Bounded-Rational Pursuit-Evasion Games

Yue Guan[1] Dipankar Maity[2] Christopher M. Kroninger[3] Panagiotis Tsiotras[4]

***Abstract*— We present a framework that incorporates the principle of bounded rationality into a pursuit-evasion game between two aerial vehicles in a stochastic wind field. We initially formulate the problem as a continuous zero-sum stochastic game under perfect rationality. We then discretize the game via the Markov Chain Approximation Method. Leveraging the cognitive hierarchy theory ("level-k thinking") we relax the perfect rationality assumption and compute the solution of the ensuing discrete game, while taking into consideration the rationality level of each agent. We also present an online algorithm to infer the rationality of the opponent, which enables the agents to deploy appropriate countermeasures. Finally, we verify the efficacy of this framework through simulations.**

I. Introduction

Pursuit-evasion games (PEGs) [1] are a special class of dynamic games introduced in the 60s. An extensive amount of literature exists on the topic; some notable examples include [2]–[4]. Most of these prior works are concerned with finding the equilibrium policy(s) of the game, which requires the assumption that all agents are rational [5]. Perfect rationality is, however, considered to be too strong and perhaps unrealistic for many applications, especially the ones involving humans [6]. Furthermore, finding the equilibrium of a game is, in general, computationally expensive [7]. To address these issues, we study pursuit-evasion games with agents that are not perfectly rational, but rather are of *bounded rationality*. Several manifestations of bounded rationality exist in the literature [8]–[10]. In this paper we will adopt the *level-k thinking* [10] to model agents under cognitive capability constraint.

Level-k thinking has demonstrated promising results modeling agents of limited cognitive capability (e.g., humans) in many scenarios [10]–[12]. Under this notion, a level-k agent no longer seeks the (Nash) equilibrium, but instead seeks a best response to its level-$(k-1)$ opponent. The notion of rationality is captured through the parameter k, and each agent has a maximum level up to which it can compute, hence the term *bounded rationality*.

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

[2]Dipankar 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

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

[4]Panagiotis 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

This work was supported by ARL under DCIST CRA W911NF-17-2-0181

Under the level-k framework, a pursuit-evasion game can be cast as a sequence of single-sided Markov-Decision-Processes (MDPs) [10], each of which can be solved efficiently. The computational complexity of finding the optimal policy for a participating agent is thus significantly reduced [7].

We are especially interested in the scenarios where an autonomous aerial vehicle competes with a human pilot in an aerial engagement. The level-k framework would enable the autonomous agent to actively exploit any weakness exhibited by the human pilot and deploy a level-$(k+1)$ optimal response, should it know that the human pilot is of level k. The important question of how well the level-k framework captures human behavior in stochastic PEG is to be answered in future work.

Contributions: The main contribution of this work is a detailed construction of a comprehensive and implementable method to find solutions to continuous stochastic PEG involving bounded rational agents. The proposed method first discretizes the continuous stochastic PEG in a way that ensures the convergence of the optimal solution of the discretized problem to that of the original continuous game. In addition, we show that if level-k thinking is applied to encode bounded rational decision-making, the discrete PEG can be solved efficiently using value iteration. We also present an inferring algorithm that updates the agent's belief regarding the rationality level of its 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.

II. Problem Formulation

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 the sequel, we use $-i$ 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 enter certain regions 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 agent i at time t is denoted by $p^i(t) = [p^i_x(t), p^i_y(t)]^\mathsf{T} \in \mathbb{R}^2$. 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 given by $s(t) = \left[p^1(t)^\mathsf{T}, p^2(t)^\mathsf{T}\right]^\mathsf{T} \in S = C \times C \subset \mathbb{R}^4$.

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 $\Theta = \Theta^1 \times \Theta^2$.

A. 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 (SDEs)

$$\mathrm{d}p^i = \begin{bmatrix} v^i \cos\theta^i + w_x(p^i) \\ v^i \sin\theta^i + w_y(p^i) \end{bmatrix} \mathrm{d}t + \mathrm{d}W_t^i(p^i), \quad i = 1, 2, \quad (1)$$

where $w_x(p)$ and $w_y(p)$ denote the mean wind velocity at p, and the two-dimensional Wiener process $\mathrm{d}W_t^i(p^i)$ satisfies $\mathbb{E}[\mathrm{d}W_t^i(p)\mathrm{d}W_t^i(p)^\mathsf{T}] = \sigma_w^2 \mathrm{I}_2$. For brevity, we denote $\theta = (\theta^1, \theta^2)$ and $\mathrm{d}W_t^i = \mathrm{d}W_t^i(p^i)$. We also denote $b^i(p^i, \theta^i) = \left[v^i \cos\theta^i + w_x(p^i),\ v^i \sin\theta^i + w_y(p^i)\right]^\mathsf{T}$.

