) \leq \mathcal{J}_i(\boldsymbol{x}_\mathrm{s}, 0, \boldsymbol{a}(\cdot))$ for all i, with the inequality strict for at least one i. We will say that $\boldsymbol{a}(\cdot)$ is Pareto-optimal if it is not dominated by any other control. Plotting the point $(\mathcal{J}_1, \ldots, \mathcal{J}_r)$ for each Pareto-optimal control we obtain a Pareto Front (PF), illustrated in Fig. 3 for two examples from section V. Each vector λ is normal to a support hyperplane of PF (on which $\mathcal{J}^\lambda = u^\lambda(\boldsymbol{x}_\mathrm{s}, 0)$). If all the elements of λ are positive, it is easy to show that any λ-optimal control is also Pareto-optimal. This is the basis of a *weighted-sum scalarization* approach to general multi-objective trajectory planning introduced in [5]. Unfortunately, it approximates only convex parts of PF [6]. Moreover, it requires imposing a fine grid on the space of λ's and solving the PDE for each λ-gridpoint. Alternative methods can recover the entire PF [7], [8], but are more computationally expensive. Luckily, for SE games we only need the points on PF corresponding to the players' *mutually* optimal policies; below we explain how this can be done through scalarization with a much smaller number of PDE solves.



(A) (B)

Fig. 3. The convex part of Pareto Fronts (PFs) corresponding to two examples from section V. PF approximations generated by using 101 λ values, solving for u^λ, and computing $(\mathcal{J}_1, \mathcal{J}_2)$ for λ-optimal trajectories. The dashed line shows a support hyperplane to PF, with $\boldsymbol{n}$ showing a scaled version of λ_*. Evader's optimal policy (shown in magenta) corresponds to the intersection of that hyperplane with a "central ray" $\mathcal{J}_1 = \mathcal{J}_2$.
LEFT: Example 1 has only one λ_*-optimal trajectory and E's optimal policy is deterministic; see Fig. 5. $\mathcal{J}_1$-optimal and $\mathcal{J}_2$-optimal trajectories are also represented on PF by the green and orange markers respectively.
RIGHT: In Example 2, E's optimal policy assigns non-zero probabilities to both λ_*-optimal trajectories (cyan and yellow); see Fig. 6.

O's goal is to find a λ maximizing

$$G(\lambda) \;=\; \min_{\boldsymbol{a}(\cdot)} \sum_{i=1}^r \lambda_i \mathcal{J}_i(\boldsymbol{x}_\mathrm{s}, 0, \boldsymbol{a}(\cdot)) \;=\; u^\lambda(\boldsymbol{x}_\mathrm{s}, 0).$$

Since G is a pointwise minimum of functions linear in λ, we know that $G(\lambda)$ is concave. Thus, O's optimal pdf λ_* can be found by standard methods of convex optimization. Our implementation relies on the projected supergradient ascent method [9, Chap. 8] to maximize G on the r-dimensional simplex of probability densities. During the iterative improvement of λ, we use gradient descent in u^λ starting from $(\boldsymbol{x}_\mathrm{s}, 0)$ to find E's λ-optimal trajectory. Integrating each of K_i's along that trajectory we obtain the supergradient of $G(\lambda)$, which is then used to update the λ for the next iteration. See Algorithm 4.2 in [1].

B. Nash Equilibria

It is restrictive to assume that E is always reacting to O's decisions. To get a more general interpretation of the game's value, we need to search for a *Nash equilibrium*. E can also define its policy probabilistically, by choosing a probability distribution θ over the infinite set of admissible controls. Given the players' policies, the expected payoff is

$$P(\lambda, \theta) = \mathbb{E}_{\lambda,\theta}\Big[\mathcal{J}(\boldsymbol{x}_\mathrm{s}, 0, \boldsymbol{a}(\cdot))\Big] = \sum_{i=1}^r \lambda_i \mathbb{E}_\theta\Big[\mathcal{J}_i(\boldsymbol{x}_\mathrm{s}, 0, \boldsymbol{a}(\cdot))\Big],$$

where $\boldsymbol{a}(\cdot)$ is a random admissible control chosen according to θ. A pair of policies (λ_*, θ_*) forms a Nash equilibrium if neither player can improve the payoff by changing its policy unilaterally. I.e.,

$$P(\lambda, \theta_*) \leq P(\lambda_*, \theta_*) \leq P(\lambda_*, \theta), \quad \forall \lambda, \theta.$$

There always exists at least one Nash equilibrium in the set of mixed/probabilistic policies. Moreover, in zero-sum two-player games, the payoff is the same for all Nash equilibria and is thus used to define the *game's value*; e.g., see [10]. Thus, when each player uses a Nash policy, it no longer matters which of them "goes first."

