

Bounded-Rational Pursuit-Evasion Games

Yue Guan¹ Dipankar Maity² Christopher M. Kroninger³ Panagiotis Tsiotras⁴

Abstract—We present a framework that incorporates the principle of bounded rationality into dynamic stochastic pursuit-evasion games. The solution of a stochastic game is generally characterized by its (Nash) equilibria in feedback form, whose calculation may require extensive computational resources. In this paper, the agents are modeled as bounded rational entities with limited computational capabilities. We illustrate the proposed framework by applying it to a pursuit-evasion game between two aerial vehicles in a stochastic wind field. We show how such a game may be discretized and properly analyzed by casting it as an iterative sequence of finite-state Markov Decision Processes (MDPs). Leveraging tools and algorithms from the cognitive hierarchy theory (“level- k thinking”) we compute the solution of the ensuing discrete game, while taking into consideration the rationality level of each agent. We also present an online algorithm for each agent to infer its opponent rationality level.

I. INTRODUCTION

Pursuit-evasion games (PEGs) are a special class of dynamic games introduced in the 60s [1]–[4]. An extensive amount of literature exists on the topic; some notable examples include [3]–[11]. Most of these prior works [3], [5], [9], [10], [12], [13] are concerned with finding the equilibrium policy(s) of the game, which requires the assumption that all agents are rational [14]. The assumption of perfect rationality is, however, considered to be too strong and perhaps unrealistic for many applications [15], [16]. Furthermore, finding the equilibrium of a game is, in general, computationally expensive [17], [18]. To address these issues, we study pursuit-evasion games with agents that are not perfectly rational, but rather *bounded rational*.

The idea of bounded rationality [19] was introduced to address the fact that most agents do not adhere to the paradigm of maximum utility theory which assumes perfectly informed, rational agents. Several manifestations of bounded rationality exist in the literature, but in this paper we will adopt the one that makes use of the model of *cognitive hierarchies* (CH) [20] and *level- k thinking* [21]. Under this notion, the agent no longer seeks the (Nash) equilibrium, but instead seeks a best response to some (believed or inferred) policy of its opponent according to an assumed level- k responding structure.

Under the level- k framework, each agent is assumed to best respond to some given policy of its opponent. This leads to an iterative decision process for the agents (the Pursuer and the Evader in our case). The iterative process begins with “level-0” Pursuer and Evader, who have no prior assumptions about their opponents, or the structure of the game, and they follow some simple policies. A uniform random policy is commonly used as a choice for the level-0 policy. At higher levels, the level- k Pursuer and Evader presume that their opponent are of level- $(k-1)$, and they best respond accordingly based on this assumption. The notion of rationality is captured through the parameter k ; the higher the value of k , the higher the level of rationality of the agent, and the more this iterative process proceeds.

In order to capture the notion of bounded rationality [21], [22], each agent has a maximum level beyond which it cannot operate (different agents may have different maximum levels). Under this framework, the computational complexity of finding the optimal policy for a participating agent is significantly reduced [22], since the game can be cast as sequence of single-sided Markov-Decision-Process (MDP) problems [21], each of which can be solved efficiently [23].

In this work, we focus on PEGs involving bounded-rational agents in a stochastic environment (e.g., UAVs in a wind field). We start by considering a continuous-time PEG with continuous state-space, finite control/action set, perfect state observation, and stochastic unbounded noise. Building upon the Markov Chain approximation method [24], we then construct a series of finite-state, discrete-time PEGs as an approximation to the original continuous problem. We subsequently introduce the level- k thinking framework to solve these discrete PEGs in order to capture the bounded-rational properties of the agents.

Under this framework, each agent needs to know the rationality level of its opponent in order to select an optimal policy. To perform such an inference, we adopt an inferring algorithm based on the observed state trajectories [21]. The algorithm assumes that both agents’ rationality levels are fixed, and then lets the agents use a maximum likelihood estimator to update their beliefs regarding their opponent’s rationality level. We then generalize this inferring algorithm to the case where each agent adapts its level based on its belief regarding its opponent’s level.

The main contribution of this paper is the detailed construction of a comprehensive and implementable method to find solutions to continuous stochastic PEGs involving bounded rational agents. The proposed method first discretizes the continuous PEGs in a way that ensures the convergence of the optimal solution of the discretized game to that of the original continuous-state game. In

¹Yue Guan is a PhD Candidate with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Email: yguan44@gatech.edu

²Dipankar Maity is an Assistant Professor with the Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, Charlotte, NC, USA. Email: dmaity@uncc.edu

³Christopher Kroninger is a Research Scientist with Combat Capabilities Development Directorate, Army Research Laboratory, APG, MD, USA. Email: christopher.m.kroninger.civ@mail.mil

⁴Panagiotis Tsiotras is the David & Andrew Lewis Chair and Professor with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Email: tsiotras@gatech.edu.

addition, we show that if level- k thinking is applied to encode bounded rational decision-making of the agents, the discrete PEGs can be solved efficiently using value iteration of the corresponding MDP problems for each agent. We also present an inferring algorithm that updates the agents' beliefs regarding the rationality level of their opponent during the game. The effectiveness of the proposed approach is demonstrated using a two-dimensional two-agent PEG in a stochastic wind field, in which the two agents' rationalities are bounded by different maximum levels.

The rest of the paper is organized as follows. Section II presents the formulation of the pursuit-evasion game in a stochastic environment and applies the Markov Chain approximation method [24] to discretize the problem. Section III introduces the level- k thinking framework and Section IV presents algorithmic methods for solving this problem. The following Section V discusses the proposed inferring algorithms, and Section VI presents an illustrative example of a pursuit-evasion game in a stochastic wind field, with two bounded-rational agents. Finally, the paper is concluded in Section VII with a summary of the results and some possible extensions for future work.

II. PROBLEM FORMULATION

A. Continuous-Time PEG

Let us consider a two-agent pursuit-evasion differential game (PEG) in which the Pursuer and the Evader are indexed by $i = 1, 2$ respectively. In what follows, we will also use the index $-i$ to denote the opponent of agent i . As the names of the agents suggest, the *Pursuer* tries to *capture* the Evader, while the *Evader* tries to *evade* the Pursuer. For simplicity, we assume that the game evolves in a two-dimensional compact domain $C \subset \mathbb{R}^2$, and the position of an agent at time t is denoted by $p^i(t) = [p_x^i(t), p_y^i(t)]^\top \in \mathbb{R}^2$, for $i = 1, 2$. We define the state of the game as the joint positions of the two agents. Specifically, at time $t \geq 0$, the state of the game is represented as $s(t) = [p^1(t)^\top, p^2(t)^\top]^\top \in S = C \times C \subset \mathbb{R}^4$.