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

$$\mathrm{d}s = \begin{bmatrix} b^1(p^1, \theta^1) \\ b^2(p^2, \theta^2) \end{bmatrix} \mathrm{d}t + \begin{bmatrix} \mathrm{d}W_t^1 \\ \mathrm{d}W_t^2 \end{bmatrix} = b(s, \theta)\,\mathrm{d}t + \mathrm{d}W_t. \quad (2)$$

Under some mild assumptions, the solution of the SDE in (2) is a stochastic process $\{s(t); t \geq 0\}$ such that

$$\begin{aligned} s(t) =& s(0) + \int_0^t b(s(\tau), \theta(\tau))\mathrm{d}\tau + \int_0^t \mathrm{d}W_\tau \\ \triangleq\ & \Phi(s(0), t, \theta^{[0,t]}, \mathrm{d}W^{[0,t]}), \end{aligned} \quad (3)$$

where the last term in equation (3) is treated as the usual Itô integral [13], $\theta^{[0,t]} = \{\theta(\tau); \tau \in [0, t]\}$ is the joint action history, and $\mathrm{d}W^{[0,t]}$ is the realization of the wind field.

B. Terminal Conditions of the Game

We define three terminal conditions for the PEG: *Crash*, *Capture* and *Evasion*.

(1) *Crash* corresponds to the scenario when an agent runs into obstacles or reaches the boundary of the domain 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 $t > 0$,

$$\mathrm{dist}\big(p^i(t), O \cup \partial C\big) = 0,$$

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

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

(2) *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 agents crashes. The capture condition defines a boundary of S via

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

(3) *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

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

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 $S^o = S \setminus \partial S$. Without loss of generality, we assume that the game starts at some initial state $s(0) \in S^o$.

C. 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 [7]. A Markov policy of agent i depends only on the current state $s(t)$ and is represented by the mapping $\mu^i(\cdot, \cdot) : S^o \times \Theta^i \to [0, 1]$. The set of all such policies for agent i is denoted by Π^i. We denote the joint policy as $\mu = (\mu^1, \mu^2)$ and the set of joint policies as Π. Define the *first exit time* T_μ under joint policy μ as

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

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 assume a zero-sum formulation and let the Pursuer be the *maximizer* and the Evader be the *minimizer*. Then, the terminal reward at terminal state $s \in \partial S$ is given by

$$g(s) = \begin{cases} 1 & \text{if } s \in \partial S_{\text{cap}} \bigcup \left(\partial S_{\text{crsh}}^2 \setminus \partial S_{\text{crsh}}^1\right), \\ -1 & \text{if } s \in \partial S_{\text{evs}} \bigcup \left(\partial S_{\text{crsh}}^1 \setminus \partial S_{\text{crsh}}^2\right), \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

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 time or fuel cost) into the formulation requires a minor modification and will be addressed elsewhere.

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

$$J_\mu(s_0) = J_{\mu^1, \mu^2}(s_0) = \mathbb{E}_{s_0, \mu}\Big[g\big(s(T_\mu)\big) \Big], \quad (5)$$

where the conditional expectation is given by

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

In the PEG setting, J_μ incorporates the *expected win rate* of the Pursuer. Each agent tries to maximize its own expected win rate by choosing policy μ^i, and the resulting optimization problem is given by

$$\sup_{\mu^1 \in \Pi^1} \inf_{\mu^2 \in \Pi^2} J_{\mu^1, \mu^2}(s_0) = \inf_{\mu^2 \in \Pi^2} \sup_{\mu^1 \in \Pi^1} J_{\mu^1, \mu^2}(s_0). \quad (6)$$

III. Markov Chain Approximation

In this section, we introduce the *Markov Chain Approximation Method* (MCAM) [14] to approximate the stochastic process in (2) via a sequence of discrete-state, discrete-time competitive Markov Decision Processes (cMDPs) [7].

