Robust Inverse Reinforcement Learning Control with Unknown States

Bosen Lian BZL0098@AUBURN.EDU
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849

Wenqian Xue W.XUE@UFL.EDU
Department of Mechanical and Aerospace, University of Florida, Gainesville, FL 32603

Nhan Nguyen NHAN.T.NGUYEN@NASA.GOV
Intelligent Systems Division, NASA Ames Research Center, Moffett Field, CA 94035

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

Abstract

This paper designs a robust inverse reinforcement learning (IRL) algorithm that observes the expert's inputs and outputs to reconstruct the underlying cost function weights and optimal control policy for optimal discrete-time (DT) output feedback (OPFB) control systems while admitting disturbances and unknown states. The expert system is captured by a zero-sum game where its OPFB controller minimizes a cost function while robustly mitigating the effect of the worst disturbance, achieving a prescribed attenuation level. The inputs and outputs of the expert can be observed, but not the states. To enable the learner to replicate the behavior of the expert, we first develop a model-based IRL algorithm and subsequently design an equivalent model-free, data-driven version. This latter infers the quadratic cost function weights that can yield the expert's static OPFB control policy, using output and input data of both the expert and learner. The convergence of the proposed algorithms is rigorously validated through theoretical analysis and numerical experiments.

Keywords: Inverse reinforcement learning, output-feedback control, discrete-time zero-sum games

1. Introduction

IRL Ng et al. (2000); Lian et al. (2024) infers the objective/cost/reward function (which is often unknown or difficult to specify explicitly) that best explains observed expert behavior, captures the underlying goals or motivations, and interprets an agent's learned behavior, making it easier to verify and trust. By optimizing the inferred objective function, learning agents can adapt to new or unseen situations, even when the exact expert behavior in those contexts was not observed.

Initially developed for Markov decision processes (MDPs) with discrete state and action spaces Ng et al. (2000); Ziebart et al. (2008); Hadfield-Menell et al. (2016); Ramachandran and Amir (2007), IRL cannot be directly applied to when applied to dynamic control systems. These systems, such as robotic or aircraft control, often operate in continuous state and action spaces and are governed by differential or difference equations Bishop (2011); Lewis et al. (2012). Such systems require rigorous consideration of control solution existence, stability, and the influence of system dynamics. Motivated by these challenges, prior works Aghasadeghi and Bretl (2011); Tesfazgi et al. (2021); Lian et al. (2024); Cao and Xie (2022); Town et al. (2025); Wulfmeier et al. (2017) have extended IRL to dynamic control problems, reconstructing objective functions and optimal control policies for demonstrated behaviors in systems such as zero-sum games Lian et al. (2023b); Xue et al. (2024), non-zero-sum games Lian et al. (2022); Martirosyan and Cao (2024), and multi-agent graphical games Donge et al. (2022); Golmisheh and Shamaghdari (2024).

Real-world control systems frequently face disturbances. Zero-sum differential or difference games Al-Tamimi et al. (2007); Rizvi and Lin (2018); Li et al. (2023); Valadbeigi et al. (2019) are valuable for designing robust and optimal control policies that stabilize systems and mitigate the effects of these disturbances while optimizing objectives such as minimizing cost or energy consumption. Zero-sum games often provide optimal control solutions, presenting a natural setting for applying IRL. However, the existing IRL method for DT zero-sum games in Lian et al. (2023b) assumes state feedback for optimal control policy design. Measuring states is often complex for applications, including aerospace systems and industrial process control. This limitation motivates the study of IRL for DT zero-sum games using output information with unknown states.

Unknown parameters in dynamic systems further challenge controller design and system stability. While inverse optimal control methods in Molloy et al. (2022); Jin et al. (2021); Wu (2022) can identify objective functions for dynamic systems, these approaches typically require explicit system dynamics and are often designed offline. Model-free IRL controls have been developed in Xue et al. (2022); Lian et al. (2023b,b) to infer the objective function and optimal control policy of behaviors in discrete-time optimal control systems without knowing system dynamics. Building on our prior work on model-free IRL of state-feedback zero-sum games, this study seeks model-free IRL for output-feedback zero-sum games.