The action set for each agent is a finite collection of desired heading angles θ^1 and θ^2 . Specifically, we assume that agent i moves along the direction of its heading angle $\theta^i \in \Theta^i = \{0, \pi/2, \pi, 3\pi/2\}$ at a fixed speed $v^i > 0$. The joint action space for the agents is denoted as $\Theta = \Theta^1 \times \Theta^2$.

1) *Wind Field and Game Dynamics*: The game is played in the presence of a stochastic wind field, which will be modeled by a variant of a Wiener process. The dynamics of the agents are governed by the following stochastic differential equations (SDE)

$$dp^i = \begin{bmatrix} v^i \cos \theta^i + w_x(p^i) \\ v^i \sin \theta^i + w_y(p^i) \end{bmatrix} dt + dW_t^i(p^i), \quad i = 1, 2, \quad (1)$$

where $w_x(p^i)$ denotes the mean wind velocity in the x -direction at p^i , while w_y denotes the mean wind velocity in the y -direction. In (1) $dW_t^i(p^i)$ is a two-dimensional Wiener process with the following properties: a) $\mathbb{E}[dW_t^i(p)dW_t^i(p)^\top] = \sigma_w^2(p)I_2$; b) At any location

$p \in \mathbb{R}^2$, the Wiener increments are independent at different time instances, i.e., $\mathbb{E}[dW_t^i(p)dW_\tau^i(p)^\top] = 0$ for all $\tau \neq t$; c) At any time t , the wind disturbance is spatially uncorrelated i.e., $\mathbb{E}[dW_t^i(p)dW_t^i(q)^\top] = 0$ for $p \neq q$ ². For brevity, we denote $\theta = (\theta^1, \theta^2)$ and $dW_t^i = dW_t^i(p^i)$. We also denote $b^i(p^i, \theta^i) = [v^i \cos \theta^i + w_x(p^i), v^i \sin \theta^i + w_y(p^i)]^\top$.

The dynamics of the game can then be re-written as

$$ds = \begin{bmatrix} b^1(p^1, \theta^1) \\ b^2(p^2, \theta^2) \end{bmatrix} dt + \begin{bmatrix} dW_t^1 \\ dW_t^2 \end{bmatrix} = b(s, \theta) dt + dW_t, \quad (2)$$

The solution of the SDE in (2) is a stochastic process $\{s(t); t \geq 0\}$ such that

$$s(t) = s(0) + \int_0^t b(s(\tau), \theta(\tau)) d\tau + \int_0^t dW_\tau \triangleq \Phi(s(0), t, \theta^{[0,t]}, dW^{[0,t]}), \quad (3)$$

where the last term in equation (3) is treated as the usual Itô integral [26], $\theta^{[0,t]} = \{\theta(\tau); \tau \in [0, t]\}$ is the joint action history, and $dW^{[0,t]}$ is the realization of the wind field in the interval $[0, t]$.

2) *Terminal Conditions of the Game*: We define three terminating conditions (the boundaries) for the PEG.

- *Crash* corresponds to the scenario when an agent runs into obstacles or reaches the boundary of the set C . Let the closed set $O \subset C$ denote the obstacle-region in C . Then, agent i crashes into an obstacle or the boundary of the set ∂C , if at some time $t > 0$

$$\text{dist}(p^i(t), O \cup \partial C) = 0, \quad (4)$$

where the distance of a point p from a set M is $\text{dist}(p, M) \triangleq \inf_{m \in M} \|p - m\|_2$. We define the two crash boundaries and their union in the state space as

$$\begin{aligned} \partial S_{\text{crsh}}^i &= \{s \in S : \text{dist}(p^i(t), O \cup \partial C) = 0\}, \quad i = 1, 2 \\ \partial S_{\text{crsh}} &= \partial S_{\text{crsh}}^1 \cup \partial S_{\text{crsh}}^2. \end{aligned}$$

- *Capture* is considered successful when the distance between the two agents at some time instance $t > 0$ is less than a prescribed positive value ρ , while neither of the agent crashes. The following two conditions are simultaneously satisfied for a successful capture:

$$\begin{aligned} \text{dist}(p^1(t), p^2(t)) &\leq \rho, \\ \text{dist}(p^i(t), \partial S_{\text{crsh}}) &> 0, \quad \text{for } i = 1, 2. \end{aligned} \quad (5)$$

The capture condition defines a boundary in the state space as follows:

$$\partial S_{\text{cap}} = \{s \in S : \text{dist}(p^1(t), p^2(t)) \leq \rho\} \setminus \partial S_{\text{crsh}}.$$

- *Evasion* is successful when the Evader enters a closed, non-empty evading region $E \subset C$ with no capture nor crash. Similarly, the boundary for evasion is defined as

$$\partial S_{\text{evs}} = \{s \in S : \text{dist}(p^2(t), E) = 0\} \setminus (\partial S_{\text{crsh}} \cup \partial S_{\text{cap}}).$$

²In case the wind disturbance is spatially correlated, then one has $\mathbb{E}[dW_t^i(p)dW_t^i(q)^\top] = h(p, q)$, where $h(p, p) = \sigma_w^2(p)I_2$, and $h(p, q) \geq 0$ for $p \neq q$. A popular choice for random field models in geostatistics is $h(p, q) = \exp(-\gamma\|p - q\|)I_2$ for some $\gamma > 0$ [25].

¹For simplicity, in this paper we will assume $\sigma_w(p) = \sigma_w$ for all p .

The boundary of the state space ∂S is defined by the three (disjoint) terminal conditions:

$$\partial S = \partial S_{\text{crsh}} \cup \partial S_{\text{cap}} \cup \partial S_{\text{evs}}.$$

When the process $\{s(t); t \geq 0\}$ hits the boundary ∂S , the game terminates and the outcome of the game is determined by the part of the boundaries $(\partial S_{\text{crsh}}, \partial S_{\text{cap}}, \partial S_{\text{evs}})$ reached.

To this end, we define the interior of the state space $S = C \times C$ as $S^\circ = S \setminus \partial S$. Without loss of generality, we assume that the game starts at some initial state $s(0) \in S^\circ$.

3) *Admissible Policies and Rewards*: An admissible policy for an agent at time t is a measurable mapping from the observation history $(\{s(\tau); \tau \in [0, t]\})$ to a probability distribution over its action set. It is well known that, with full state information, the best Markov policy performs as well as the best admissible policy [26], [27]. A Markov policy depends only on the current state $s(t)$ and not on the entire history. The Markov policy of agent i is represented by the mapping $\mu^i(\cdot, \cdot) : S^\circ \times \Theta^i \rightarrow [0, 1]$, and $\mu^i(s, \theta^i)$ denotes the probability of agent i choosing action $\theta^i \in \Theta^i$ at state s . Consequently, for all $s \in S^\circ$, $\sum_{\theta^i \in \Theta^i} \mu^i(s, \theta^i) = 1$. The set of all such policies for agent i is denoted by Π^i . We denote the joint policy as μ and the set of joint policies as Π .

