PACE: A Framework for Learning and Control in Linear Incomplete-Information Differential Games

Seyed Yousef Soltanian SSOLTAN2@ASU.EDU
School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85287, USA

Wenlong Zhang WENLONG.ZHANG@ASU.EDU
School of Manufacturing Systems and Networks, Arizona State University, Mesa, AZ, 85212, USA.

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

In this paper, we address the problem of a two-player linear quadratic differential game with incomplete information, a scenario commonly encountered in multi-agent control, human-robot interaction (HRI), and approximation methods to solve general-sum differential games. While solutions to such linear differential games are typically obtained through coupled Riccati equations, the complexity increases when agents have incomplete information, particularly when neither is aware of the other's cost function. To tackle this challenge, we propose a model-based Peer-Aware Cost Estimation (PACE) framework for learning the cost parameters of the other agent. In PACE, each agent treats its peer as a learning agent rather than a stationary optimal agent, models their learning dynamics, and leverages this dynamic to infer the cost function parameters of the other agent. This approach enables agents to infer each other's objective function in real time based solely on their previous state observations and dynamically adapt their control policies. Furthermore, we provide a theoretical guarantee for the convergence of parameter estimation and the stability of system states in PACE. Additionally, using numerical studies, we demonstrate how modeling the learning dynamics of the other agent benefits PACE, compared to approaches that approximate the other agent as having complete information, particularly in terms of stability and convergence speed.

Keywords: Multi-Agent Interactions, Differential Games, Learning from a Learner

1. Introduction

General-sum differential games, referring to a type of game where agents do not necessarily cooperate or compete, is a powerful tool for modeling multi-agent interactions (Schwarting et al., 2019) and human-robot interactions (HRI) (Li et al., 2019; Losey et al., 2018). The feedback Nash equilibrium (FBNE) solution to general-sum non-cooperative differential games can be obtained by solving a set of coupled Hamilton-Jacobi-Isaacs (HJI) equations (Bressan, 2010; Crandall and Lions, 1983). However, it is well known that traditional dynamic programming methods for solving these HJI equations suffer from the curse of dimensionality (Powell, 2007). To address this challenge, recent research has proposed iterative linear-quadratic approximations of these games (Fridovich-Keil et al., 2020), showing the importance of studying linear-quadratic differential games. When dealing with infinite-horizon linear quadratic differential games, it has been shown that the problem of finding the feedback Nash equilibrium can be reduced to the problem of solving a set of algebraic Riccati equations (ARE) (Başar and Olsder, 1998). Although there have been methods for solving coupled Riccati equations (Başar and Olsder, 1998; Cherfi et al., 2005; Li and Gajic, 1995), it requires all the players to have complete information about each other's objective function. The problem of finding the feedback Nash equilibrium will become challenging when dealing with

agents with incomplete information. This scenario commonly arises in linear game modeling of HRI tasks when the human and the robot are not aware of each other's objective (Li et al., 2019; Wang et al., 2022; Ji et al., 2018; Franceschi et al., 2023; Wu et al., 2024), or in multi-agent interactions where agents are not aware of each other's intent (Peters et al., 2021; Wang et al., 2019; Liu et al., 2016; Li et al., 2024), or in games with control imperfection (Rabbani et al., 2025).

In this paper, we address the problem of a two-player, general-sum linear game with a quadratic infinite horizon cost under incomplete information. Our proposed Peer Aware Cost Estimation (PACE) algorithm leverages prior knowledge of the other agent's learning dynamics, enabling both agents to learn each other's objective function in real-time while simultaneously updating their control policies based on these updated estimates. Each agent uses a history of past state observations (without any requirement to observe the other agent's actions) to minimize the error between the expected trajectory and the actual observed trajectory by updating its belief over the other agent's cost parameters using gradient descent. Our method presents a major departure from common approximation methods where an agent models the other agents as an agent with complete information or experts (Le Cleac'h et al., 2021; Schwarting et al., 2019), during the rest of this paper, we refer to these methods as "complete information peer approximation". Despite the empirical success of these approximations in many scenarios, these methods can fail when dealing with two learning agents (Liu et al., 2016), as shown in Section 5. On the contrary, in PACE, each agent simulates the learning process of the other agent at each belief update stage. As a result, avoiding biased estimates at each step rather than treating the other agent as a complete information agent. The structure of PACE allows us to theoretically guarantee the convergence of the online cost parameter inference of each agent to a bounded region around the true parameters while simultaneously ensuring the stability of the coupled system states when both agents are performing PACE. Although our proposed algorithm utilizes a linear time-invariant system model, its core idea can also be applied to time-varying LQ games using coupled differential Riccati equations and further extended to nonlinear games through iterative LQ approximations of general multi-agent systems (Fridovich-Keil et al., 2020). **Contributions. (1) Learning from an Incomplete-Information Agent.** Although differential games with multiple learners have been explored, such as in (Lian et al., 2022), existing work focuses on learners interacting with an expert. Our framework departs from expert-based approaches and addresses interactions between two incomplete information agents, each inferring the other's objectives. **(2) Learning from Shared State Observations.** In practical scenarios, observing another agent's actions may be impractical. Unlike works such as (Tian et al., 2023; Li et al., 2016), our algorithm relies solely on shared state observations to infer the other agent's objective. **(3) Theoretical Guarantee.** The PACE algorithm provides theoretical convergence and stability guarantees. Experimentally, we have shown its robustness and stability across diverse initial parameter estimation guesses, initial policies, and learning rates compared to the idea of treating the other agent as an expert, highlighting the importance of modeling the other agent's learning dynamics.

2. Related Works

Multi-agent Interactions. Multi-agent interactions have been extensively studied (Bloembergen et al., 2015) through multi-agent reinforcement learning (Canese et al., 2021), adaptive control (Chen et al., 2019), and game theory (Yang and Wang, 2020). Many of these interactions can be modeled as general-sum differential games (Zhang et al., 2023, 2024), which are challenging to solve over continuous state and action spaces. While adaptive dynamic programming methods exist

for finding feedback Nash equilibria (FBNE) (Vamvoudakis and Lewis, 2011), classical dynamic programming faces the "curse of dimensionality" (Powell, 2007). Recent work has explored iterative linear-quadratic approximations for multi-agent interactions (Fridovich-Keil et al., 2020). Under linear system and quadratic cost assumptions, Nash equilibria have been well studied (Possieri and Sassano, 2015; Engwerda, 1998) within the linear-quadratic non-cooperative differential games (LQ games) (Başar and Olsder, 1998; Lukes and Russell, 1971). Our model-based (LQ) game approach simplifies the challenge of finding value functions in general-sum differential games by focusing on inferring the peer agent's objective function. This enables us to study the benefits of modeling the learning dynamics of other agents in multi-agent interaction tasks.