This paper's contributions are threefold. First, it investigates IRL for DT static output-feedback (OPFB) zero-sum games without knowing system states. This extends IRL in Lian et al. (2023a) for DT OPFB control to scenarios involving uncontrollable disturbances and also revolutes continuous-time OPFB IRL Xue et al. (2024) to the discrete-time domain. Second, a model-based IRL policy iteration approach is designed, and a final model-free, data-driven IRL algorithm is derived, leveraging measured inputs and outputs for cost function and optimal control policy reconstruction. Third, the algorithm's convergence is guaranteed.

Notations. ceil$\{\}$ defines the smallest integer greater than or equal to a given number. $\|\cdot\|$ defines the Euclidean norm. I defines an identity matrix. $\mathbf{0}$ denotes a zero-element vector or matrix. $S > \mathbf{0}$ ($\geq \mathbf{0}$), $S \in \mathcal{R}^{n\times n}$ denotes that $S = S^{\mathrm{T}}$ is positive definite or positive semi-definite. For $S, M \in \mathcal{R}^{n\times n}$, $S > M$ ($\geq M$) represents that $S - M > \mathbf{0}$ or $S - M \geq \mathbf{0}$.

2. Problem Formulation

This section presents a learner system described by a DT static OPFB-based zero-sum game with trail cost function weights and an expert system with identical dynamics and the desired but unknown cost function weights. Then, an IRL problem will be formulated.

2.1. OPFB-Based DT Zero-Sum Games

Consider a learner system with the following dynamics

$$x_{k+1} = Ax_k + Bu_k + D\omega_k, \;\; y_k = Cx_k, \tag{1}$$

where $x_k \in \mathcal{R}^n$, $u_k \in \mathcal{R}^m$, $\omega_k \in \mathcal{R}^l$, $y_k \in \mathcal{R}^p$ are states, control inputs, disturbance, and outputs, respectively. $A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times m}$, $D \in \mathcal{R}^{n\times l}$, and $C \in \mathcal{R}^{p\times n}$ are constant dynamics matrices.

Assumption 1 *The pair (A, B) is controllable, (A, C) is observable, and C has full row rank.*

Definition 1 *System (1) is called OPFB stabilizable if there is a K such that $A - BKC$ is stable.*

The learner's control input minimizes the following trial cost function while the disturbance maximizes it

$$V(x_k, u_k, w_k) = \sum_{j=k}^{\infty} \left(y_j^{\mathrm{T}} Q y_j + u_j^{\mathrm{T}} R u_j - \gamma^2 \omega_j^{\mathrm{T}} \omega_j \right), \tag{2}$$

where $Q > \mathbf{0}$, $Q \in \mathcal{R}^{p\times p}$ and $R > \mathbf{0}$, $R \in \mathcal{R}^{m\times m}$ are *learner's penalty weights* on states and control inputs; $\gamma > 0$ is a antagonist attenuation level on disturbance; and $(A, \sqrt{Q}C)$ is observable.

Definition 2 *System* (1) *is* L_2*-gain bounded by* γ^2 *if* $\sum_{k=0}^{\infty} \left(y_k^T Q y_k + u_k^T R u_k - \gamma^2 \omega_k^T \omega_k \right) < 0$.

The optimized cost value takes the form of $V(x_k) = \min_{u_k} \max_{\omega_k} V = x_k^{\mathrm{T}} P x_k$ with $P > 0$, which is a zero-sum game and solves disturbance-rejection H_∞ control Başar and Olsder (1998); Lewis et al. (2012). System (1) is L_2-gain bounded by γ^2 by using the Nash policy solution