Define the *first exit time* T_μ under the joint policy μ as

$$T_\mu = \inf \left\{ t : s(t) \in \partial S, s(t) = \Phi(s(0), t, \theta^{[0,t]}, W^{[0,t]}) \right\}$$

where the control sequence $\theta^{[0,t]}$ is a realization under the joint policy μ . Therefore, T_μ is a random variable that reflects the first time a successful capture, or evasion, or crash occurs. We define the terminal reward for the Pursuer at the terminal state $s(T_\mu) \in \partial S$ as

$$g^1(s(T_\mu)) = \begin{cases} 1, & \text{if } s(T_\mu) \in \partial S_{\text{cap}}, \\ -1, & \text{if } s(T_\mu) \in \partial S_{\text{evs}}, \\ -1, & \text{if } s(T_\mu) \in \partial S_{\text{crsh}}^1 \setminus \partial S_{\text{crsh}}^2, \\ 1, & \text{if } s(T_\mu) \in \partial S_{\text{crsh}}^2 \setminus \partial S_{\text{crsh}}^1, \\ 0, & \text{if } s(T_\mu) \in \partial S_{\text{crsh}}^1 \cap \partial S_{\text{crsh}}^2. \end{cases} \quad (6)$$

In addition, we assume a zero-sum game formulation, and we define the terminal reward for the Evader by

$$g^2(s(T_\mu)) = -g^1(s(T_\mu)). \quad (7)$$

This work formulates the PEG as a *game of type* [1], where we use the win-rate as the performance index of the agents, instead of other metrics such as capture time. As a result, only the terminal reward is included, and no running reward is introduced. An extension to include running reward (such as fuel cost or time) into the formulation requires a minor modification and will be addressed elsewhere.

We use J_μ^i to denote the expected reward-to-go function for agent i under the joint policy μ . Starting from initial state s_0 , the expected reward is given by

$$J_\mu^i(s_0) = J_{\mu^1, \mu^2}^i(s_0) = \mathbb{E}_{s_0, \mu} \left[g^i(s(T_\mu)) \right], \quad i = 1, 2, \quad (8)$$

where the conditional expectation is given by

$$\mathbb{E}_{s_0, \mu} [\cdot] = \mathbb{E} \left[\cdot \mid s(t) = \Phi(s_0, t, \theta^{[0,t]}, W^{[0,t]}) \mid \mu \right].$$

In the PEG setting, $g^i(\cdot)$ incorporates the reward for either a successful capture, evasion, or crash. Both agents try to maximize their own expected rewards by choosing their respective policies μ^i . As the game is zero-sum, by defining $J_\mu = J_\mu^1$, the optimization problem can be written as

$$\sup_{\mu^1 \in \Pi^1} \inf_{\mu^2 \in \Pi^2} J_{\mu^1, \mu^2}(s_0) = \inf_{\mu^2 \in \Pi^2} \sup_{\mu^1 \in \Pi^1} J_{\mu^1, \mu^2}(s_0). \quad (9)$$

In the next section, we discuss the discretization of this PEG such that the optimal solution of the discretized PEG converges to the optimal solution of the continuous PEG as the discretization step being reduced to zero.

B. Markov Chain Approximation

In this section, we introduce the *Markov Chain Approximation Method* (MCAM) [24], which we use to discretize the controlled stochastic process in (2) into a discrete-state, discrete-time competitive Markov Decision Process (cMDP) [23]. In the specific case of a pursuit-evasion game, we refer to the discretized process as the discrete Pursuit-Evasion Game (dPEG).

A discretized two-agent zero-sum cMDP is a tuple $\mathcal{M}_h = \langle S_h, \Theta^1, \Theta^2, P_h, G_h \rangle$ where h is the discretization size and S_h is the discretized finite state space. We let Θ^i be the same action set as the original problem, and $P_h(\cdot | \cdot, \cdot, \cdot) : S_h \times S_h \times \Theta^1 \times \Theta^2 \rightarrow [0, 1]$ denotes the transition probabilities satisfying $\sum_{s' \in S_h} P_h(s' | s, \theta^1, \theta^2) = 1$ for all $s \in S_h$ and all $\theta^i \in \Theta^i$. The mapping $G_h : S_h \rightarrow \mathbb{R}$ is the terminal reward function.

Suppose we start from an initial state $s_0 \in S^\circ$. At time $n \geq 0$, the current state is s_n , and we apply a joint control $\theta_n \in \Theta$ to visit the next state s_{n+1} according to the transition function P . The result is a controlled Markov Chain $\{s_n; n \in \mathbb{N}_0\}$ [24]. Notice that such a chain depends on the choice of the initial state s_0 and the control sequences $\{\theta_n^i : n \in \mathbb{N}_0\}$. Up to any time $N > 0$, the observed chain is denoted as $s^{[0,N]} = \{s_0, \dots, s_N\}$ and is generated by the joint action sequence $\theta^{[0,N-1]} = \{\theta_0, \dots, \theta_{N-1}\}$. The chain $s^{[0,N]}$ and the joint action sequence $\theta^{[0,N-1]}$ are also referred to as the *trajectory* of $\mathcal{M}$ under the given sequence of controls and the initial state.

Given a PEG defined by the dynamics (2) with the terminal rewards as in (6), we adopt a spatio-temporal discretization scheme to reduce the problem to a finite-state cMDP. We start by discretizing the compact region C with a grid-size $h > 0$ (see Figure 1). The discretization size h is a user-defined parameter that determines the resolution of the discretization. Let us define the set $C_h = \{c_{h,1}, \dots, c_{h,N}\}$ denoting the elementary squares of the grid, so that we can always properly cover the compact set C with C_h , namely, $C \subseteq C_h$. Due to the compactness of C , the cardinality of C_h is always finite for any $h > 0$. Note that each $c_h \in C_h$ represents a unique square cell of size h in $\mathbb{R}^2$ (including boundary). Let c_h° be the interior of the square cell c_h . We extend the notion of obstacles and boundaries from C to C_h in the following way. The cell c_h is labeled as an *obstacle cell* if and only if $c_h^\circ \cap O \neq \emptyset$. The cell c_h is labeled as a *boundary cell* if and only if $c_h^\circ \cap \partial C \neq \emptyset$. Similarly, if

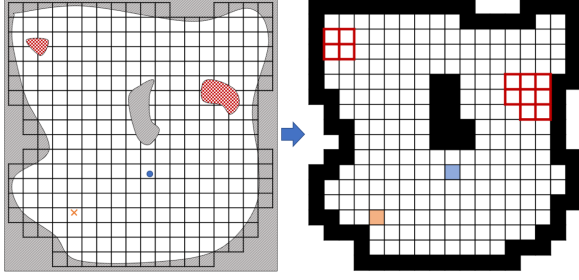


Fig. 1: An example of a discretization of the two-dimensional space. Left is the continuous space C , where the shaded red and gray areas are the evasion regions and the obstacles respectively. The orange and blue markers represent the positions of the Pursuer and the Evader. Right is the discretized space C_h , where the red and black cells correspond to the evasion cells and the obstacle cells. The orange and blue cells are the cells in which the agents are located.