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. The transition function is $P_h(\cdot|\cdot,\cdot,\cdot) : S_h \times S_h \times \Theta^1 \times \Theta^2 \to [0,1]$, and $G_h : S_h \to \mathbb{R}$ is the terminal reward.

A. Discrete State Space S_h and Terminal Reward G_h

We start by discretizing the compact region C with a grid-size $h > 0$ (see Fig. 1). The discretization size h is a user-defined parameter that determines the resolution of the discretization. Let us define the set $C_h = \{c_{h,1}, \ldots, c_{h,N}\}$ 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 represents a unique square cell of size h in $\mathbb{R}^2$ (including boundary). Let c_h^o be the interior of the square cell c_h. The cell c_h is labeled as an *obstacle cell* if and only if $c_h^o \cap O \neq \emptyset$. The *boundary cells* and the *evasion cells* are defined similarly. We say that agent i is in cell c_h, if $p^i(t) \in c_h^o$[1]. The discretization of S is then $S_h = C_h \times C_h$. 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 and define

$$\begin{aligned}
\partial S^i_{h,\text{crsh}} &= \{s_h \in S_h : s_h^o \cap \partial S^i_{\text{crsh}} \neq \emptyset\},\\
\partial S_{h,\text{crsh}} &= \partial S^1_{h,\text{crsh}} \cup \partial S^2_{h,\text{crsh}} \ ,\\
\partial S_{h,\text{cap}} &= \{s_h \in S_h \setminus \partial S_{h,\text{crsh}} : s_h^o \cap \partial S_{\text{cap}} \neq \emptyset\},\\
\partial S_{h,\text{evs}} &= \{s_h \in S_h \setminus (\partial S_{h,\text{crsh}} \cup \partial S_{h,\text{cap}}) : s_h^o \cap \partial S_{\text{evs}} \neq \emptyset\},
\end{aligned} \tag{7}$$

and the whole boundary $\partial S_h = \partial S_{h,\text{crsh}} \cup \partial S_{h,\text{cap}} \cup \partial S_{h,\text{evs}}$.

The discretized terminal cost G_h is defined similar to (4):

$$G_h(s_h) = \begin{cases} 1 & \text{if } s_h \in \partial S_{h,\text{cap}} \bigcup \left(\partial S^2_{h,\text{crsh}} \setminus \partial S^1_{h,\text{crsh}}\right),\\ -1 & \text{if } s_h \in \partial S_{h,\text{evs}} \bigcup \left(\partial S^1_{h,\text{crsh}} \setminus \partial S^2_{h,\text{crsh}}\right),\\ 0 & \text{otherwise.}\end{cases}$$

B. Discrete Transition P_h

For each discretization h, we have $s_h^{[0,N]} = \{s_h^n; n \leq N\}$ as a *controlled Markov Chain* [14] under some policy, which terminates when it hits ∂S_h. The superscript n in s_h^n denotes the time instance in the discrete cMDP $\mathcal{M}_h$.

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* [14]. 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

[1]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.

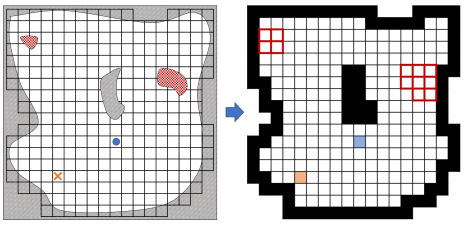


Fig. 1: An example of a discretization of the two-dimensional domain. Left is the continuous domain C, where the shaded red and gray areas are the evasion regions and the obstacles respectively. The orange and blue markers represent the positions of the Pursuer and the Evader. Right is the discretized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

denote Δt_h^n to be the holding time at state s_h^n, i.e., $\Delta t_h(s_h^n)$, and define $t_h^n = \sum_0^{n-1} \Delta t_h^n$ for $n \geq 1$ and $t_h^0 = 0$.

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* [14], with respect to (2), which are given by the following conditions.