$$u_k = -K y_k, \;\; \omega_k = -K_\omega x_k, \tag{3}$$

where K and K_ω are given by

$$\begin{aligned} K = {} & (R + B^{\mathrm{T}} P B + B^{\mathrm{T}} P D (\gamma^2 - D^{\mathrm{T}} P D)^{-1} D^{\mathrm{T}} P B)^{-1} \\ & \times (-B^{\mathrm{T}} P A - B^{\mathrm{T}} P D (\gamma^2 I - D^{\mathrm{T}} P D)^{-1} D^{\mathrm{T}} P A + M) C^{\mathrm{T}} (C C^{\mathrm{T}})^{-1}, \end{aligned} \tag{4a}$$

$$\begin{aligned} K_\omega = {} & (D^{\mathrm{T}} P D - \gamma^2 I - D^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P D)^{-1} \\ & \times (-D^{\mathrm{T}} P A - D^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A), \end{aligned} \tag{4b}$$

with $P > \mathbf{0}$ and M satisfying the DT algebraic Riccati equation

$$\begin{aligned} \mathbf{0} = {} & A^{\mathrm{T}} P A - P + C^{\mathrm{T}} Q C + \begin{bmatrix} M^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} R + B^{\mathrm{T}} P B & B^{\mathrm{T}} P D \\ D^{\mathrm{T}} P B & \gamma^2 I - D^{\mathrm{T}} P D \end{bmatrix} \begin{bmatrix} M \\ \mathbf{0} \end{bmatrix} \\ & - \begin{bmatrix} A^{\mathrm{T}} P B & A^{\mathrm{T}} P D \end{bmatrix} \begin{bmatrix} R + B^{\mathrm{T}} P B & B^{\mathrm{T}} P D \\ D^{\mathrm{T}} P B & \gamma^2 I - D^{\mathrm{T}} P D \end{bmatrix} \begin{bmatrix} B^{\mathrm{T}} P A \\ D^{\mathrm{T}} P A \end{bmatrix}, \end{aligned} \tag{5}$$

and the equivalent Lyapunov equation

$$\begin{aligned} \mathbf{0} = {} & (A - BKC - DK_\omega)^{\mathrm{T}} P (A - BKC - DK_\omega) - P + C^{\mathrm{T}} Q C \\ & + (KC)^{\mathrm{T}} R K C - \gamma^2 K_\omega^{\mathrm{T}} K_\omega. \end{aligned} \tag{6}$$

Assumption 2 *Let* M *in* (5) *satisfy*

$$\begin{aligned} & C^T (\hat{K} - K)^T (R + B^T P B + B^T P D (\gamma^2 I - D^T P D)^{-1} D^T P B) (\hat{K} - K) C \\ & + (M)^T (\hat{K} - K) C + C^T (\hat{K} - K)^T M \geq \boldsymbol{0}, \end{aligned} \tag{7}$$

where $\hat{K} \neq K$ *is any stabilizing OPFB control policy.*

This assumption helps prove that K in (4a) and K_ω in (4b) will constitute a saddle-point solution in the zero-sum game. The following lemma ensures the solution to (6) and the stability of system (1).

Lemma 3 *With Assumptions 1 and 2. The $P > 0$ is the solution of (5) and (1) is OPFB stabilizable with the L_2 gain bounded by γ^2 and K in (4a) and K_ω in (4b) will constitute a saddle-point solution.*

Proof Please refer to the proofs of Theorems 1-2 in Valadbeigi et al. (2019). ■

Definition 4 ***(Expert System)*** *The agent is said to be an expert if it follows identical derivation from (1)-(7) with the cost function weights $Q_e > \mathbf{0}$, $Q_e \in \mathcal{R}^{p\times p}$, $R_e > \mathbf{0}$, $R_e \in \mathcal{R}^{m\times m}$, and $\gamma_e > 0$ in (2) instead of (Q, R, γ). The expert has the optimal OPFB control gain K_e in the form of (4a), and the corresponding (M_e, P_e) satisfies (5). The expert exhibits the behaviors $(x_{ek}, u_{ek}, y_{ek}, \omega_{ek})$.*

2.2. Robust Inverse RL Problem with Unknown States

Based on the above, we will formulate a robust inverse RL problem, i.e., only use the expert's (u_{ek}, y_{ek}) and learner's (y_k, u_k, ω_k) to reconstruct the underlying penalty weights that capture the static OPFB policy K_e, while both agents are subject to external disturbances.