$c_h^o \cap E \neq \emptyset$, then c_h is an *evasion cell*. We say that agent i is in cell c_h , if $p^i(t) \in c_h^o$. If an agent is located on the common boundary of two cells, it is assigned to one of the cells based on a prior assignment rule. Having designed C_h , we define the discretization of S as $S_h = C_h \times C_h$. Each $s_h \in S_h$ denotes a hyper-cube of side length h in $\mathbb{R}^4$. We denote the interior of the hyper-cube s_h as s_h^o . We now define

$$\begin{aligned} \partial S_{h,\text{crsh}} &= \{s_h \in S_h : s_h^o \cap \partial S_{\text{crsh}}^i \neq \emptyset\}, \\ \partial S_{h,\text{crsh}} &= \partial S_{h,\text{crsh}}^1 \cup \partial S_{h,\text{crsh}}^2, \\ \partial S_{h,\text{cap}} &= \{s_h \in S_h \setminus \partial S_{h,\text{crsh}} : s_h^o \cap \partial S_{\text{cap}} \neq \emptyset\}, \\ \partial S_{h,\text{evs}} &= \{s_h \in S_h \setminus (\partial S_{h,\text{crsh}} \cup \partial S_{h,\text{cap}}) : s_h^o \cap \partial S_{\text{evs}} \neq \emptyset\}, \\ \text{and finally } \partial S_h &= \partial S_{h,\text{crsh}} \cup \partial S_{h,\text{cap}} \cup \partial S_{h,\text{evs}}. \end{aligned} \quad (10)$$

The Markov Chain approximation method approximates the continuous zero-sum PEG using a sequence of cMDPs $\{\mathcal{M}_h\}$, where $\mathcal{M}_h = \langle S_h, \Theta^1, \Theta^2, P_h, G_h \rangle$ and $S_h = C_h \times C_h$. For each discretization $h > 0$, we have $s_h^{[0,N]} = \{s_h^n; n \leq N\}$ as a controlled Markov Chain on $\mathcal{M}_h$, which terminates when it hits ∂S_h .

We associate each state $s \in S^o$ in the original continuous space with a non-negative interpolation interval $\Delta t_h(s)$, known as the *holding time* [24]. For each elementary hyper-cube $s_h \in S_h$, the centroid of s_h is denoted as $\alpha(s_h)$. Let us also define $\Delta s_h^n = \alpha(s_h^{n+1}) - \alpha(s_h^n)$. Since the mapping α is bijective, with a slight abuse of the notation, we use s_h to denote both the hyper-cube and its centroid. For brevity, we denote Δt_h^n to be the holding time at state s_h^n , i.e., $\Delta t_h(s_h^n)$, and define $t_h^n = \sum_{i=0}^{n-1} \Delta t_h^i$ for $n \geq 1$ and $t_h^0 = 0$. The discretized terminal cost G_h is defined similar to (6):

$$G_h(s_h) = \begin{cases} 1, & \text{if } s_h \in \partial S_{h,\text{cap}}, \\ -1, & \text{if } s_h \in \partial S_{h,\text{evs}}, \\ -1, & \text{if } s_h \in \partial S_{h,\text{crsh}}^1 \setminus \partial S_{h,\text{crsh}}^2, \\ 1, & \text{if } s_h \in \partial S_{h,\text{crsh}}^2 \setminus \partial S_{h,\text{crsh}}^1, \\ 0, & \text{if } s_h \in \partial S_{h,\text{crsh}}^1 \cap \partial S_{h,\text{crsh}}^2. \end{cases} \quad (11)$$

C. Discrete Transition P_h

Let Ω_h be the sample space of $\mathcal{M}_h$, let $\theta_h^n = (\theta_h^{1,n}, \theta_h^{2,n})$ be the joint action at time n . The holding times Δt_h^n and the transition probabilities P_h are chosen to satisfy the *local*

consistency property [24], with respect to (2), which are given by the following conditions.

- 1) For all $s_h \in S_h$, $\lim_{h \rightarrow 0^+} \Delta t_h(s_h) = 0$.
- 2) For all $s_h \in S_h$ and all joint controls $\theta \in \Theta$:

$$\begin{aligned} \lim_{h \rightarrow 0^+} \frac{\mathbb{E}_{P_h}[\Delta s_h^n | s_h^n = s, \theta_h^n = \theta]}{\Delta t_h(s)} &= b(s, \theta) \\ \lim_{h \rightarrow 0^+} \frac{\text{Cov}_{P_h}[\Delta s_h^n | s_h^n = s, \theta_h^n = \theta]}{\Delta t_h(s)} &= \sigma_w \sigma_w^\top \\ \lim_{h \rightarrow 0^+} \sup_{N \in \mathbb{N}_0, s_h^{[0,N]} \in \Omega_h} \|\Delta s_h^n\|_2 &= 0. \end{aligned}$$

As the chain $\{s_h^n; n \in \mathbb{N}_0\}$ is a discrete-time process, we use an *approximate continuous-time interpolation* [24] to approximate the continuous-time process in (2). We define the continuous-time interpolation $s_h(\cdot)$ of the chain $\{s_h^n\}$ and the continuous-time interpolation $\theta_h(\cdot)$ of the action sequence $\{\theta_h^n\}$ under the holding time function Δt_h as follows: $s_h(\tau) = s_h^n$, and $\theta_h(\tau) = \theta_h^n$ for all $\tau \in [t_h^n, t_h^{n+1})$.

There are multiple ways to construct such a locally consistent Markov Chain. For our specific problem, we follow the construction found in [24] that splits the control inputs of agent 1 (Pursuer) and agent 2 (Evader). Specifically, given the dynamics of the PEG as in (2), we rewrite the drift term as

$$b(s, \theta) = [b_1(s, \theta), b_2(s, \theta), b_3(s, \theta), b_4(s, \theta)]^\top.$$

We then define the quantity $Q_h(s_h)$ as

$$Q_h(s_h) = h \max_{\theta \in \Theta} \left\{ \sum_{i=1}^4 |b_i(s_h, \theta)| \right\} + 4 \sigma_w^2,$$

and define the interpolation interval as

$$\Delta t_h(s_h) = \frac{h^2}{Q_h(s_h)}. \quad (12)$$

Notice that $\Delta t_h(s_h) \rightarrow 0$ as $h \rightarrow 0$, for all $s_h \in S_h$. The transition probabilities of the cMDP that approximates the original PEG can be calculated via

$$\begin{aligned} P_h(s_h \pm h e_j | s_h, \theta) &= \left(\frac{\sigma_w^2}{2} + h b_j^\pm(s_h) \right) / Q_h(s_h), \\ P_h(s_h | s_h, \theta) &= 1 - \sum_{j=1}^4 P_h(s_h \pm h e_j | s_h, \theta), \\ P_h(s'_h | s_h, \theta) &= 0, \text{ if } s'_h \neq s_h \pm h e_j \text{ and } s'_h \neq s_h, \end{aligned} \quad (13)$$

where $b^+ = \max\{b, 0\}$ and $b^- = \max\{-b, 0\}$, and e_i are the unit vectors in $\mathbb{R}^4$ (e.g. $e_3 = [0 \ 0 \ 1 \ 0]^\top$). The next state $s_h \pm h e_j$ are the state s'_h such that $\alpha(s'_h) = \alpha(s_h) \pm h e_j$. One can verify that the transition probabilities defined in (13) are locally consistent with (2).