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

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

As the chain $\{s_h^n; n \in \mathbb{N}_0\}$ is a discrete-time process, we use an *approximate continuous-time interpolation* [14] 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 [14] 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

$$b(s,\theta) = [b_1(s,\theta),\ b_2(s,\theta),\ b_3(s,\theta),\ b_4(s,\theta)]^\mathsf{T}.$$

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

$$\Delta t_h(s_h) = \frac{h^2}{Q_h(s_h)}. \tag{8}$$

Notice that $\Delta t_h(s_h) \to 0$ as $h \to 0$, for all $s_h \in S_h$. The transition probabilities of the cMDP that approximates the

original PEG can be calculated via

$$\begin{aligned} P_h(s_h \pm he_j|s_h,\theta) &= \Big(\frac{\sigma_w^2}{2} + hb_j^{\pm}(s_h)\Big)/Q_h(s), \\ P_h(s_h|s_h,\theta) &= 1 - \sum_{j=1}^{4} P_h(s_h \pm he_j|s_h,\theta), \\ P_h(s_h'|s_h,\theta) &= 0, \text{ if } s_h' \neq s_h \pm he_j \text{ and } s_h' \neq s_h, \end{aligned} \tag{9}$$

where $b^+ = \max\{b, 0\}$ and $b^- = \max\{-b, 0\}$, and e_i are the unit vectors in $\mathbb{R}^4$ (e.g., $e_3 = [0\ 0\ 1\ 0]^\mathsf{T}$). The next states $s_h \pm he_j$ are the states s_h' such that $\alpha(s_h') = \alpha(s_h) \pm he_j$. One can verify that the transition probabilities defined in (9) are locally consistent with (2).

Under this discretization scheme, any advantage an agent has in its speed is translated into a probabilistic advantage in (9). Specifically, P_h depends on v^i through $b_i(s_h, \theta)$ and one can observe that within the same wind field, the agent with a higher speed is more likely to end up in the cell that it intends to visit.

As stated in [14], 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 $\mu_h^i : S_h \times \Theta^i \to [0, 1]$. The set of all admissible policies is denoted by Π_h^i. To this end, given a joint policy μ_h, the expected reward-to-go from s_h due to μ_h is

$$J_{h,\mu_h}(s_h) = J_{h,\mu_h^1,\mu_h^2}(s_h) = \mathbb{E}_{P_h,\mu_h}\Big[G_h\big(s_h^{I_h}\big)\Big], \tag{10}$$

where the expectation is taken under P_h and joint policy μ_h, and $s_h^{I_h}$ is the terminal state reached at the terminal time I_h[2].

IV. 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 a cMDP $\mathcal{M}_h$ in Section II, suppose that agent i is given the policy of its opponent μ_h^{-i}, it can marginalize the transition in (9) using μ_h^{-i}. The cMDP $\mathcal{M}_h$ is then reduced to a standard MDP optimizing with respect to only μ_h^i. In what follows, we use the superscript $k \in \{0, 1, 2, \ldots\}$ 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 [7]:

$$V^{i,(k)}(s) = \operatorname*{ext}_{\mu^{i,(k)}} \Big\{ \sum_{s'\in S} P^{i,(k)}\big(s'|s,\mu^{i,(k)}\big)V^{i,(k)}(s')\Big\}, \tag{11}$$

with the boundary condition $V^{i,(k)}(s) = G_h(s)$ for $s \in \partial S_h$. The ext operator in (11) corresponds to a sup when the Pursuer optimizes and to an inf for the Evader. It has been shown that there exists at least one pure Markov policy [7] that solves (11).

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

$$\mu^{i,(0)}(s,\theta^i) = \frac{1}{|\Theta^i|} \quad \text{for all } \theta^i \in \Theta^i, s \in S, \tag{12}$$

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 optimization[3].

The level-1 agents then calculate their best response to their opponent's level-0 policy via (11). Similarly, the level-2 agents compute their best response to the given level-1 policies via (11). 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 $k_{\max}^i$ [10].

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. We want to emphasize the fact that the Nash policy [7], even though robust, is not always optimal given the opponent's policy. For example, if an autonomous agent is certain that its opponent (say, a human, bounded rational pilot) is unable to deploy a Nash policy but instead uses a policy at level-k, then there is an incentive for the autonomous agent to respond using a level-$(k+1)$ policy rather than a Nash policy to exploit the weakness of its opponent and thus maximize its reward.

As discussed above, the level-wise best response structure in (11) can be equivalently represented by

$$\mu^{i,(k+1)} \in \text{BestResponse}(\mu^{-i,(k)}). \tag{13}$$

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

Lemma 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 joint policy $\big(\mu^{i,(K)}, \mu^{-i,(K+1)}\big)$.*

Proof. By the construction of level-k policies, we have

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

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

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

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

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

[3]One may choose different level-0 policies. For example, a simple Pursuer policy is one that keeps the heading towards the Evader. Different level-0 policies, however, may result in different level-k policies.