Assumption 3 *The dynamics (A, B, C, D) and the expert's control parameters $(Q_e, R_e, \gamma_e, K_e)$ are unknown. The states x_k and x_{ek} of both the expert and learner are unmeasurable but observable.*

Assumption 4 *The learner's behavior data (u_k, ω_k, y_k) is available. The expert's behavior data (y_{ek}, u_{ek}) is available during $k \in [\underline{k}, \bar{k}]$. The expert's disturbance ω_{ke} is not required.*

Definition 5 *Provided a $R > \mathbf{0}$, $R \in \mathcal{R}^{m\times m}$ and a $\gamma > 0$, if there exists a $Q > \mathbf{0}$, $Q \in \mathcal{R}^{p\times p}$ in (2) such that (Q, R, γ) derives $K = K_e$ through (4)-(6), we call the Q an equivalent weight to Q_e with the corresponding (R, M, P).*

Assumption 5 *The γ in Definition 5 is large enough such that $\gamma > \underline{\gamma} > 0$, where $\underline{\gamma}$ is the minimum attenuation level associated with (Q, R). This ensures the existence of solution P in (5) Başar and Olsder (1998); Lewis et al. (2012).*

Definition 6 *A matrix or vector X is said to be approximately equal to X^* if $\|X - X^*\| \leq e$ holds with an acceptably small threshold $e > 0$.*

Problem 1 *Suppose Assumptions 1-4 hold. Arbitrarily selecting a $R > \mathbf{0}$, $R \in \mathcal{R}^{m\times m}$ and a large $\gamma > 0$, the goal is to find an output-penalty weight $Q \in \mathcal{R}^{p\times p}$ equivalent to Q_e.*

3. Model-Free Robust Input-Output Inverse RL

To solve Problem 1, this section designs an online data-driven model-free IRL algorithm that uses inputs and outputs, i.e., the expert's (u_{ek}, y_{ek}) and learner's (y_k, u_k, ω_k) without using states of either. An offline model-based policy iteration is provided as a base.

3.1. Offline Model-Based Inverse RL

The next theorem formulates the solution to Problem 1 (i.e., Q) such that $K = K_e$.

Theorem 7 *Suppose Assumptions 1-2 hold. Select $R > \mathbf{0}$ and a large $\gamma > 0$, if $P > \mathbf{0}$ satisfies the Lyapunov equation (6) where K and K_ω follow the forms in (4), M satisfies*

$$\begin{aligned} M = &(R + B^T P B + B^T P D(\gamma^2 I - D^T P D)^{-1} D^T P B) K C \\ &+ B^T P D(\gamma^2 I - D^T P D)^{-1} D^T P A + B^T P A, \end{aligned} \tag{8}$$

and $C^T Q C$ is selected as

$$\begin{aligned} C^T Q C = &P - (A - BKC - DK_\omega)^T P (A - BKC - DK_\omega) - (KC)^T R K C \\ &+ \gamma^2 K_\omega^T K_\omega + (KC - K_e C)^T R (KC - K_e C), \end{aligned} \tag{9}$$

then one obtains $K = K_e$, and Q is thus an equivalent weight to Q_e.

Proof Subtract (6) from (9) yields $\mathbf{0} = (KC - K_e C)^{\mathrm{T}} R (KC - K_e C)$, which implies $K = K_e$ because C is of full row rank. This completes the proof. ■

Considering Theorem 7, we propose the following offline algorithm to find its mentioned Q.

Algorithm 1 Inverse Policy Iteration for Weight Improvement

1: Select initial $Q^0 > \mathbf{0}$, $Q^0 \in \mathbb{R}^{p \times p}$, $R > \mathbf{0}$, $R \in \mathbb{R}^{m \times m}$, large $\gamma > 0$, initial stabilizing K^0 and K_ω^0, $M^0 = \mathbf{0}$, the tuning parameter $\alpha \in (0, 1]$, and a threshold $\zeta > 0$. Set $i = 0$.