Under this discretization scheme, any advantage an agent have in its speed is translated into a probabilistic advantage, as in (13). Specifically, P_h depends on v^i through $b_i(s_h, \theta)$ and one can observe that with the same wind field present, the agent with a higher speed is more likely to end up in the cell that it intends to visit.

As stated in [24], the local consistency property implies the convergence of the continuous-time interpolations of the

trajectories of the controlled Markov Chain to the trajectories of the stochastic dynamical system in (2). It also guarantees the convergence of the optimal reward-to-go functions of the discrete cMDPs to that of the original problem.

Finally, we define a (mixed) policy for $\mathcal{M}_h$ as a mapping $\mu_h^i : S_h \times \Theta^i \rightarrow [0, 1]$, such that $\sum_{\theta^i} \mu_h^i(s_h, \theta^i) = 1$. The set of all admissible policies is denoted by Π_h^i . To this end, given a joint policy μ_h and the initial state $s_h \in S_h$, the expected reward-to-go from s_h due to μ_h is

$$J_{h, \mu_h}(s_h) = J_{h, \mu_h^1, \mu_h^2}(s_h) = \mathbb{E}_{P_h, \mu_h} [G_h(s_h^{I_h})], \quad (14)$$

where the $\mathbb{E}_{P_h, \mu_h}$ denotes the conditional expectation under P_h given $s_h^0 = s_h$, and $\{s_h^n; n \in \mathbb{N}_0\}$ is the sequence of states of the controlled Markov Chain under the policy μ_h , and the terminal time is $I_h = \inf\{n : s_h^n \in \partial S_h\}$ ³.

In later sections, we show that if the opponent's policy is given, an optimal policy exists for the discretized cMDP and we also discuss the methods to find it.

III. LEVEL-K THINKING

Under the framework of level- k thinking, an agent best responds to a given level- $(k-1)$ policy of its opponent. Consider the cMDP $\mathcal{M}_h$ constructed in Section II. Suppose that agent i is given the policy of its opponent μ_h^{-i} , one can then marginalize the transition in (13) and the reward-to-go in (14) using μ_h^{-i} . The cMDP $\mathcal{M}_h$ is then reduced to a standard MDP optimizing with respect to the policy μ_h^i . In what follows, we use the superscript $k \in \{0, 1, 2, \dots\}$ within a parenthesis to denote the rationality level. We also use s to denote the state $s_h \in S_h$ for brevity.

Given the level- $(k-1)$ policy of the opponent $\mu^{-i, (k-1)}$, we can define a one-sided MDP for the level- k agent i via $\mathcal{M}_h^{i, (k)} = \langle S_h, \Theta^i, P_h^{i, (k)}, G_h \rangle$, where the marginalized transition $P_h^{i, (k)}$ for each action θ^i is given by

$$P_h^{i, (k)}(s' | s, \theta^i) \triangleq \sum_{\theta^{-i} \in \Theta^{-i}} P_h(s' | s, \theta^i, \theta^{-i}) \mu^{-i, (k-1)}(s, \theta^{-i}).$$

Then the optimal value for agent k at level- k can be computed from the fixed point of the Bellman equation [23]:

$$V^{i, (k)}(s) = \text{ext}_{\mu^{i, (k)}} \left\{ \sum_{s' \in S} P^{i, (k)}(s' | s, \mu^{i, (k)}) V^{i, (k)}(s') \right\}, \quad (15)$$

with the boundary condition $V^{i, (k)}(s) = G_h(s)$ for $s \in \partial S_h$, and the controlled transition

$$P^{i, (k)}(s' | s, \mu^{i, (k)}) \triangleq \sum_{\theta^i \in \Theta^i} P^{i, (k)}(s' | s, \theta^i) \mu^{i, (k)}(s, \theta^i).$$

The ext operator in (15) corresponds to a sup when the Pursuer optimizes and to an inf for the Evader.

It has been shown that there exist at least one pure Markov policy [23] that solves (15). That is, there is an optimal policy $\mu^{i, (k^*)}$, such that at each state $s \in S_h$, there exists $\theta^{i*} \in \Theta^i$ and $\mu^{i, (k^*)}(s, \theta^{i*}) = 1$.

³The terminal time is well-defined, i.e. $I_h < \infty$ w.p.1, since the generated cMDP does not have any recurrent states, assuming $\sigma_w > 0$.

To initialize the level- k policy construction, we choose the level-0 policies $\mu^{i, (0)}$ to be

$$\mu^{i, (0)}(\theta^i, s) = \frac{1}{|\Theta^i|} \quad \text{for all } \theta^i \in \Theta^i, s \in S, \quad (16)$$

where $|\Theta^i|$ denotes the cardinality of the action set Θ^i . This level-0 policy places a uniform distribution over the action set Θ^i . Such policy reflects the idea that a level-0 agent is most naive and does not perform any optimization⁴.

The level-1 agents are then given the level-0 policies, and they calculate their best response to their opponent's level-0 policy via (15). Similarly, the level-2 agents compute their best response to the given level-1 policies via (15). This process of building policies level-after-level continues.

In practice, an agent with limited computational resources (e.g., a human) can continue this process only up to a certain level $k_{\max}^i$. Clearly, the best response to a level- k policy $\mu^{-i, (k)}$ is, by definition, $\mu^{i, (k+1)}$ and not necessarily the Nash policy $\mu^{i, *}$. We want to emphasize the fact that the Nash policy $\mu^{i, *}$, even though robust, is not always optimal given the opponent's policy. If agent i is certain that its opponent is unable to compute the Nash policy $\mu^{-i, *}$ and plays policy $\mu^{-i, (k)}$ for some $k \in \{0, \dots, k_{\max}^i\}$ instead, then there is an incentive for agent i to play the one-level-higher policy $\mu^{i, (k+1)}$ other than Nash to maximize its reward. As a result, both agents naturally have the tendency to iteratively select a policy that is one level higher than the opponents', until either of them reaches its maximum rationality level $k_{\max}^i$.