Differential Games with Incomplete Information. The existence of value and player policies for general-sum differential games with incomplete information remains an open question, unlike their zero-sum or discrete action/state counterparts (Aumann et al., 1995; Cardaliaguet and Rainer, 2012). Solving these games often requires simplifications, such as treating one agent as an uncertain learner and the others as fully informed (Le Cleac'h et al., 2021; Laine et al., 2021), updating via belief space (Chen et al., 2021), learning from offline datasets (Mehr et al., 2023), one-step cost minimization (Liu et al., 2016), or using observer-based adaptive control methods to estimate other agents' parameters (Li et al., 2019; Lin et al., 2023; Franceschi et al., 2023), and still majority of works tackling incomplete information games focusing on one uncertain agent learning from the expert agent. Our work is motivated by the fact that incomplete information differential games, where all agents are treated as learners, have rarely been analyzed (Jacq et al., 2019), even for linear systems. One example of such a study can be found in (Liu et al., 2016), which compares the drawbacks of approximating the other agent as an expert versus treating them as a learner, but only for games focused on immediate cost minimization.

Inverse Optimal Control and Inverse Reinforcement Learning. Traditional IOC and IRL methods rely on observing optimal trajectories to retrieve cost or reward parameters but are unsuitable for online learning and control due to their need for full trajectory observations (Molloy et al., 2022; Ng et al., 2000). This has led to the development of online IOC and IRL (Molloy et al., 2020; Rhinehart and Kitani, 2017; Self et al., 2022) and specifically inverse LQR (Priess et al., 2014; Zhang and Ringh, 2024; Xue et al., 2021). Recent work on inverse non-cooperative dynamic games focuses on recovering cost functions by observing the Nash equilibria (Molloy et al., 2022), particularly for linear quadratic dynamic games (Inga et al., 2019; Yu et al., 2022; Li et al., 2023). Another research addresses the problem of controlling a learner system based on observing the Nash equilibrium of an expert system and recovering their cost parameters (Lian et al., 2022). There has been less focus on learning from another learner (Jacq et al., 2019; Foerster et al., 2017), and a limited number of studies, to our knowledge, explore two agents learning each other's objective function and interacting simultaneously (Liu et al., 2016; Amatya et al., 2022).

3. Preliminaries

Problem Statement. We consider a continuous-time differential game between two agents i and j interacting over an infinite horizon. The system dynamics are given by:

$$\dot{x}(t) = Ax(t) + B_i u_i(t) + B_j u_j(t), \quad x(0) = x_0, \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, and $u_i(t), u_j(t) \in \mathbb{R}^m$ are the control inputs of agents i and j, respectively. The system matrices $A \in \mathbb{R}^{n\times n}$, $B_i, B_j \in \mathbb{R}^{n\times m}$ are known to both agents. Each

agent aims to minimize its own infinite-horizon cost function. For agent $k \in \{i, j\}$, the cost function is defined as:

$$J_k(u_k, u_{-k}) = \int_0^\infty \left(x(t)^\top Q_k x(t) + u_k(t)^\top R_k u_k(t) \right) dt, \tag{2}$$

where $Q_k \in \mathbb{R}^{n\times n}$ is a positive semi-definite state weighting matrix, $R_k \in \mathbb{R}^{m\times m}$ is a positive definite control weighting matrix, and $u_{-k}(t)$ denotes the control input of the other agent.

We assume that all states $x(t)$ are observable by both agents, but the agents are not able to observe the actions of their peer $u_{-k}(t)$. For each agent $k \in \{i, j\}$, the pair (A, B_k) is controllable, and the pair $(A, \sqrt{Q_k})$ is detectable. Notation: $\hat{\cdot}$ denotes estimations (e.g., $\hat{Q}_{-k}$ is agent k's estimate of the other's cost and $\hat{Q}_k$ is agent k estimation of agent $-k$'s estimation of Q_k), while subscripts k and $-k$ refer to agent k and the opposing agent, respectively. For the sake of brevity, the operator $\mathcal{RIC}_k(P_k, P_{-k}, Q_k)$ represents an algebraic Riccati equation from agent k's perspective to find P_k with a coupling term containing P_{-k} and the state cost matrix of Q_k. In this work, for the sake of brevity, we assume $R_k = R_{-k} = I$ for the rest of this paper and focus on learning the Q_k matrices.

Nash Equilibrium. The objective of such games is to find the feedback Nash equilibrium policies $u_i^*(t), u_j^*(t)$ such that, for each agent k:

$$J_k(u_k^*, u_{-k}^*) \leq J_k(u_k, u_{-k}^*), \quad \forall u_k. \tag{3}$$

Under the assumption of linear dynamics and quadratic costs, the value function of each agent k takes the quadratic form $V_k^*(x) = x^\top P_k^* x$. where $P_k^* \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix to be determined.

Coupled HJB Equations. The Hamilton-Jacobi-Bellman (HJB) equation for agent k is:

$$0 = \min_{u_k} \left\{ x^\top Q_k x + u_k^\top R_k u_k + \left(\frac{\partial V_k^*}{\partial x} \right)^\top (Ax + B_i u_i + B_j u_j) \right\}. \tag{4}$$

Coupled Riccati Equations. Solving the above equation for each agent at the Nash equilibrium results in a linear control policy of $u_k^* = -K_k^* x = -R_k^{-1} B_k P_k^* x$. Substituting this into the HJB equation and noting that $\frac{\partial V_k^*}{\partial x} = 2P_k^* x$, we obtain the coupled Algebraic Riccati Equations (AREs) for agents i and j:

$$0 = \left(A - B_j R_j^{-1} B_j^\top P_j^* \right)^\top P_i^* + P_i^* \left(A - B_j R_j^{-1} B_j^\top P_j^* \right) - P_i^* B_i R_i^{-1} B_i^\top P_i^* + Q_i, \tag{5}$$

$$0 = \left(A - B_i R_i^{-1} B_i^\top P_i^* \right)^\top P_j^* + P_j^* \left(A - B_i R_i^{-1} B_i^\top P_i^* \right) - P_j^* B_j R_j^{-1} B_j^\top P_j^* + Q_j, \tag{6}$$

which can be written as $\mathcal{RIC}_k(P_j^*, P_i^*, Q_i) = 0$ and $\mathcal{RIC}_k(P_i^*, P_j^*, Q_j) = 0$. These equations are coupled due to the presence of P_{-k}^* in each agent's equation. Solving these AREs yields the Nash equilibrium solutions P_i^*, P_j^*, and the resultant optimal policies.