By the min-max theorem, the value of this game can be found by computing

$$P(\lambda_*, \theta_*) \;=\; \min_\theta \max_\lambda P(\lambda, \theta) \;=\; \max_\lambda \min_\theta P(\lambda, \theta). \quad (7)$$

In finite games, this is typically accomplished by linear programming [10], but since E has infinitely many possible paths, our game is *semi-infinite* [11], [12] and this approach is inapplicable. Luckily, it is easy to see that the last expression in (7) is equal to $\max_\lambda G(\lambda)$ and the previous subsection explains how to find the maximizer λ_* efficiently.

Once the algorithm converges to O's optimal λ_*, we need to find the second half of the Nash equilibrium: E's optimal probability distribution θ_* over the set of admissible trajectories. We accomplish this using the following properties, proven for a stationary Observer in [1], which similarly hold for the time-dependent case considered here. Suppose (λ_*, θ_*) is a Nash equilibrium, $\mathcal{I} = \{i | \lambda_{*,i} > 0\}$, and $\mathcal{A}$ is the set of all controls that have positive probability under θ_*.

1) If λ_* maximizes $G(\lambda)$, then there always exists a probability distribution θ_* over Evader's Pareto-optimal trajectories such that (λ_*, θ_*) is a Nash equilibrium and the set $\mathcal{A}$ has at most $|\mathcal{I}| \leq r$ elements.

2) If $\boldsymbol{a}(\cdot) \in \mathcal{A}$ then $\boldsymbol{a}(\cdot)$ is λ_*-optimal.
   I.e., $\sum_{i=1}^{r} \lambda_i \mathcal{J}_i(\boldsymbol{x}_\mathrm{S}, 0, \boldsymbol{a}(\cdot)) = u^{\lambda_*}(\boldsymbol{x}_\mathrm{S}, 0)$.
3) If $i \in \mathcal{I}$, then $\mathbb{E}_{\theta_*}[\mathcal{J}_i(\boldsymbol{x}_\mathrm{S}, 0, \boldsymbol{a}(\cdot))] = u^{\lambda_*}(\boldsymbol{x}_\mathrm{S}, 0)$.

For all starting positions where u^{λ_*} is differentiable, the optimal direction of motion is $-\nabla u^{\lambda_*}/|\nabla u^{\lambda_*}|$ and the optimal control is unique. Thus, it is natural to expect that, *generically*, there should be only one λ_*-optimal control $\boldsymbol{a}_*(\cdot)$ leading from $\boldsymbol{x}_\mathrm{S}$ to $\boldsymbol{x}_\mathrm{T}$ by the deadline T. This would imply that $\mathcal{A} = \{\boldsymbol{a}_*(\cdot)\}$; i.e., E's half of the Nash Equilibrium is pure/deterministic and $\mathcal{J}_i(\boldsymbol{x}_\mathrm{S}, 0, \boldsymbol{a}_*(\cdot)) = G(\lambda_*) = u^{\lambda_*}(\boldsymbol{x}_\mathrm{S}, 0)$ for all $i \in \mathcal{I}$. (We say that $\boldsymbol{a}_*(\cdot)$ is the intersection of the Pareto Front and the "central ray" in the $|\mathcal{I}|$-dimensional cost space.) This situation, illustrated in Fig. 3(A) is indeed quite common. But surprisingly, it is not generic; i.e., for many problems $\mathcal{A}$ must include more than one λ_*-optimal control. This is a counter-intuitive side-effect of changing λ to maximize G: K^λ keeps changing until $(\boldsymbol{x}_\mathrm{S}, 0)$ falls onto a "shockline", a set of discontinuities of ∇u^{λ_*}. A numerically implemented gradient descent in u^{λ_*} usually yields only one λ_*-optimal trajectory. Approximating *all* of them is much harder. We accomplish this by perturbing λ_* in various directions and computing the resulting unique optimal trajectories corresponding to the perturbed $(\lambda_* + \delta\lambda)$. We refer readers to Alg. 4.3 in [1] for a detailed discussion.