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

Remark 2. *Lemma 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.*

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 of the MDPs, the optimization is only over a single agent's policy and we utilize value iteration [15] to find the optimal policy. We use the subscript m to denote the iteration index. Following this notation, the value iterations are given by

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

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

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.

V. Inferring the Opponent's Level

In Section IV, we argued that agent i can deploy its "countermeasure" policy $\mu^{i,(k+1)}$, if agent i knows its opponent is of level k and $k+1$ does not exceed its maximum level $k_{\max}^i$. In real world situations, it is not always possible to know the opponent's exact level, hence, the agent must infer the level of its opponent based on the trajectory of the game. 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 similar to the one in [16].

To make the interactions between the two agents more realistic, we allow the two agents to adapt their levels based on their observations. We assume that agent i always plays one level higher than the estimated level of its opponent, without exceeding its maximum level $k_{\max}^i$. With an observation window of length $w+1$, denote the observed trajectory at t_N as $s^{[N-w,N]} = \{s_{N-w}, \ldots, s_N\}$ and denote the levels played by agent i over this period as $k^{i,[N-w,N]} = \{k_{N-w}^i, \ldots, k_N^i\}$. With a maximum level $k_{\max}^i$, agent i can infer the opponent's level in the range of $\mathcal{K}^i = \{0, 1, \ldots, k_{\max}^i\text{-}1\}$[4]. The probability of observing this specific trajectory $s^{[N-w,N]}$ with a fixed $k^{-i} \in \mathcal{K}^i$ is

$$\mathbb{P}(s^{[N-w,N]} | k^{i,[N-w,N]}, k^{-i}) = \prod_{n=N-w}^{N-1} P_h\Big(s_{n+1}|s_n,\ \mu^{i,(k_n^i)}, \mu^{-i,(k^{-i})}\Big). \tag{14}$$

[4]To construct its policy up to level $k_{\max}^i$, agent i is given or has computed its opponent's policies up to level $k_{\max}^i - 1$, which decides the set $\mathcal{K}^i$.

We use maximum likelihood estimator to infer the opponent's level at t_N via

$$\hat{k}_N^{-i} \in \underset{k^{-i} \in \mathcal{K}^i}{\text{argmax}}\ \mathbb{P}\Big(s^{[N-w,N]} | k^{-i}, k^{i,[N-w,N]}\Big). \tag{15}$$

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

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 original PEG to obtain the grid to the right. The reward G_h and the transition P_h are assigned according to (4) and (9). Both agents have the same speed $v^1 = v^2 = 1$ and the same action set $\Theta^1 = \Theta^2 = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.

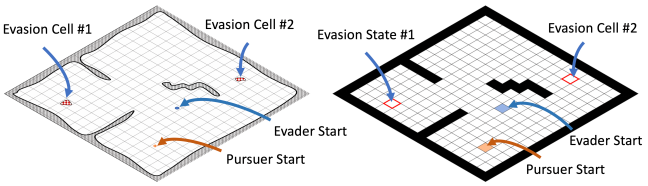


Fig. 2: The continuous space (left) and the discretized grid (right).

We construct the level-k thinking hierarchy as presented in Section IV. We then simulate the computed policies at some selected level pairs over 1500 games. We present two tables of win rates to illustrate the results.

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.

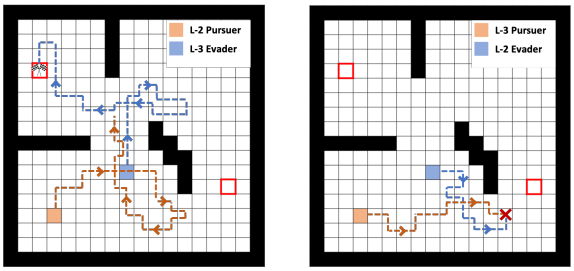
Pursuer Level	1	2	3	4	5	6
Pursuer Wins	48.5	47.6	51.7	50.9	51.2	51.1
Capture	36.4	38.2	45.3	43.1	42.7	42.9

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. Fig. 3(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.

Fig. 3(b) depicts a successful 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.



(a) (b)

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.

Fig. 4 presents an example of the outcomes from the inferring algorithm introduced in Section V. The color depicts the (normalized) conditional probability in (14).