4. PACE for Solving Incomplete Information Linear Differential Games

In scenarios where agents are unaware of each other's cost matrices Q_{-k}, they need to estimate $\hat{Q}_{-k}$ and subsequently the other agent's Riccati solution P_{-k} in their own Riccati equation as $\hat{P}_{-k}$. We introduce PACE as a two-step algorithm: first, we explain how each agent in PACE utilizes $\hat{P}_{-k}$ to update their control policies dynamically; next, we detail how $\hat{P}_{-k}$ and $\hat{Q}_{-k}$ are updated and learned during the interaction based on modeling the learning dynamic of the other agent and previous state observations.

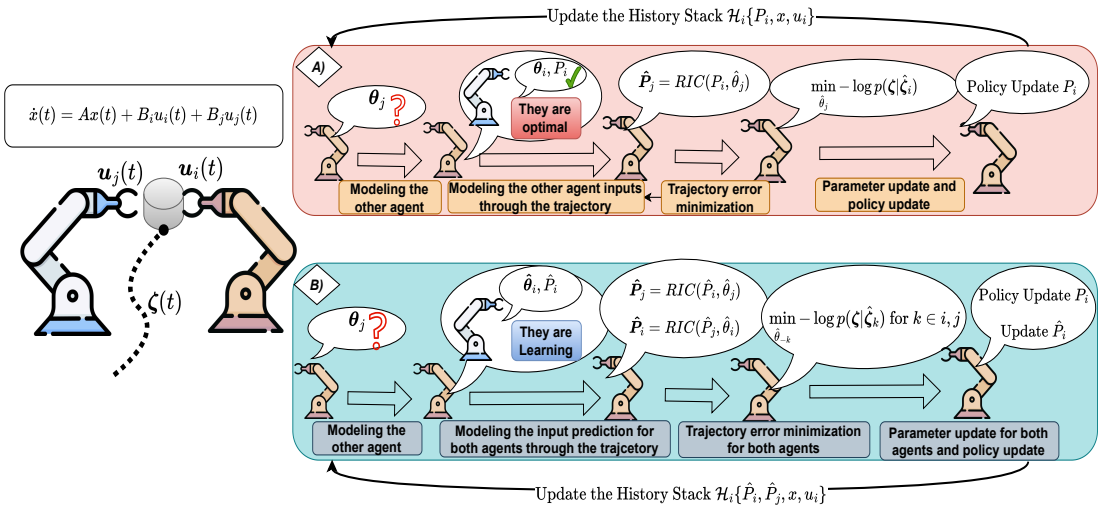

Figure 1: An illustrative example of two robotic agents moving an object with full state trajectory observation $x(t)$, although agents are not able to observe each other's interaction force $u(t)$. Assuming an accurate low-level control of the end effectors in the task space, the interaction dynamics is modeled as a linear system. The agents are unaware of each other's cost function parameter, denoted as θ. The agent i focuses on minimizing the observed trajectory error and updating the parameter estimates in real time. However, in (A), i assumes its partner has complete information (resulting in a biased estimation), whereas in (B), i not only performs its own parameter estimation but also accounts for its partner's learning process.

PACE: Policy Update

In PACE, at each decision-making time step t when agent k updates their estimation of the other agent cost parameters $\hat{Q}_{-k}^{(t)}$ and $\hat{P}_{-k}^{(t)}$, they use the following Riccati equation to update their $P_k^{(t)}$ and their corresponding policy $u_k(t) = -B_k^\top P_k^{(t)} x(t)$:

$$0 = \left(A - B_{-k} B_{-k}^\top \hat{P}_{-k}^{(t)}\right)^\top P_k^{(t)} + P_k^{(t)} \left(A - B_{-k} B_{-k}^\top \hat{P}_{-k}^{(t)}\right) - P_k^{(t)} B_k B_k^\top P_k^{(t)} + Q_k. \quad (7)$$

To address the incomplete information scenario, it is necessary to rely on observations of past interactions (states) to estimate Q_{-k} for both agents and recover the complete information game. A common approximation assumes that each agent treats their peer as a complete information agent, meaning their actions are considered optimal (Laine et al., 2021). Consequently, each agent attempts to infer the peer's objective, as illustrated in Fig. 1(A). However, when dealing with two learner agents, this approach, in general, is a biased estimation (Liu et al., 2016). We will experimentally demonstrate its potential failure in certain scenarios, as shown in Section 5. In PACE, we develop an online learning algorithm to estimate $\hat{Q}_{-k}$ and $\hat{P}_{-k}$ by accounting for the learning dynamics of the peer agent, as conceptually illustrated in Fig. 1(B).

PACE: Belief Update

Our multi-agent learning algorithm is inspired by cognitive theories of human learning, which suggest that humans may update their forward model using the model's prediction error as loss functions (Schaefer et al., 2012). Additionally, to develop our learning algorithms, we draw on the concept of history stacks from concurrent learning (CL) adaptive control (Kamalapurkar et al., 2017).

Definition 1. A **history stack** for agent k at time t, denoted by $\mathcal{H}_k^t\{x, u_k, \hat{P}_{-k}, \hat{P}_k\}$, is a collection of $x(\cdot)$, $u_k(\cdot)$, $\hat{P}_{-k}^{(\cdot)}$, and $\hat{P}_k^{(\cdot)}$ recorded at sample times $\tau_1 < \cdots < \tau_N \leq t$.

Definition 2. To define the **trajectory** in a continuous-time (CT) system, we represent it at time t as a sequence of sampled state values drawn from the history stack $\mathcal{H}_k^t$, which contains $x(\tau)$ at decision-making sample times up to t. Thus, the trajectory $\boldsymbol{\zeta}(t)$ is defined as $\boldsymbol{\zeta}(t) = \{x(\tau) \mid \tau \in \{\tau_1, \tau_2, \ldots, \tau_N\}\}$, where $\{\tau_1, \tau_2, \ldots, \tau_N\}$ are the same times recorded in $\mathcal{H}_k^t$.