IV. Numerical Implementation

A. Numerics for HJB equation

We numerically solve (5) using a time-explicit first-order upwind finite-difference scheme on a 9-point stencil. Let $\boldsymbol{x}_{i,j} = (ih, jh)$, $t_k = k\Delta t$, and $U^k_{i,j} \approx u(\boldsymbol{x}_{i,j}, t_k)$. We will solve the equation backwards in time, computing $U^{k-1}_{i,j}$ using the values of U at time slice k. More specifically, we will compute an update from each of the eight triangular simplices generated by $\boldsymbol{x}_{i,j}$ and its eight nearest neighbors (see Fig. 4).

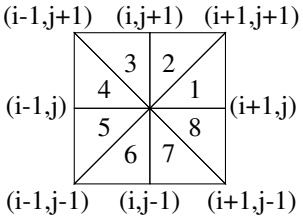


Fig. 4. Nine-point stencil used to discretize HJB equation at $(\boldsymbol{x}_{i,j}, t_k)$.

We first compute the first-order approximation $(U_x, U_y) \approx \nabla u(\boldsymbol{x}_{i,j}, t_k)$ based on a specific simplex. We then check the *upwinding condition*, requiring the approximate gradient to point from that same simplex. E.g., for simplex 1

$$U_x = \frac{U^k_{i+1,j} - U^k_{i,j}}{h}, \qquad U_y = \frac{U^k_{i+1,j+1} - U^k_{i+1,j}}{h},$$

and the corresponding upwinding condition is $U_x \le U_y \le 0$.

Focusing on a single simplex $l \in \{1, ..., 8\}$ with vertices $\boldsymbol{x}_{i,j}, \boldsymbol{x}^{l,1}$, and $\boldsymbol{x}^{l,2}$, the update from that simplex is

$$U_l = U^k_{i,j} - \Delta t f(\boldsymbol{x}_{i,j}) D^k_{i,j} + \Delta t K(\boldsymbol{x}_{i,j}, t_k),$$

where $D^k_{i,j}$ approximates $|\nabla u(\boldsymbol{x}_{i,j}, t_k)|$. If the upwinding condition is satisfied, we use $D^k_{i,j} = \sqrt{U_x^2 + U_y^2}$. Otherwise, we resort to a "one-sided" semi-Lagrangian update, with

$$D^k_{i,j} = \max\left(\frac{U^k_{i,j} - U(\boldsymbol{x}^{l,1}, t_k)}{|\boldsymbol{x}_{i,j} - \boldsymbol{x}^{l,1}|}, \frac{U^k_{i,j} - U(\boldsymbol{x}^{l,2}, t_k)}{|\boldsymbol{x}_{i,j} - \boldsymbol{x}^{l,2}|}, 0 \right).$$

We then set $U^{k-1}_{i,j}$ to be the smallest of simplex-specific updates $\{U_l\}$. The boundary conditions and terminal conditions are enforced by setting

$$U^k_{i,j} = \begin{cases} 0, & \boldsymbol{x}_{i,j} = \boldsymbol{x}_\mathrm{T} \text{ and } t_k \le T; \\ +\infty, & \boldsymbol{x}_{i,j} \notin \overline{\Omega} \text{ **or** } (\boldsymbol{x}_{i,j} \ne \boldsymbol{x}_\mathrm{T} \text{ and } t_k = T). \end{cases}$$

As long as the CFL stability condition $\Delta t \le h/(\max f(\boldsymbol{x}_{i,j}))$ is satisfied, this method is monotone and consistent [13], and therefore converges to the viscosity solution of (5) under grid refinement [14].