2: Evaluate P^i by

$$\begin{aligned} 0 = &(A - BK^iC - DK_\omega^i)^{\mathrm{T}} P^i (A - BK^iC - DK_\omega^i) + (K^iC)^{\mathrm{T}} R K^i C \\ &- \gamma^2 (K_\omega^i)^{\mathrm{T}} K_\omega^i + C^{\mathrm{T}} Q^i C - P^i. \end{aligned} \tag{10}$$

3: Improve output-penalty weight Q^{i+1} by

$$\begin{aligned} Q^{i+1} = &\alpha (CC^{\mathrm{T}})^{-1} C [P^i - (A - BK^iC - DK_\omega^i)^{\mathrm{T}} P^i (A - BK^iC - DK_\omega^i) - (K^iC)^{\mathrm{T}} R \\ &\times K^iC + \gamma^2 (K_\omega^i)^{\mathrm{T}} K_\omega^i] C^{\mathrm{T}} (CC^{\mathrm{T}})^{-1} + (1 - \alpha) Q^i + \alpha (K^i - K_e)^{\mathrm{T}} R (K^i - K_e). \end{aligned} \tag{11}$$

4: Improve $(K^{i+1}, K_\omega^{i+1}, M^{i+1})$ by

$$\begin{aligned} K^{i+1} = &(R + B^{\mathrm{T}} P^i B + B^{\mathrm{T}} P^i D(\gamma^2 I - D^{\mathrm{T}} P^i D)^{-1} D^{\mathrm{T}} P^i B)^{-1} \\ &\times (-B^{\mathrm{T}} P^i A - B^{\mathrm{T}} P^i D(\gamma^2 I - D^{\mathrm{T}} P^i D)^{-1} D^{\mathrm{T}} P^i A + M^i) C^{\mathrm{T}} (CC^{\mathrm{T}})^{-1}, \end{aligned} \tag{12a}$$

$$\begin{aligned} M^{i+1} = &(R + B^{\mathrm{T}} P^i B + B^{\mathrm{T}} P^i D(\gamma^2 I - D^{\mathrm{T}} P^i D)^{-1} D^{\mathrm{T}} P^i B) K^{i+1} C \\ &+ B^{\mathrm{T}} P^i D(\gamma^2 I - D^{\mathrm{T}} P^i D)^{-1} D^{\mathrm{T}} P^i A + B^{\mathrm{T}} P^i A, \end{aligned} \tag{12b}$$

$$\begin{aligned} K_\omega^{i+1} = &(D^{\mathrm{T}} P^i D - \gamma^2 I - D^{\mathrm{T}} P^i B (R + B^{\mathrm{T}} P^i B)^{-1} B^{\mathrm{T}} P^i D)^{-1} \\ &\times (-D^{\mathrm{T}} P^i A + D^{\mathrm{T}} P^i B (R + B^{\mathrm{T}} P^i B)^{-1} B^{\mathrm{T}} P^i A). \end{aligned} \tag{12c}$$

5: Stop if $\|K^{i+1} - K_e\| \leq \zeta$. Otherwise, set $i \to i + 1$ and go to Step 2.

Remark 8 *The tuning parameter $\alpha \in (0, 1]$ is designed for the convergence of Algorithm 1 as shown in the following analysis. Algorithm 1 fuses RL (i.e., Steps 1 and 3) with inverse optimal control (i.e., Step 2) in a single iteration loop.*

Remark 9 *It is worth noting that with R and γ arbitrarily selected, only Q needs to be recovered from the algorithm. The reasons are the following. First, the goal is to find a cost function in terms of (2) that best captures the expert's trajectories. Second, it is well known that the reconstructed cost function capturing the same control policy would be non-unique in inverse optimal control Jean and Maslovskaya (2018). Then, given the fact, only recovering Q addresses the goal (as convergence is guaranteed below). Besides, we are interested in designing model-free algorithms using adaptive dynamic programming that sequentially recovers all parameters. Recovering all parameters may result in non-convergent solutions. Furthermore, only recovering Q reduces the complexity of calculation, guarantees convergence, and achieves a model-free iteration in all steps.*