In our framework, PACE, each agent $k \in \{i, j\}$ attempts to learn its peer $-k$'s state cost parameters Q_{-k} to ultimately recover the complete information game. To achieve this, each agent updates its estimate about the other agent's objective function parameter, $\hat{Q}_{-k}$, based on the observed trajectory $\zeta(t)$ using a learning dynamic function f_k. A natural choice for f_k is a gradient descent function, modeling the agents as gradient learners. The approximation methods, which treat the peer agent as a complete information agent, assume that each agent only has access to its own f_k, ignoring the learning dynamics of their peer. In contrast, PACE models both agents as learning agents, assuming that each agent k has access to both learning dynamic functions, f_k and f_{-k}. The assumption that agents have access to each other's learning dynamics or are aware of each other's initial estimates is not limiting in many multi-agent planning scenarios, particularly when the goal is to plan for known agents or robots to work together effectively. In cases where the learning dynamic of the other agent is unknown, such as in some human-robot interaction scenarios, data-driven methods such as using transformers to model the human's (or any other agent) learning dynamics (Tian et al., 2023) can be used.

Online Cost Parameter Learning. We define $\hat{\theta}_{-k} = \text{vec}(\hat{Q}_{-k}) \in \mathbb{R}^{n^2}$, where $\text{vec}(\cdot)$ denotes the vectorization operator. In general, assuming the agents are gradient learners, we model each learning function as $\dot{\hat{\theta}}_{-k}^t = f_k(\hat{\theta}_{-k}^t, \mathcal{H}_k^t\{x, u_k, \hat{P}_{-k}, \hat{P}_k\})$. To infer $\hat{\theta}_{-k}$, knowing that agents cannot observe each other's control actions, we propose that agent k uses a likelihood function over the observed trajectory $\boldsymbol{\zeta}(t)$. As a result, agent k evaluates the accuracy of its estimated $\hat{u}_{-k}$, by assuming the state dynamics evolve from $t = \tau_1$ to the current time according to:

$$\dot{\hat{x}}_k(\tau) = A\hat{x}_k(\tau) + B_k u_k(\tau) + B_{-k}\hat{u}_{-k}(\tau), \quad \hat{x}_k(\tau_1) = x(\tau_1), \tag{8}$$

where $\hat{u}_{-k}(t)$ is computed according to the latest estimates of $\hat{\theta}_{-k}^t$, and $u_k(\tau)$ is obviously known to agent k from $\mathcal{H}_k$. The resulting sampled estimated trajectory, $\hat{\boldsymbol{\zeta}}_k = \{\hat{x}_k(\tau_1), \ldots, \hat{x}_k(\tau_N)\}$, is used to maximize the likelihood of observing $\boldsymbol{\zeta}$ by minimizing the negative log-likelihood:

$$\min_{\hat{\theta}_{-k}} -\log p(\boldsymbol{\zeta}|\hat{\boldsymbol{\zeta}}_k), \quad \text{s.t.} \quad \hat{Q}_{-k} > 0 \tag{9}$$

To approximate this minimization, we replace the negative log-likelihood with a squared error loss $\mathcal{L}_k(\hat{\theta}_{-k}) = \frac{1}{N}\sum_{\tau\in\{\tau_1,\ldots,\tau_N\}} \|x(\tau) - \hat{x}_k(\tau)\|^2$, which encourages alignment between the observed trajectory $\boldsymbol{\zeta}$ and the estimated trajectory $\hat{\boldsymbol{\zeta}}_k$. Each element $\hat{\theta}_{-k(i)}$ of the vector $\hat{\theta}_{-k}$ can be updated at each decision-making step via the gradient descent method as follows

$$\dot{\hat{\theta}}_{-k(i)}^{(t)} = f_k^i(\hat{\theta}_{-k(i)}^{(t)}; \mathcal{H}_k^t\{x, u_k, \hat{P}_{-k}, \hat{P}_k\}) = -\alpha \frac{\partial \mathcal{L}_k(\hat{\theta}_{-k(i)}^{(t)}; \mathcal{H}_k^t\{x, u_k, \hat{P}_{-k}, \hat{P}_k\})}{\partial \hat{\theta}_{-k(i)}^{(t)}}, \tag{10}$$

where α is a constant learning rate. For agent k to model the other agent's update rule, i.e., f_{-k}, agent k cannot use the same equation as (8) because they lack knowledge about $\mathcal{H}_{-k}$, as they are

unable to observe the other agent's actions at previous time steps. To address this issue, we note that agent k is aware of the fact that $-k$ (the other agent) is also trying to minimize the log-likelihood of the observed trajectory based on their prediction (similar to (9)). As a result, knowing that the observed trajectory error for $-k$ (the other agent) is due to their incorrect estimations in $\hat{u}_k$, agent k can use the following equation to form the error dynamics for modeling the trajectory minimization of $-k$:

$$\dot{e}_{-k}(\tau) = Ae_{-k}(\tau) + B_k(u_k(\tau) - \hat{u}_k(\tau)), \quad e_{-k}(\tau_1) = 0, \tag{11}$$

yielding $\mathcal{L}_{-k}(\hat{\theta}_k) = \frac{1}{N}\sum_{\tau\in\{\tau_1,\dots,\tau_N\}} \|e_{-k}\|^2$. This allows agent k to simulate $-k$'s learning with the gradient descent update, resulting in

$$\dot{\hat{\theta}}^{(t)}_{k(i)} = f^i_{-k}(\hat{\theta}^{(t)}_{k(i)}; \mathcal{H}^t_k\{x, u_k, \hat{P}_{-k}, \hat{P}_k\}) = -\alpha \frac{\partial \mathcal{L}_k(\hat{\theta}^{(t)}_{k(i)}; \mathcal{H}^t_k\{x, u_k, \hat{P}_{-k}, \hat{P}_k\})}{\partial \hat{\theta}^{(t)}_{k(i)}}. \tag{12}$$

As a result, each agent will have access to both f_k and f_{-k}. We have used $\mathcal{H}^t_k\{x, u_k, \hat{P}_{-k}, \hat{P}_k\}$ in (12) to emphasize that the information available in $\mathcal{H}^t_k$ is sufficient to construct f_{-k} in PACE.

Remark 1. If both agents agree on their initial estimates, the proposed update rules in (12) and (10) allow both agents to track and agree on the estimation pairs $(\hat{\theta}_k, \hat{\theta}_{-k})$ and $(\hat{P}_k, \hat{P}_{-k})$ at each step. This is an important property of PACE, which results in a centralized learning dynamic for both agents, meaning that both agents have access to the same $(\hat{\theta}_k, \hat{\theta}_{-k})$ all the time, although they are performing their estimation independently.

Despite being non-convex, the loss functions $\mathcal{L}_{-k}(\hat{\theta}^{(t)}_k)$ and $\mathcal{L}_k(\hat{\theta}^{(t)}_{-k})$ are differentiable with respect to $\hat{\theta}_k$ and $\hat{\theta}_{-k}$, as stated in Theorem 3.2 of (Laine et al., 2023). Consequently, the choice of gradient descent for the mentioned updates (12) and (10) is inspired by its demonstrated success in non-convex optimization problems (Boyd and Vandenberghe, 2004; Sutskever et al., 2013).