B. Time-dependent Visibility

Our definition of $\mathcal{V}_i(t)$ requires computing the set of points that have direct line of sight from $\boldsymbol{z}_i(t)$. Computational efficiency is very important as this needs to be computed for every time slice. Following [15], we first construct a continuous function $\phi(\boldsymbol{x})$ such that

$$\phi(\boldsymbol{x}) \begin{cases} > 0, & \boldsymbol{x} \in \Omega, \\ = 0, & \boldsymbol{x} \in \partial\Omega, \\ < 0, & \boldsymbol{x} \in \mathbb{R}^2 \setminus \overline{\Omega} \quad \text{(e.g., inside obstacles).} \end{cases}$$

Within each time slice t, we numerically solve the quasi-variational inequality

$$\max\{\nabla \psi^t(\boldsymbol{x}) \cdot \boldsymbol{r}(\boldsymbol{x}),\ \psi^t(\boldsymbol{x}) - \phi(\boldsymbol{x})\} = 0,$$
$$\psi^t(\boldsymbol{z}(t)) = \phi(\boldsymbol{z}(t)),$$

where $\boldsymbol{r}(\boldsymbol{x})$ is a unit vector pointing from $\boldsymbol{z}(t)$ to $\boldsymbol{x}$. Then $\{\psi^t \ge 0\}$ will define the set of points that are not occluded by obstacles. After computing ψ^t, we then enforce any additional distance or angular restrictions to get $\mathcal{V}_i(t)$. Combined with (6), this allows us to pre-compute and store $K_i(\boldsymbol{x}, t)$ for each patrol trajectory $\boldsymbol{z}_i(t)$ before starting to plan E's optimal responses to various Observer policies λ.

C. Time-dependent Optimal Path Tracer

Given our numerical solution $U^k_{i,j}$, we now wish to recover the Evader's optimal trajectory $\boldsymbol{y}(t)$. We approximate the trajectory with a series of points $\boldsymbol{y}^m \approx \boldsymbol{y}(m\Delta t)$, where Δt is the timestep used to numerically solve (5).

E's initial position is $\boldsymbol{y}^0 = \boldsymbol{y}(0) = \boldsymbol{x}_\mathrm{S}$. The rest of the path can be found in a semi-Lagrangian manner by computing

$$\boldsymbol{a}^m_* = \arg\min_{|\boldsymbol{a}| \le 1} \left\{ \widetilde{U}(\boldsymbol{y}^m + \Delta t f(\boldsymbol{y}^m)\boldsymbol{a},\ m\Delta t) \right\},$$
$$\boldsymbol{y}^{m+1} = \boldsymbol{y}^m + \Delta t \boldsymbol{a}^m_*.$$

where $\widetilde{U}$ is the function obtained from the grid values of U by trilinear interpolation. We terminate the path once we get close enough to the target so that $|\boldsymbol{y}_m - \boldsymbol{x}_{\mathrm{T}}| \leq f(\boldsymbol{y}_m)\Delta t$. In our numerical experiments, we optimize over $\{|\boldsymbol{a}| = 1\} \cup \{(0,0)\}$ instead of the entire unit disk $\{|\boldsymbol{a}| \leq 1\}$.

V. Numerical Experiments

In all of our examples[1], we assume that the domain is a unit square, sometimes containing impenetrable and occluding obstacles. This square is discretized on a 201×201 grid ($h = 0.005$). E's speed is uniform ($f(\boldsymbol{x}) = 1$) and the timestep is chosen according to the CFL stability condition: $\Delta t = h$. Our examples use observability functions $K_i(\boldsymbol{x}, t)$ of the form found in (6), with $K_0 = 1$ and $\sigma = 0.1$. In all examples, the Evader's deadline for reaching the target is $T = 4$, but all Nash equilibrium controls lead to a much earlier arrival. (In our setting, the cumulative observability is computed up to the arrival time $T_{\boldsymbol{a}}$ only.)

Example 1: No obstacles, two patrol trajectories, and omnidirectional sensors (i.e., $\alpha = 2\pi$). Observer uses the same angular speed and moves counterclockwise on both patrol trajectories, which were already introduced in Fig. 2. Nash equilibrium policies are shown in Fig. 5. See also the Pareto Front in Fig. 3(A).