As discussed above, the level-wise best response structure in (15) can be equivalently represented by

$$\mu^{i, (k+1)} \in \text{BestResponse}(\mu^{-i, (k)}). \quad (17)$$

We present the following Lemma regarding the convergence of level- k policies to the Nash equilibrium.

Lemma III.1. *Given a two-agent cMDP solved under the level- k thinking framework, if $\mu^{i, (K+2)} = \mu^{i, (K)}$ for an agent at some finite level K , then the two agents reach a Nash equilibrium by applying the policy pair $(\mu^{i, (K)}, \mu^{-i, (K+1)})$.*

Proof. By the construction of level- (k) policies in (15), we know that

$$\begin{aligned} \mu^{i, (K+2)} &\in \text{BestResponse}(\mu^{-i, (K+1)}), \\ \mu^{-i, (K+1)} &\in \text{BestResponse}(\mu^{i, (K)}). \end{aligned}$$

From the assumption $\mu^{i, (K+2)} = \mu^{i, (K)}$, we have the following fixed point property:

$$\begin{aligned} \mu^{i, (K)} &\in \text{BestResponse}(\mu^{-i, (K+1)}), \\ \mu^{-i, (K+1)} &\in \text{BestResponse}(\mu^{i, (K)}). \end{aligned}$$

The policy pair $(\mu^{i, (K)}, \mu^{-i, (K+1)})$ then corresponds to a Nash equilibrium, by definition. $\square$

Under the level- k framework, we have reduced a cMDP $\mathcal{M}_h$ into a series of one-sided MDPs $\{\mathcal{M}_h^{i, (k)}\}$. For each

⁴One may choose different level-0 policies, for example, a simple Pursuer policy that keeps the heading towards the Evader. Different level-0 policies, however, may result in different level- k policies.

of the MDPs, the optimization is only over a single agent's policy and hence plenty of efficient numerical algorithms exist to solve for the optimal policy [28]. In the following section, we use the well-known value iteration to compute the best responses at each level.

Remark 1. Suppose no maximum level constraint is imposed. Then the level- k iterative process terminates at some level K if the condition $\mu^{i,(K+2)} = \mu^{i,(K)}$ is satisfied for some $i \in \{1, 2\}$.

Remark 2. Lemma III.1 ensures that any converging level- k policy is a Nash Equilibrium. However, in general, there is no guarantee that the previous iterative level- k construction will converge. Furthermore, in case of multiple Nash equilibria for a general-sum game, one cannot comment on which equilibrium the level- k policy will converge to, as it may depend on the selection of the level-0 policies.

IV. ALGORITHMIC SOLUTION

Based on the results in [23], [28], the optimal policy of $\mathcal{M}_h^{i,(k)}$ can be found by value iteration. We use the subscript m to denote the iteration index. Following this notation, the value iterations are given by

$$V_{m+1}^{i,(k)}(s) = \text{ext}_{\theta^i \in \Theta^i} \left\{ \sum_{s' \in S} P^{i,(k)}(s'|s, \theta^i) V_m^{i,(k)}(s') \right\}, \quad s \in S_h^o,$$

$$V_{m+1}^{i,(k)}(s) = G_h(s), \quad s \in \partial S_h.$$

As the action space is assumed to be finite, solving the value iteration at step $m+1$ becomes a problem of finding the heading angle at each non-terminal state s that maximizes the expected future rewards, given the value function $V_m^{i,(k)}$ from the previous iteration m .

Notice that the previous iteration in does not have a discount factor due to the absence of a running reward in (8). The presence of a discount factor is essential to prove the convergence of value iteration in infinite horizon MDPs [29]. However, with the transition probability of the controlled Markov Chain obtained from the Markov Chain approximation method, we have that no recurrent states exists for our problem aside from the absorbing boundary states (the terminal states) defined in (10). It can then be shown that for this specific case the value iteration converges as the iteration step $m \rightarrow \infty$. However, in practice, this iteration is truncated after a finite but large number of iterations, so that the truncated value function is close to the actual value function.

V. INFERRING THE OPPONENT'S LEVEL

In Section III, we argued that agent i can play its best policy if it knows its opponent's level k^{-i} , and is able to compute the best response $\mu^{i,(k^{-i}+1)}$. In real world situations, it is not always possible to know the opponent's level exactly, and hence, the agent must infer the level of its opponent based on the trajectory of the system. Below we propose an online algorithm to estimate the opponent's rationality level k^{-i} based on the state trajectory $s^{[0,N]}$. The algorithm uses a maximum likelihood inferring algorithm utilizing the idea in [30].

A. Fixed Level Model

We first make the assumption that both agents set their levels at the beginning of the game and do not change their levels during the game. Suppose the maximum rationality level of agent i is $k_{\max}^i$. Based on the discussion in the Section III, agent i should also have access to its opponent's policies till level $(k_{\max}^i - 1)$. We denote the set of the opponent's level that agent i has access to as $\mathcal{K}^i = \{0, 1, \dots, k_{\max}^i - 1\}$. Given that agent i plays at level k^i and the game starts from the initial state s_0 , suppose that agent i observes a trajectory $s^{[0,N]} = \{s_0, \dots, s_N\}$. Assuming that the opponent plays at level $k^{-i} \in \mathcal{K}^i$, the probability of observing the specific trajectory is given by

$$\mathbb{P}(s^{[0,N]} | k^i, k^{-i}) = \mathbb{P}(s^{[0,N]} | \mu^{i,(k^i)}, \mu^{-i,(k^{-i})})$$

$$= \prod_{n=0}^{N-1} P_h(s_{n+1} | s_n, \mu^{i,(k^i)}, \mu^{-i,(k^{-i})}). \quad (18)$$

The estimate of the opponent's level can then be selected as the maximum likelihood estimator

$$\hat{k}^{-i} \in \underset{k^{-i} \in \mathcal{K}^i}{\operatorname{argmax}} \mathbb{P}(s^{[0,N]} | k^{-i}, k^i). \quad (19)$$

One can easily extend this algorithm to infer the opponent's level from a moving observation window of a certain length.

B. Dynamic Level Model

The Fixed Level Model does not allow agents to change their rationality levels as the game evolves. To make the interactions between the two agents more realistic, we now let them adapt their levels based on their observations. To this end, suppose the opponent's level k^{-i} remains constant over time. We let agent i adapt its rationality level based on its belief regarding its opponent's level. We also assume that the agent always plays one level higher than the estimated level of its opponent, without exceeding its maximum level $k_{\max}^i$. Let us pick a observation window of length $w+1$ and let the current time step to be t_N . Denote the observed trajectory from t_{N-w} to t_N as $s^{[N-w,N]} = \{s_{N-w}, \dots, s_N\}$ and denote the levels played by agent i over this period as $k^{i,[N-w,N]} = \{k_{N-w}^i, \dots, k_N^i\}$. The probability of observing this specific trajectory $s^{[N-w,N]}$ with a fixed k^{-i} is

$$\mathbb{P}(s^{[N-w,N]} | k^{i,[N-w,N]}, k^{-i})$$

$$= \prod_{n=N-w}^{N-1} P_h(s_{n+1} | s_n, \mu^{i,(k_n^i)}, \mu^{-i,(k^{-i})}), \quad (20)$$

and we use the maximum likelihood estimator to infer the opponent's level at t_N

$$\hat{k}_N^{-i} \in \underset{k^{-i} \in \mathcal{K}^i}{\operatorname{argmax}} \mathbb{P}(s^{[N-w,N]} | k^{-i}, k^{i,[N-w,N]}), \quad (21)$$

Using this estimator, at the next time step t_{N+1} , agent i would play at level $\min\{\hat{k}_N^{-i} + 1, k_{\max}^i\}$ as an adaptation based on the observed trajectory of the system.