Equations (12) and (10) require computing the estimates $\hat{u}_{-k}(t)$ and $\hat{u}_k(t)$, as well as taking their derivatives. Building on the policy update shown in (7), we propose that agents update the estimation of others' control signals at each $t = \tau_i$ in the history as $\hat{u}_{-k}(t) = -B^\top_{-k} P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau_i)x(t)$ and $\hat{u}_k(t) = -B^\top_k P_k(\hat{\theta}^{(t)}_k, \tau_i)x(t)$. Here, $P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau_i)$ and $P_k(\hat{\theta}^{(t)}_k, \tau_i)$ can be obtained by solving the following equations for both $k \in \{i, j\}$ at each τ_i in the history stack $\mathcal{H}^t_k$:

$$\begin{aligned} 0 = &\left(A - B_k B^\top_k \hat{P}^{(\tau)}_k\right)^\top P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau) + P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau)\left(A - B_k B^\top_k \hat{P}^{(\tau)}_k\right) \\ &- P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau) B_{-k} B^\top_{-k} P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau) + \hat{Q}_{-k}(\hat{\theta}^{(t)}_{-k}), \quad \text{for } k \in \{i, j\},\ \tau \in \{\tau_1, \dots, \tau_N\}. \end{aligned} \tag{13}$$

Consequently, taking derivative from the estimated control signals requires knowing the sensitivity of $P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau)$ to each $\hat{\theta}^{(t)}_{-k(i)}$ by finding $\partial P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau)/\partial \hat{\theta}^{(t)}_{-k(i)}$ through (13). This sensitivity can be obtained by solving a Lyapunov equation with a state matrix $A_k(\tau) = A - B_k B^\top_k P^{(\tau)}_k - B_{-k} B^\top_{-k} P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau)$, where $A_k(\tau)$ is a stable Hurwitz matrix since $P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau)$ is the stabilizing solution of the Riccati equation (13)), ensuring that $\partial P_{-k}(\hat{\theta}^{(t)}_{-k}, \tau)/\partial \hat{\theta}^{(t)}_{-k(i)}$ exists.

Remark 2. Equation (13) represents agent k's estimation of agent $-k$'s policy update equation based on (7). Agent k knows that $-k$ uses $\hat{P}^{(\tau)}_k$ in the coupling term of their Riccati equation. If one (wrongly) assumes the other agent has complete infromation, agent k replaces the coupling

term $\hat{P}_k(\tau)$ in (13) with their true matrix $P_k^{(\tau)}$ during parameter learning, assuming the other agent $-k$ is optimal and has complete information about Q_k and $P_k^{(\tau)}$.

Remark 3. Although PACE is developed for differential games, like many other algorithms, practically it needs to be implemented in discrete time. As a result, Algorithm 1 and the subsequent discussion on parameter and policy updates are presented in a discrete-time format for clarity.

Prediction and Updating the History Stack. With the updated parameters $\hat{\theta}_{-k}^{(t+1)}$ and $\hat{\theta}_k^{(t+1)}$, agents can, at their next decision-making step, solve a new set of coupled Riccati equations to obtain $\hat{P}_k^{(t+1)}$ and $\hat{P}_{-k}^{(t+1)}$, which are then used in the policy update (7) as predictions of the other agent's gains while also updating the history stack.

Algorithm 1: PACE for Agent k

Initialize: $\hat{Q}_{-k}(0)$, $\hat{Q}_k(0)$ and their corresponding Riccati solutions $\hat{P}_k^{(0)}$ and $\hat{P}_{-k}^{(0)}$ that stabilize the closed loop system, empty history stack $\mathcal{H}_k$

for *each time step* t **do**

1. **Observe** $x(t)$, compute $u_k(t) = -B_k^\top P_k^{(t)} x(t)$, apply $u_k(t)$, and update history stack by adding $(x(t), u_k(t), \hat{P}_k^{(t)}, \hat{P}_{-k}^{(t)})$; maintain stack size N by removing the oldest data point if necessary
2. **Trajectory Generation: for** *each* τ *in* $\mathcal{H}_k$ **do**
   - **a.** Solve (13) to find $P_{-k}(\hat{\theta}_{-k}^{(t)}, \tau)$ for both $k \in \{i, j\}$ using $\hat{P}_k^{(\tau)}$ and $\hat{P}_{-k}^{(\tau)}$ from $\mathcal{H}_k$ and the current estimates $\hat{Q}_{-k}$ and $\hat{Q}_k$
   - **b.** Set $\hat{u}_{-k}(\tau) = -B_{-k}^\top P_{-k}(\hat{\theta}_{-k}^{(t)}, \tau) x(\tau)$ and $\hat{u}_k(\tau) = -B_k^\top P_k(\hat{\theta}_k^{(t)}, \tau) x(\tau)$
   - **c.** Sample the point $\hat{x}_k(\tau)$ using (8)
   - **d.** Sample the point $\hat{e}_k(\tau)$ using (11)

   end
3. **Form losses** $\mathcal{L}_k(\hat{\theta}_{-k}^{(t)}) = \frac{1}{N} \sum_{\tau \in \{\tau_1 \dots \tau_N\}} \|x(\tau) - \hat{x}_k(\tau)\|^2$ and $\mathcal{L}_{-k}(\hat{\theta}_k^{(t)}) = \frac{1}{N} \sum_{\tau \in \{\tau_1, \dots, \tau_N\}} \|e_{-k}\|^2$
4. **Parameter Update: for** *each* $\hat{\theta}_{-k(i)}^{(t)}$ *in* $\hat{\theta}_{-k}^{(t)}$ *and* $\hat{\theta}_{k(i)}^{(t)}$ *in* $\hat{\theta}_k^{(t)}$ **do**
   - **a.** Compute $\frac{\partial \mathcal{L}_k}{\partial \hat{\theta}_{-k}^i}$ and $\frac{\partial \mathcal{L}_{-k}}{\partial \hat{\theta}_k^i}$ by calculating all $\frac{\partial \hat{x}_k(\tau)}{\partial \hat{\theta}_{-k}^{(i)}}$ and $\frac{\partial \hat{e}_{-k}(\tau)}{\partial \hat{\theta}_k^{(i)}}$ and derivation through (13)
   - **b.** Update $\hat{\theta}_{-k(i)}^{(t+1)} = \hat{\theta}_{-k(i)}^{(t)} - \alpha \frac{\partial \mathcal{L}_k}{\partial \hat{\theta}_{-k(i)}^{(t)}}$ and $\hat{\theta}_k^{(t+1)} = \hat{\theta}_{k(i)}^{(t)} - \alpha \frac{\partial \mathcal{L}_{-k}}{\partial \hat{\theta}_{k(i)}^{(t)}}$