Example 2: One obstacle, with the same sensors and patrol trajectories as Example 1. Nash equilibrium policies are shown in Fig. 6. Evader's optimal policy is probabilistic, relying on two λ_*-optimal trajectories that pass above and below the obstacle. See also the Pareto Front in Fig. 3(B).

Example 3: Three obstacles, two patrol trajectories with direction-restricted sensors ($\alpha = 2\pi/3$). The patrol trajectories each follow a figure-eight pattern. Nash equilibrium policies are shown in Fig. 7.

Example 4: Many obstacles, with 4 patrol trajectories on the same circle, but at different phases. Observer's angular speed is $\omega = 1/2\pi$, and with direction-restricted sensors (i.e., $\alpha = 2\pi/3$). Nash equilibrium policies are shown in Fig. 8.

VI. Conclusions

We have developed and tested a numerical method for SE games under uncertainty with isotropic Evader dynamics, a time-dependent Observer, and a direction-limited sensor. The efficiency of our approach stems from combining existing techniques from game theory, convex optimization, and dynamic programming in continuous state/time. The latter requires efficient numerical methods for time-dependent Eikonal PDEs. Incorporating more realistic models of Evader dynamics will require solving more general Hamilton-Jacobi PDEs. While our current implementation relies on an explicit first-order upwind scheme, the use of higher-order accurate discretizations [13], [16] would be desirable in the future. Additional speed improvements might be also attained through a time-dependent version of causal domain restriction [17]. All numerical examples presented above assumed that the Observer is a robotic platform with the direction of sensor/camera fixed relative to the Observer's heading. However, the same methods would work almost without change for panning/tilting cameras, Observers with multiple sensors, and more sophisticated observability models in the Observer's field of view [18]. It is also relatively easy to extend the approach to a centralized planner controlling several Evaders, but if all of them have separate targets, the cost of each iteration will increase accordingly. For a large group of Evaders, one will need to rely on distributed optimal control techniques or on Mean Field Game models, if these Evaders engage in selfish/independent trajectory planning.

Acknowledgements: The authors are indebted to Marc Gilles, whose algorithms and implementation for the stationary case served as a foundation for this work. The authors are also grateful to Casey Garner and Tristan Reynoso for their valuable help in implementation and testing during the summer REU-2018 program at Cornell University.

[1] Movies for all of these examples can be found at `https://eikonal-equation.github.io/TimeDependent_SEG`.

References

[1] M. Gilles and A. Vladimirsky, "Surveillance-evasion game under uncertainty," *submitted to Dynamic Games and Applications*, 2018. [Online]. Available: http://arxiv.org/abs/1812.10620

[2] J. Lewin and J. Breakwell, "The surveillance-evasion game of degree," *JOTA*, vol. 16, no. 3-4, pp. 339–353, 1975.

[3] J. Lewin and G. Olsder, "Conic surveillance evasion," *JOTA*, vol. 27, no. 1, pp. 107–125, 1979.

[4] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer, 2008.

[5] I. M. Mitchell and S. Sastry, "Continuous path planning with multiple constraints," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5. IEEE, 2003, pp. 5502–5507.

[6] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Struct. Optim.*, vol. 14, no. 1, pp. 63–69, 1997.

[7] A. Kumar and A. Vladimirsky, "An efficient method for multiobjective optimal control and optimal control subject to integral constraints," *Journal of Computational Mathematics*, pp. 517–551, 2010.

[8] Z. Clawson, X. Ding, B. Englot, T. A. Frewen, W. M. Sisson, and A. Vladimirsky, "A bi-criteria path planning algorithm for robotics applications," *arXiv preprint*, 2015. [Online]. Available: http://arxiv.org/abs/1511.01166

[9] A. Beck, *First-Order Methods in Optimization*. SIAM, 2017, vol. 25.

[10] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.

[11] S. Tijs, *Semi-infinite linear programs and semi-infinite matrix games*. Katholieke Universiteit Nijmegen. Mathematisch Instituut, 1976.

[12] T. Raghavan, "Zero-sum two-person games," *Handbook of game theory with economic applications*, vol. 2, pp. 735–768, 1994.