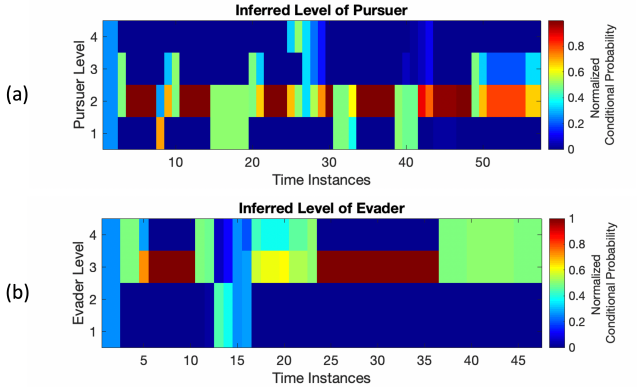


Fig. 4: Examples of inference results. (a) A level-3 Evader's inference of a level-2 Pursuer; (b) A level-2 Pursuer's inference of a level-3 Evader. In both cases, agents have maximum level of 5 and use a fixed level.

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 Fig. 4(b), from time step 37 to 49, the two agents have entered such a region. 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 adaptive level selection 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, but eventually, the Pursuer starts to play at its highest level, and the Evader plays at one level higher accordingly till the game terminates.

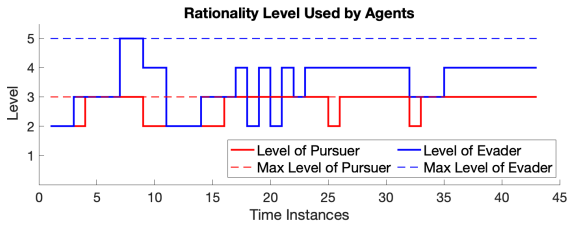


Fig. 5: An example of the dynamic level model. The Evader has a maximum rationality level of 3 and the Pursuer has 5. Both agents start at level 2. The game terminates at $n = 43$.

VII. Conclusions

In this work, we provide a framework to incorporate the notion of bounded rationality for continuous stochastic pursuit-evasion problems. We first presented a discretization scheme to reduce the original game to a discrete competitive MDP. We then utilized the level-k framework to model agents of bounded rationality and demonstrated how the game can then be solved efficiently using value iteration. Moreover, we proposed an inferring algorithm that estimates the opponent's rationality level and enables the agents to adapt their levels accordingly. Finally, we demonstrated the behavioral and statistical outcomes of a game with bounded-rational agents with a simulation example.

Future work will examine how well the level-k framework models human behaviors in PEGs. It is also of interest to investigate this framework in a team game setup.

References

[1] R. Isaacs, *Differential Games*. New York: John Wiley and Sons, 1965.

[2] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory*. Society for Industrial and Applied Mathematics, 1999.

[3] 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.

[4] 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.

[5] R. Myerson, *Game Theory - Analysis of Conflict*. Harvard Press, 1997.

[6] H. A. Simon, "Models of bounded rationality," *Organization Studies*, vol. 6, no. 3, pp. 308–308, 1985.

[7] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Berlin, Heidelberg: Springer-Verlag, 1996.

[8] D. Kahneman, "Maps of bounded rationality: Psychology for behavioral economics," *American Economic Review*, vol. 93, no. 5, pp. 1449–1475, 2003.

[9] S. J. Gershman, E. J. Horvitz, and J. B. Tenenbaum, "Computational rationality: A converging paradigm for intelligence in brains, minds, and machines," *Science*, vol. 349, no. 6245, pp. 273–278, 2015.

[10] T.-H. Ho and X. Su, "A dynamic level-k model in sequential games," *Management Science*, vol. 59, no. 2, pp. 452–469, 2013.

[11] V. P. Crawford and N. Iriberri, "Level-k auctions: Can a nonequilibrium model of strategic thinking explain the winner's curse and overbidding in private-value auctions?," *Econometrica*, vol. 75, no. 6, pp. 1721–1770, 2007.

[12] A. Arad and A. Rubinstein, "The 11-20 money request game: A level-k reasoning study," *American Economic Review*, vol. 102, no. 7, pp. 3561–73, 2012.

[13] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*. Hochschultext / Universitext, Springer, 2003.

[14] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, 1992.

[15] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA: Athena Scientific, 1995.

[16] W. Yoshida, R. J. Dolan, and K. J. Friston, "Game theory of mind," *PLOS Computational Biology*, vol. 4, no. 12, pp. 1–14, 2008.