   end
5. **Prediction:** Solve coupled Riccati equations corresponding to $\hat{\theta}_{-k}^{(t+1)}$ and $\hat{\theta}_k^{(t+1)}$ to predict $\hat{P}_{-k}^{(t+1)}$.
6. **Policy Update:** using $\hat{P}_{-k}^{(t+1)}$ update the policy by updating $P_k^{(t+1)}$ from (7)

end

Theorem 1 *If two agents begin with initial guesses for each other's cost parameters that yield admissible policies, then under a sufficiently small learning rate α and a persistently exciting system state signal in the history stack $\mathcal{H}_k$,* ***Algorithm 1 (PACE)*** *converges to the true cost parameters exponentially, while maintaining system stability until reaching the Nash equilibrium.*

Proof [See Appendix.A of the full version of this work (Soltanian and Zhang, 2025)] ■

The convergence speed of Algorithm 1 depends on the choice of the learning rate, and the number of samples in the history stack. Increasing the number of observations in the history stack can improve the convergence speed, while decreasing the upper bound on the learning rate. For more discussion on the effect of learning rate and history stack see Appendix.B of the full version of this work (Soltanian and Zhang, 2025).

5. Numerical Experiments

In this section, we evaluate PACE through two examples. First, we perform an ablation study of a shared driving task to compare PACE with complete information peer approximation methods. Second, we examine multi-parameter estimation in a repetitive task involving physical interaction between a human agent and a robotic arm's end effector. The codes are available at Github.

Example 1: Monte Carlo Study in Shared Steering Driving

The vehicle dynamics for the shared steering system is represented by the state vector $x(t) = [e_1, \dot{e}_1, e_2, \dot{e}_2]^T$, where e_1 and e_2 denote the lateral position and orientation errors, respectively. The system dynamics is governed by the model proposed in (Rajamani, 2011), with details of the matrices $A \in \mathbb{R}^{4\times4}$ and $B \in \mathbb{R}^{4\times1}$ available in the Appendix.C of (Soltanian and Zhang, 2025). The control inputs $u_1(t)$ and $u_2(t)$ represent the contributions from the human driver and the machine, respectively. The cost parameter matrices are defined as $Q_1 = \theta_1 \cdot \text{diag}(1, 0.5, 0.5, 0.25)$ and $Q_2 = \theta_2 \cdot \text{diag}(1, 2, 1, 0.5)$, where $\theta_2 = 1$ and $\theta_1 = 2$. These parameters are unknown to the other agent and need to be estimated. During the simulation, we chose a sampling time of 0.01 seconds.

In our first Monte Carlo study, we compared PACE with the optimal peer approximation method by modifying our learning algorithm as described in **Remark 2** (to have a fair comparison). We ran 500 simulations for both algorithms, varying the initial guesses for $\hat{\theta}_k^{(0)}$ by sampling uniformly from the range $[0, 10]$. The learning rate and history size are kept constant across all simulations, with $\alpha = 0.15$ and $N = 35$. The results are shown in Fig. 2 Left, where the system with PACE converged to at least 80% of the true values in all 500 simulations, regardless of the initial guesses. In contrast, 85% of the optimal peer approximation simulations, particularly those with larger initial errors for $\hat{\theta}_k^{(0)}$, failed to converge to the true values and even became unstable during the interaction. PACE also demonstrated faster convergence, highlighting its robustness and advantage over the peer approximation method.

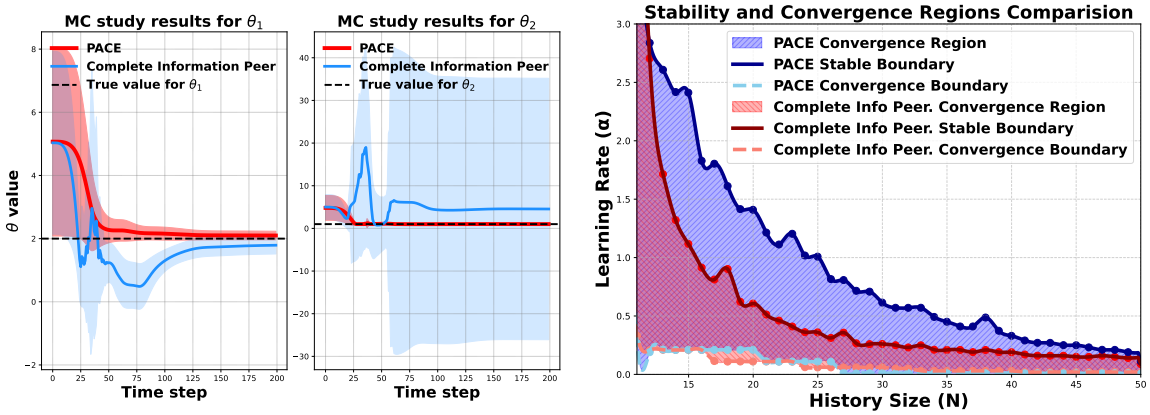

Figure 2: Monte Carlo study results of 500 random guesses for agents' initial estimates, $\hat{\theta}_k^{(0)}$ and $\hat{\theta}_{-k}^{(0)}$ (left); stability region analysis comparing PACE with the complete info peer approximation, showing how increasing the learning rate for each history size affect the stability boundaries(right).

In our second study, we fixed the initial guess $\hat{\theta}_k^{(0)}$ as a near-zero random value and varied the history size N from 1 to 50. For each N, we increased the learning rate incrementally, recording

two critical points for each N: the smallest learning rate where convergence to 80% of the true value was achieved (convergence boundary) and the learning rate at which instability occurred (stability boundary). Spline fitting visualized the convergence regions in Figure 2 Right. The results show PACE has a wider convergence region than the optimal peer approximation method and faster convergence. Intuitively, this result makes sense as higher learning rates make agents faster learners and more nonstationary, while larger history sizes include more outdated data.