[13] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*. Philadelphia, PA: Society for Industrial and Applied Mathematics, dec 2013. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9781611973051

[14] G. Barles and P. E. Souganidis, "Convergence of approximation schemes for fully nonlinear second order equations," *Asymptotic analysis*, vol. 4, no. 3, pp. 271–283, 1991.

[15] Y.-H. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro, "Visibility and its dynamics in a PDE based implicit framework," *Journal of Computational Physics*, vol. 199, no. 1, pp. 260–290, 2004.

[16] C.-W. Shu, "High order numerical methods for time dependent Hamilton-Jacobi equations," in *Mathematics and computation in imaging science and information processing*. World Scientific, 2007, pp. 47–91.

[17] Z. Clawson, A. Chacon, and A. Vladimirsky, "Causal domain restriction for eikonal equations," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. A2478–A2505, 2014. [Online]. Available: https://doi.org/10.1137/130936531

[18] C. Wang, F. Qi, G. Shi, and X. Wang, "A sparse representation-based deployment method for optimizing the observation quality of camera networks," *Sensors*, vol. 13, no. 9, pp. 11 453–11 475, 2013.

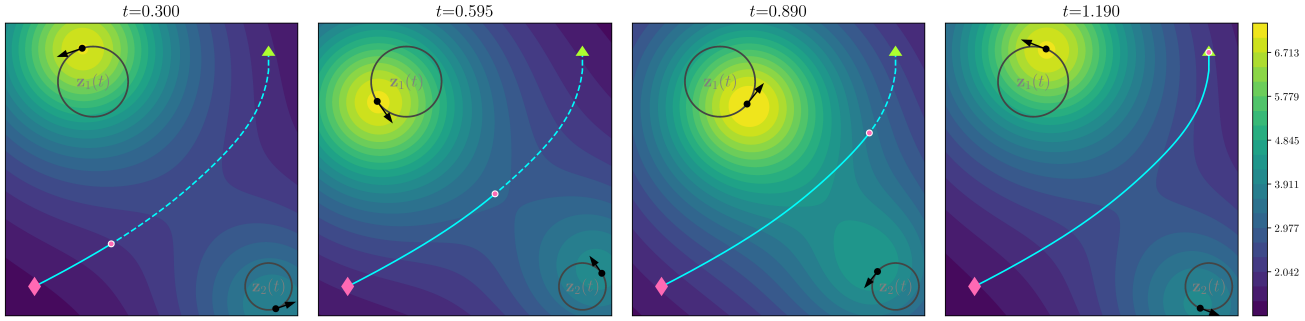


Fig. 5. Four snapshots of the Nash Equilibrium solution for Example 1. Observer's optimal policy is $\lambda_* = (0.67, 0.33)$. Observer's patrol trajectories (in gray) and Evader's unique λ_*-optimal trajectory (magenta) shown on top of the contour plots of $K^{\lambda_*}(\boldsymbol{x}, t)$. Observer's possible position on each patrol trajectory is shown by a black circle, with an arrow indicating Observer's current heading. Evader's starting, current, and terminal positions are shown by a magenta diamond, a magenta circle, and a green triangle respectively.