VI. NUMERICAL EXAMPLE

In this section, we illustrate the proposed framework via a numerical example. We consider a two-agent pursuit-evasion game in a continuous wind field and discretize it using an 18×18 grid. The mean wind velocity is generated randomly, while the wind covariance is set to $\sigma_w = 0.4$ with no spatial correlation. The starting positions of the agents, the evasion regions and the obstacles are shown in Fig. 2. We use the MCAM to discretize the state space to obtain the grid to the right. The terminal reward G_h is assigned according to (6). Both agents have the same speed $v^1 = v^2 = 1$ and the same action set $\Theta^1 = \Theta^2 = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.

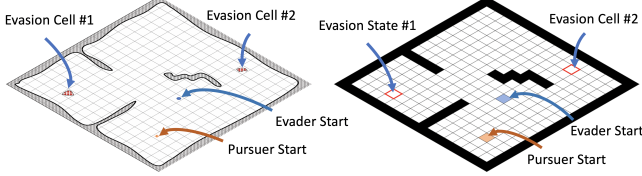


Fig. 2: The continuous space (left) and the discretized grid (right).

We assume that the level-0 agents pick their headings such that they will not cause an immediate crash, while choosing their action uniformly randomly. We construct the level- k thinking hierarchy as presented in Section III. We then simulate the computed policies at some selected level pairs over 1500 games. To illustrate the results, we use two tables to present the win rate of the selected scenarios.

In Tables I and II, we fix one agent to be level-2 while varying the level of the other agent. In both tables, the highest winning rates are attained at level 3. This result is expected, since level 3 policies are by definition the best responses to the level-2 opponents. It can also be observed that after level 5 the performance only varies slightly, since the policies at high levels do not differ much from each other.

TABLE I: Pursuer win percentages against level-2 Evader.

Pursuer Level	1	2	3	4	5	6
Pursuer Wins	48.5	47.6	51.7	50.9	51.2	51.1
Capture	36.4	38.2	45.3	43.1	42.7	42.9

TABLE II: Evader win percentages against level-2 Pursuer.

Evader Level	1	2	3	4	5	6
Evader Wins	47.5	52.4	53.8	52.9	53.2	53.1
Due to Evasion	35.6	42.2	45.3	44.5	44.7	44.2

In Fig. 3 two sample trajectories are shown. The first trajectory in (a) depicts a level-3 Evader against a level-2 Pursuer. One observes the “deceiving” behavior of the Evader: it first moves towards evasion cell #2 (on the right), which tricks the level-2 Pursuer to also go right and take the shorter route beneath the obstacles for defending the evasion cell #2. The Evader then suddenly turns left and goes to the evasion cell #1 (on the left). When the Evader reveals its true intention of evading at evasion cell #1, it is too late for the Pursuer to capture the Evader. It is also noteworthy that the stochastic wind field does disrupt the trajectories of the agents. For example, at point A, the Evader intends to go left directly to the evasion state but is instead pushed upward due to the stochastic wind field.

The second trajectory depicts a capture. Different from the previous scenario, the level-3 Pursuer has perfect knowledge on the level-2 policies of the Evader, and it predicts well regarding the next action the Evader will take.

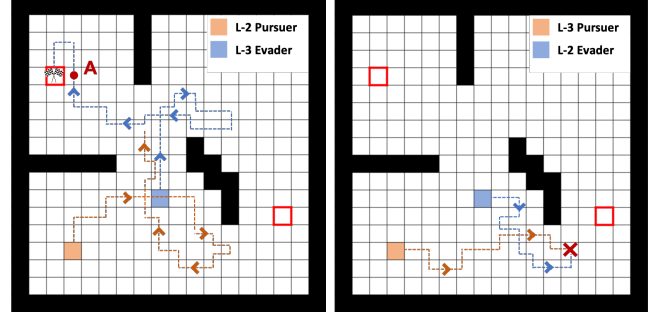


Fig. 3: Sample trajectories of agents with fixed levels. (a) Level-2 Pursuer vs. level-3 Evader; (b) Level-3 Pursuer vs. level-2 Evader.

The following color maps present an example of the outcomes from the fixed level model inferring algorithm introduced in Section V. Both agents have a maximum level of 5. The vertical axis represents the rationality levels of the opponents, and the x -axis represents the time instances. The color depicts the (normalized) conditional probability of the corresponding level.

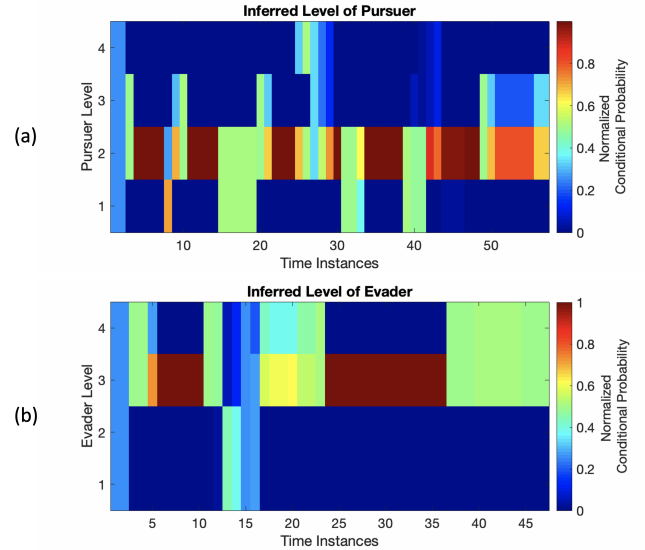


Fig. 4: Examples of inference results. (a) A level-3 Evader’s inference of a level-2 Pursuer; (b) A level-2 Pursuer’s inference of a level-3 Evader. In both cases, agents have maximum level of 5 and use a fixed level.