Theorem 10 ***(Convergence of Algorithm 1).*** *Select Q^0 such that $(A, \sqrt{Q^0}C)$ is observable. Then, $P^i > 0$ in (10) holds for any i, and Q^i in (11) will converge to an approximate solution (see Definition 6) of Q which is an equivalent weight to Q_e, within finite iterations.*

Proof Referring to the convergence proof in Lian et al. (2023b), we will prove that Algorithm 1 learns a solution $P^i > 0$ at each iteration and Q^i converges within a finite number of iterations. Based on (10), (11) is rewritten as

$$\begin{aligned} Q^{i+1} &= \alpha(CC^{\mathrm{T}})^{-1}C[C^{\mathrm{T}}Q^iC]C^{\mathrm{T}}(CC^{\mathrm{T}})^{-1} + (1-\alpha)Q^i + \alpha(K^i - K_e)^{\mathrm{T}}R(K^i - K_e) \\ &= Q^i + \alpha(K^i - K_e)^{\mathrm{T}}R(K^i - K_e). \end{aligned} \tag{13}$$

Since $R = R^{\mathrm{T}} > 0$, we have

$$Q^{i+1} \geq Q^i \geq \cdots \geq Q^1 \geq Q^0 > \mathbf{0}. \tag{14}$$

This shows that $\{Q^i\}_0^\infty$ is monotonically increasing. Denote $\Delta^i = \alpha(K^i - K_e)^{\mathrm{T}}R(K^i - K_e)$. Then, (13) becomes

$$Q^{i+1} = \Delta^i + \Delta^{i-1} + \cdots + \Delta^0 + Q^0. \tag{15}$$

Suppose $\gamma > \underline{\gamma}$ and refer to Kleinman Theorem Kleinman (1968), $A - BK^iC - DK^i_\omega$ will be stabilizable. To prove $P^i > 0$ for all i, we now prove that $(A - BK^iC - DK^i_\omega, \sqrt{Q^i}C)$ is observable by contradiction. As the pair of initial (K^0, K^0_ω) is selected as stabilizing and $(A - BK^0C - DK^0_\omega, \sqrt{Q^0}C)$ is observable. Then, $\sqrt{Q^0}Cv = 0$ holds only if $v = \mathbf{0}$. At $i = 1$, we suppose that $(A - BK^1C - DK^1_\omega, \sqrt{Q^1}C)$ is not observable. That is, there is nonzero v such that $\sqrt{Q^1}Cv = \mathbf{0}$. As $Q^1 = Q^0 + \alpha\Delta^0$ and $\Delta^0 \geq 0$, one has $v^TC^TQ^1Cv = v^TC^TQ^0Cv + \alpha v^TC^T\Delta^0Cv$, and $\sqrt{Q^1}Cv = \mathbf{0}$ implies $\sqrt{Q^0}Cv = \mathbf{0}$ with nonzero v, which conflicts with the initial suppose. Thus, $(A - BK^1C - DK^1_\omega, \sqrt{Q^1}C)$ is observable. By deduction, we can conclude that $(A - BK^iC - DK^i_\omega, \sqrt{Q^i}C)$ is observable for all i and thus (10) generates $P^i \geq 0$ for all i.