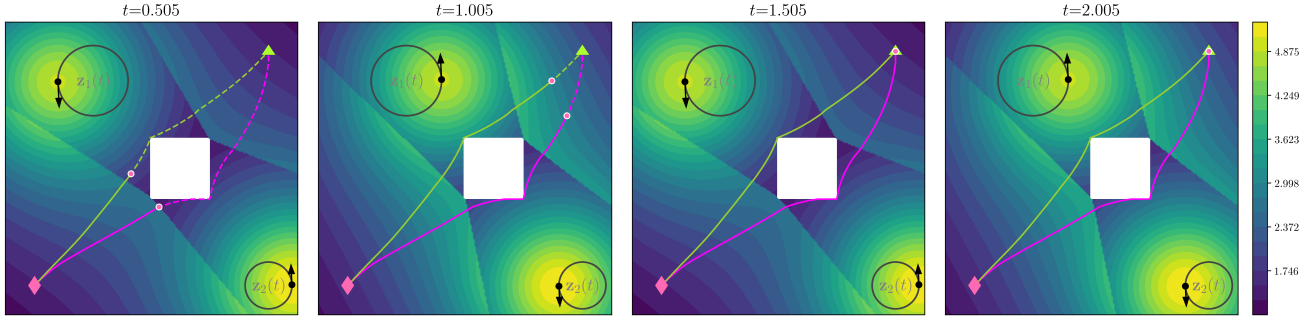


Fig. 6. Four snapshots of the Nash Equilibrium solution for Example 2. Observer's optimal policy is $\lambda_* = (0.48, 0.52)$, and Evader's optimal policy is $\theta_* = (0.691, 0.309)$, with two different λ_*-optimal trajectories shown, one in yellow and one in blue corresponding to the yellow and blue points in Fig. 3B.

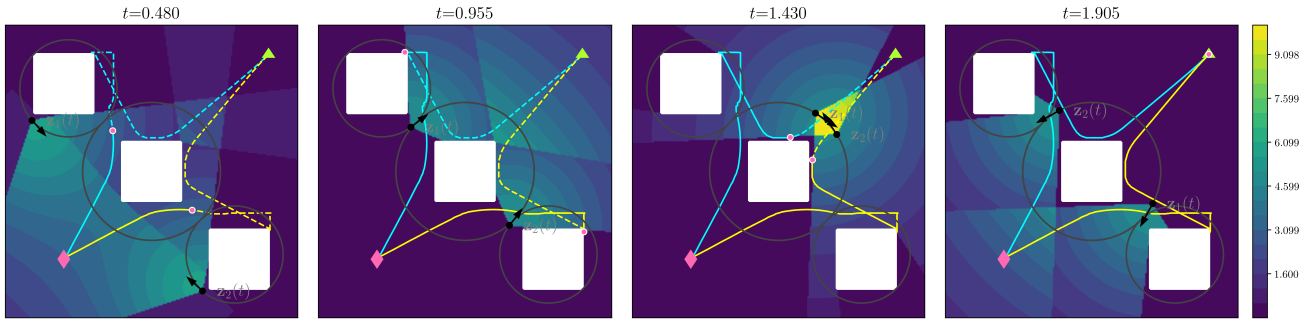


Fig. 7. Four snapshots of the Nash Equilibrium solution for Example 3. Observer's optimal policy is $\lambda_* = (0.5, 0.5)$, and Evader's optimal policy is $\theta_* = (0.5, 0.5)$, with two different λ_*-optimal trajectories shown in cyan and yellow.

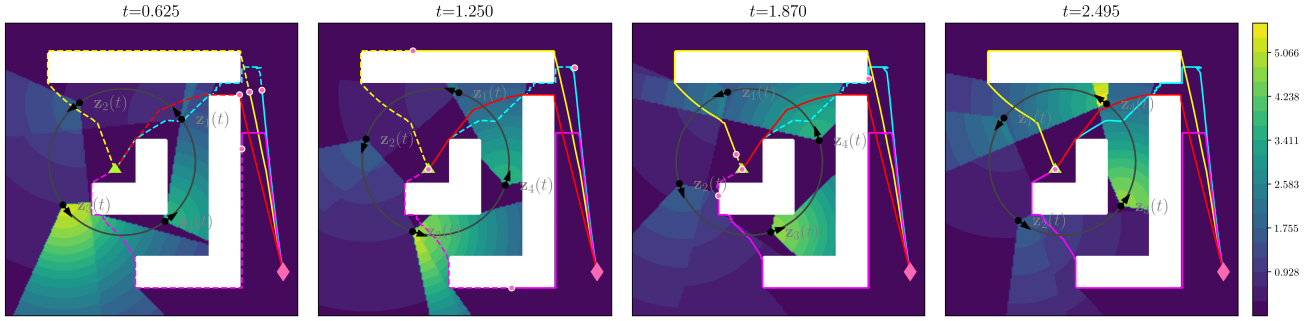


Fig. 8. Four snapshots of the Nash Equilibrium solution for Example 4. Observer's optimal policy is $\lambda_* = (0.077, 0.127, 0.452, 0.344)$, and Evader's optimal policy is $\theta_* = (0.592, 0.084, 0.145, 0.180)$, with four different λ_*-optimal trajectories shown in cyan, yellow, magenta, and red.