In Fig. 4(a), the level-3 Evader can infer the rationality level of the level-2 Pursuer with high confidence for most of the time steps. However, in Fig. 4(b), the level-2 Pursuer has some trouble inferring the level-3 Evader accurately after $t = 37$. As discussed earlier, at high levels the policies of the agents are the same at most of the states. For example, a level-4 Pursuer and a level-3 Pursuer may take the same action in certain regions of the state space. In (b), from time step 37 to 49, the two agents have entered such a region, where the policies of the level-3 Pursuer and the

level-4 Pursuer are the same. This phenomenon makes the inferring process challenging at high rationality levels, in general. However, since the policies of both the Pursuer and the Evader become similar at high levels, even picking a wrong level does not harm the performance significantly.

Finally, we present in Fig. 5 the outcome of the dynamic level model in Section V, where each agent always plays one level higher than the inferred level of its opponent without exceeding its maximum rationality level. One may notice some oscillations at the beginning of the game, but eventually, the Pursuer starts to play at its highest level, and the Evader plays at one level higher accordingly till the game terminates.

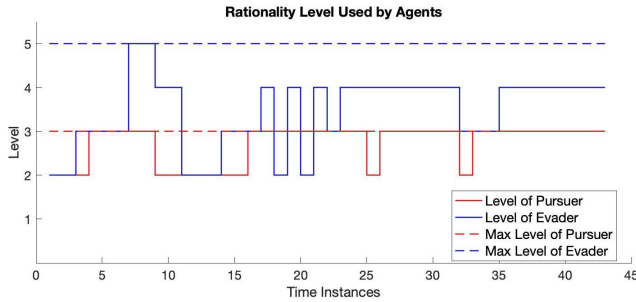


Fig. 5: An example of the dynamic level model. The Evader has a maximum rationality level of 3 and the Pursuer has 5. Both agents start at level 2. The game terminates at $n = 43$.

VII. CONCLUSIONS

In this work, we provide a framework to incorporate the notion of bounded rationality in decision-making for stochastic pursuit-evasion problems. Under this framework, we first discretize the game via the Markov chain approximation method and then solve the discretized game using the k -level thinking formulation. The resulted bounded-rational agents do not compute the Nash equilibrium, but they rather compute their level- k policy which is constructed from their opponent's level- $(k-1)$ policy. A bounded-rational agent also has a maximum rationality level restriction, such that it can only employ a policy of level up to $k_{\max}$. To enable agents select their levels adaptively during a game, we propose two inferring algorithms (fixed and dynamic) that assist the agent with estimating its opponent's level based on the observed trajectory. Through simulation results we have demonstrated the behavioral and statistical outcomes of the game with bounded-rational agents. In particular, we have demonstrated how an agent playing at a higher rationality level may be able to deceive the other lower-level agent. As part of future work, it is of interest to investigate this framework in a team game setup and study the implications of different rationality levels on the overall performance of the team.

ACKNOWLEDGEMENT

This work has been supported by ARL under DCIST CRA W911NF-17-2-0181.

REFERENCES

- [1] R. Isaacs, *Differential Games*. New York: John Wiley and Sons, 1965.
- [2] W. Willman, "Formal solutions for a class of stochastic pursuit-evasion games," *IEEE Transactions on Automatic Control*, pp. 504–509, 1969.
- [3] Y. Ho, A. Bryson, and S. Baron, "Differential games and optimal pursuit-evasion strategies," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 385–389, 1965.
- [4] R. Behn and Y. Ho, "On a class of linear stochastic differential games," *IEEE Transactions on Automatic Control*, vol. 13, pp. 227–240, 1968.
- [5] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory*. Society for Industrial and Applied Mathematics, 1999.
- [6] A. V. Balakrishnan, *Stochastic Differential Systems I*. Springer, 1973.
- [7] J. P. Hespanha, H. J. Kim, and S. Sastry, "Multiple-agent probabilistic pursuit-evasion games," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 3, pp. 2432–2437, 1999.
- [8] S. D. Bopardikar, F. Bullo, and J. P. Hespanha, "On discrete-time pursuit-evasion games with sensing limitations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1429–1439, 2008.
- [9] J. Shinar and S. Gutman, "Three-dimensional optimal pursuit and evasion with bounded controls," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 492–496, 1980.
- [10] W. Lin, Z. Qu, and M. A. Simaan, "Nash strategies for pursuit-evasion differential games involving limited observations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, pp. 1347–1356, 2015.
- [11] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.
- [12] J. P. Hespanha, M. Prandini, and S. Sastry, "Probabilistic pursuit-evasion games: a one-step nash approach," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 3, pp. 2272–2277 vol.3, 2000.
- [13] W. Willman, "Formal solutions for a class of stochastic pursuit-evasion games," *IEEE Transactions on Automatic Control*, vol. 14, no. 5, pp. 504–509, 1969.
- [14] R. Myerson, *Game Theory - Analysis of Conflict*. Harvard Press, 1997.
- [15] H. A. Simon, "Models of bounded rationality," *Organization Studies*, vol. 6, no. 3, pp. 308–308, 1985.
- [16] T. Gilovich, D. Griffin, and D. Kahneman, *Heuristics and Biases: The Psychology of Intuitive Judgment*. Cambridge University Press, 2002.
- [17] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," in *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing, STOC '06*, (New York, NY, USA), pp. 71–78, ACM, 2006.
- [18] A. W. Starr and Y. C. Ho, "Nonzero-sum differential games," *Journal of Optimization Theory and Applications*, vol. 3, pp. 184–206, 1969.
- [19] D. Kahneman, "Maps of bounded rationality: Psychology for behavioral economics," *American Economic Review*, vol. 93, no. 5, pp. 1449–1475, 2003.
- [20] C. F. Camerer, T.-H. Ho, and J.-K. Chong, "A cognitive hierarchy model of games," *The Quarterly Journal of Economics*, vol. 119, no. 3, pp. 861–898, 2004.
- [21] T.-H. Ho and X. Su, "A dynamic level-k model in sequential games," *Management Science*, vol. 59, no. 2, pp. 452–469, 2013.
- [22] J. Li, G. Kendall, and R. John, "Computing Nash equilibria and evolutionarily stable states of evolutionary games," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 460–469, 2016.
- [23] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Berlin, Heidelberg: Springer-Verlag, 1996.
- [24] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, 1992.
- [25] E. Isaacs and R. Srivastava, *Applied Geostatistics*. Oxford University Press, 1989.
- [26] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*. Hochschultext / Universitext, Springer, 2003.
- [27] I. Karatzas and S. Shreve, *Brownian Motion and Stochastic Calculus*. Graduate Texts in Mathematics, Springer New York, 2014.
- [28] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA: Athena Scientific, 1995.
- [29] A. J. Kleinman and H. J. Kushner, "Linear programming in a markov chain," *Oper. Res.*, vol. 10, no. 5, pp. 702–710, 1962.
- [30] W. Yoshida, R. J. Dolan, and K. J. Friston, "Game theory of mind," *PLOS Computational Biology*, vol. 4, no. 12, pp. 1–14, 2008.