Example 2: Human-Robot Co-manipulation

Inspired by (Li et al., 2019), we examine human-robot collaboration to move a robot arm's end effector between $x_d = -10\,\text{cm}$ and $x_d = 10\,\text{cm}$ every 2 seconds. The system states are $x(t) = [e(t), \dot{e}(t)]^T$, where $e(t) = x(t) - x_d$ is the position error and $\dot{e}(t)$ the velocity. The system dynamics and baseline control/estimation algorithms follow (Li et al., 2019), with a sampling time of 0.01 seconds. The cost matrices are $Q_H = \text{diag}(100, 25)$ for the human and $Q_R = \text{diag}(75, 50)$ for the robot. Unlike Example 1, this example involves multi-parameter estimation for all diagonal elements of these Q matrices. The initial state is $x(0) = [0, 0]^T$. Both agents estimate the other's cost parameters using PACE and the complete information peer approximation, with a learning rate $\alpha = 0.1$ and history size $N = 15$. Figure 3 compares PACE with the adaptive control-based method in (Li et al., 2019) (baseline) and the complete information peer approximation described in **Remark 2**. PACE achieves faster convergence and reduced overshoot. Its monotonic convergence, except when x_d changes sign, is evident in Fig. 3, outperforming the baseline method. This is crucial for tasks like autonomous vehicle planning where accurate intent estimation is critical (Amatya et al., 2022). More detailed results for this experiment and its model can be found in Appendix.D of the full version of this work (Soltanian and Zhang, 2025).

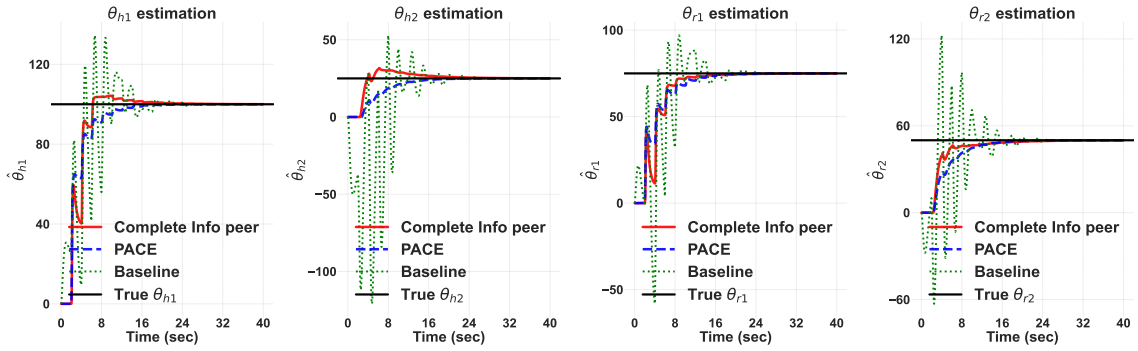

Figure 3: Comparison of three algorithms for multi-parameter estimation in human-robot co-manipulation.

6. Conclusion and Future Work

In this paper, we highlighted the critical role of modeling the learning dynamics of peer agents in incomplete-information differential games. We demonstrated that PACE outperforms methods that approximate the other agent as a complete information agent in terms of robustness, stability, monotonic convergence, theoretical guarantees, and convergence speed. PACE's applicability extends to time-varying linear systems and holds promise for adaptation to nonlinear general-sum games using iterative LQR methods in subsequent research and to be applied in HRI applications.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1944833. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Also, we thank Dr. Changliu Liu from Carnegie Mellon University for her valuable inputs on this paper.

References

Sunny Amatya, Mukesh Ghimire, Yi Ren, Zhe Xu, and Wenlong Zhang. When shall i estimate your intent? costs and benefits of intent inference in multi-agent interactions. In *2022 American Control Conference (ACC)*, pages 586–592. IEEE, 2022.

Robert J Aumann, Michael Maschler, and Richard E Stearns. *Repeated games with incomplete information*. MIT press, 1995.

Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Alberto Bressan. Noncooperative differential games. a tutorial. *Department of Mathematics, Penn State University*, 81, 2010.

Lorenzo Canese, Gian Carlo Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spanò. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11):4948, 2021.

Pierre Cardaliaguet and Catherine Rainer. Games with incomplete information in continuous time and for continuous types. *Dynamic Games and Applications*, 2:206–227, 2012.

Fei Chen, Wei Ren, et al. On the control of multi-agent systems: A survey. *Foundations and Trends® in Systems and Control*, 6(4):339–499, 2019.

Yi Chen, Lei Zhang, Tanner Merry, Sunny Amatya, Wenlong Zhang, and Yi Ren. When shall i be empathetic? the utility of empathetic parameter estimation in multi-agent interactions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2761–2767. IEEE, 2021.

Lynda Cherfi, Yacine Chitour, and Hisham Abou-Kandil. A new algorithm for solving coupled algebraic riccati equations. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 1, pages 83–88. IEEE, 2005.

Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

Jacob C Engwerda. On scalar feedback nash equilibria in the infinite horizon lq-game. *IFAC Proceedings Volumes*, 31(16):193–198, 1998.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

P Franceschi, N Pedrocchi, and M Beschi. Identification of human control law during physical human–robot interaction. *Mechatronics*, 92:102986, 2023.

David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D Dragan, and Claire J Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 1475–1481. IEEE, 2020.

Jairo Inga, Esther Bischoff, Timothy L Molloy, Michael Flad, and Sören Hohmann. Solution sets for inverse non-cooperative linear-quadratic differential games. *IEEE Control Systems Letters*, 3 (4):871–876, 2019.

Alexis Jacq, Matthieu Geist, Ana Paiva, and Olivier Pietquin. Learning from a learner. In *International Conference on Machine Learning*, pages 2990–2999. PMLR, 2019.

Xuewu Ji, Kaiming Yang, Xiaoxiang Na, Chen Lv, and Yahui Liu. Shared steering torque control for lane change assistance: A stochastic game-theoretic approach. *IEEE Transactions on Industrial Electronics*, 66(4):3093–3105, 2018.

Rushikesh Kamalapurkar, Benjamin Reish, Girish Chowdhary, and Warren E Dixon. Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Transactions on Automatic Control*, 62(7):3594–3601, 2017.

Forrest Laine, David Fridovich-Keil, Chih-Yuan Chiu, and Claire Tomlin. Multi-hypothesis interactions in game-theoretic motion planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8016–8023. IEEE, 2021.

Forrest Laine, David Fridovich-Keil, Chih-Yuan Chiu, and Claire Tomlin. The computation of approximate generalized feedback nash equilibria. *SIAM Journal on Optimization*, 33(1):294–318, 2023.

Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. Lucidgames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning. *IEEE Robotics and Automation Letters*, 6(3):5485–5492, 2021.

Jingqi Li, Chih-Yuan Chiu, Lasse Peters, Somayeh Sojoudi, Claire Tomlin, and David Fridovich-Keil. Cost inference for feedback dynamic games from noisy partial state observations and incomplete trajectories. *arXiv preprint arXiv:2301.01398*, 2023.

Jingqi Li, Anand Siththaranjan, Somayeh Sojoudi, Claire Tomlin, and Andrea Bajcsy. Intent demonstration in general-sum dynamic games via iterative linear-quadratic approximations. *arXiv preprint arXiv:2402.10182*, 2024.

TY Li and Z Gajic. Lyapunov iterations for solving coupled algebraic riccati equations of nash differential games and algebraic riccati equations of zero-sum games. In *New Trends in Dynamic Games and Applications*, pages 333–351. Springer, 1995.

Yanan Li, Keng Peng Tee, Rui Yan, Wei Liang Chan, and Yan Wu. A framework of human–robot coordination based on game theory and policy iteration. *IEEE Transactions on Robotics*, 32(6): 1408–1418, 2016.

Yanan Li, Gerolamo Carboni, Franck Gonzalez, Domenico Campolo, and Etienne Burdet. Differential game theory for versatile physical human–robot interaction. *Nature Machine Intelligence*, 1(1):36–43, 2019.

Bosen Lian, Vrushabh S Donge, Frank L Lewis, Tianyou Chai, and Ali Davoudi. Data-driven inverse reinforcement learning control for linear multiplayer games. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):2028–2041, 2022.

Jie Lin, Mi Wang, and Huai-Ning Wu. Composite adaptive online inverse optimal control approach to human behavior learning. *Information Sciences*, 638:118977, 2023.

Changliu Liu, Wenlong Zhang, and Masayoshi Tomizuka. Who to blame? learning and control strategies with information asymmetry. In *2016 American Control Conference (ACC)*, pages 4859–4864. IEEE, 2016.

Dylan P Losey, Craig G McDonald, Edoardo Battaglia, and Marcia K O'Malley. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Applied Mechanics Reviews*, 70(1):010804, 2018.

Dahlard L Lukes and David L Russell. A global theory for linear-quadratic differential games. *Journal of Mathematical Analysis and Applications*, 33(1):96–123, 1971.

Negar Mehr, Mingyu Wang, Maulik Bhatt, and Mac Schwager. Maximum-entropy multi-agent dynamic games: Forward and inverse solutions. *IEEE transactions on robotics*, 39(3):1801–1815, 2023.

Timothy L Molloy, Jason J Ford, and Tristan Perez. Online inverse optimal control for control-constrained discrete-time systems on finite and infinite horizons. *Automatica*, 120:109109, 2020.

Timothy L Molloy, Jairo Inga Charaja, Sören Hohmann, and Tristan Perez. *Inverse optimal control and inverse noncooperative dynamic game theory*. Springer, 2022.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

Lasse Peters, David Fridovich-Keil, Vicenç Rubies-Royo, Claire J Tomlin, and Cyrill Stachniss. Inferring objectives in continuous dynamic games from noise-corrupted partial state observations. *arXiv preprint arXiv:2106.03611*, 2021.

Corrado Possieri and Mario Sassano. An algebraic geometry approach for the computation of all linear feedback nash equilibria in lq differential games. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5197–5202. IEEE, 2015.

Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

M Cody Priess, Richard Conway, Jongeun Choi, John M Popovich, and Clark Radcliffe. Solutions to the inverse lqr problem with application to biological systems analysis. *IEEE Transactions on control systems technology*, 23(2):770–777, 2014.

Mahdis Rabbani, Navid Mojahed, and Shima Nazari. Optimal modified feedback strategies in lq games under control imperfections. *arXiv preprint arXiv:2503.19200*, 2025.

Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017.

Sydney Y Schaefer, Iris L Shelly, and Kurt A Thoroughman. Beside the point: motor adaptation without feedback-based error correction in task-irrelevant conditions. *Journal of Neurophysiology*, 107(4):1247–1256, 2012.

Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50): 24972–24978, 2019.

Ryan Self, Moad Abudia, SM Nahid Mahmud, and Rushikesh Kamalapurkar. Model-based inverse reinforcement learning for deterministic systems. *Automatica*, 140:110242, 2022.

Seyed Yousef Soltanian and Wenlong Zhang. Pace: A framework for learning and control in linear incomplete-information differential games, 2025. URL https://arxiv.org/abs/2504.17128.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

Ran Tian, Masayoshi Tomizuka, Anca D Dragan, and Andrea Bajcsy. Towards modeling and influencing the dynamics of human learning. In *Proceedings of the 2023 ACM/IEEE international conference on human-robot interaction*, pages 350–358, 2023.

Kyriakos G Vamvoudakis and Frank L Lewis. Multi-player non-zero-sum games: Online adaptive learning solution of coupled hamilton–jacobi equations. *Automatica*, 47(8):1556–1569, 2011.

Yiwei Wang, Yi Ren, Steven Elliott, and Wenlong Zhang. Enabling courteous vehicle interactions through game-based and dynamics-aware intent inference. *IEEE Transactions on Intelligent Vehicles*, 5(2):217–228, 2019.

Yiwei Wang, Pallavi Shintre, Sunny Amatya, and Wenlong Zhang. Bounded rational game-theoretical modeling of human joint actions with incomplete information. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10720–10725. IEEE, 2022.

Huai-Ning Wu, Xiao-Yan Jiang, and Mi Wang. Human cognitive learning in shared control via differential game with bounded rationality and incomplete information. *IEEE Transactions on Artificial Intelligence*, 1(01):1–12, 2024.

Wenqian Xue, Patrik Kolaric, Jialu Fan, Bosen Lian, Tianyou Chai, and Frank L Lewis. Inverse reinforcement learning in tracking control based on inverse optimal control. *IEEE Transactions on Cybernetics*, 52(10):10570–10581, 2021.

Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

Chengpu Yu, Yao Li, Shukai Li, and Jie Chen. Inverse linear quadratic dynamic games using partial state observations. *Automatica*, 145:110534, 2022.

Han Zhang and Axel Ringh. Inverse optimal control for averaged cost per stage linear quadratic regulators. *Systems & Control Letters*, 183:105658, 2024.

Lei Zhang, Mukesh Ghimire, Wenlong Zhang, Zhe Xu, and Yi Ren. Approximating discontinuous nash equilibrial values of two-player general-sum differential games. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3022–3028. IEEE, 2023.

Lei Zhang, Mukesh Ghimire, Zhe Xu, Wenlong Zhang, and Yi Ren. Pontryagin neural operator for solving general-sum differential games with parametric state constraints. In *6th Annual Learning for Dynamics & Control Conference*, pages 1728–1740. PMLR, 2024.