Due to the ill-posedness of inverse problems Ng et al. (2000); Xue et al. (2021), there can be many equivalent weights to Q_e. Denote one by $Q > 0$, by selecting an initial $Q^0 > 0$ and a $\alpha \in (0, 1]$ to adjust the increment of Q^i, a small threshold η exists such that $\|Q^i - Q\| \leq \eta$

holds. Also, a cost weight matrix uniquely defines a feedback gain in the Riccati equation Lancaster (1995). As Q^i increases to Q, K^i would approach K_e in terms of $\|K^i - K_e\| \leq \zeta$, where $\zeta > 0$ is a small threshold to stop the computation. Then $\|Q\| + \eta \geq \|Q^i\| \geq \|Q^i - Q^0\| = \|\Delta^{i-1} + \cdots + \Delta^0\|$ which implies

$$i \leq \text{ceil}\left\{\frac{\|Q\| + \eta + \|Q^0\|}{\alpha e^2 \lambda_{min}(R)}\right\} \equiv \bar{i}. \tag{16}$$

According to Definition 5, $\|Q\|$ is bounded. Algorithm 1 terminates, at most, in $\bar{i}$ iterations. ■

3.2. Model-Free Input-Output Inverse RL

We now propose a model-free, data-driven IRL algorithm that is equivalent to Algorithm 1 using (y_k, u_k, w_k) of learner and (y_{ek}, u_{ek}) of expert to solve Problem 1. Specifically, we present three phases that replace (10), (11), (12a)-(12c), respectively.

Phase 1: Design a model-free, data-driven equation to replace (10). Using the x_k into (10) yields

$$\begin{aligned}
x_k^{\mathrm{T}} P^i x_k &= x_k^{\mathrm{T}}(A - BK^iC - DK_\omega^i)^{\mathrm{T}} P^i (A - BK^iC - DK_\omega^i)x_k + x_k^{\mathrm{T}}(K^iC)^{\mathrm{T}} R K^i C x_k \\
&\quad - \gamma^2 x_k^{\mathrm{T}}(K_\omega^i)^{\mathrm{T}} K_\omega^i x_k + x_k^{\mathrm{T}} C^{\mathrm{T}} Q^i C x_k \\
&= (Ax_k + Bu_k + D\omega_k)^{\mathrm{T}} P^i (Ax_k + Bu_k + D\omega_k) + y_k^{\mathrm{T}}(K^i)^{\mathrm{T}} R K^i y_k \\
&\quad - \gamma^2 x_k^{\mathrm{T}}(K_\omega^i)^{\mathrm{T}} K_\omega^i x_k + y_k^{\mathrm{T}} Q^i y_k \\
&\equiv X_k^{\mathrm{T}} H^i X_k,
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
&X_k \equiv \begin{bmatrix} x_k \\ u_k \\ \omega_k \end{bmatrix}, H^i \equiv \begin{bmatrix} H_{xx}^i & H_{xu}^i & H_{x\omega}^i \\ H_{ux}^i & H_{uu}^i & H_{u\omega}^i \\ H_{\omega x}^i & H_{\omega u}^i & H_{\omega\omega}^i \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}, \\
&H_{xx}^i \equiv C^{\mathrm{T}} Q^i C + A^{\mathrm{T}} P^i A \in \mathbb{R}^{n\times n}, H_{xu}^i = (H_{ux}^i)^{\mathrm{T}} \equiv A^{\mathrm{T}} P^i B \in \mathbb{R}^{n\times m}, \\
&H_{x\omega}^i = (H_{\omega x}^i)^{\mathrm{T}} \equiv A^{\mathrm{T}} P^i D \in \mathbb{R}^{n\times l}, H_{uu}^i \equiv R_e + B^{\mathrm{T}} P^i B \in \mathbb{R}^{m\times m} \\
&H_{u\omega}^i = (H_{\omega u}^i)^{\mathrm{T}} \equiv B^{\mathrm{T}} P^i D \in \mathbb{R}^{l\times m}, H_{\omega\omega}^i \equiv D^{\mathrm{T}} P^i D - \gamma^2 I \in \mathbb{R}^{l\times l}.
\end{aligned}$$

However, note that x_k is unavailable (see Assumption 4), but (u_k, y_k, ω_k) is available. The next lemma demonstrates how to use (u_k, y_k, ω_k) to reconstruct x_k of the learner.

Lemma 11 *Lewis and Vamvoudakis (2010); Rizvi and Lin (2018) With Assumption 1, the learner system state at time k is uniquely reconstructed by a sequence of historical control inputs, disturbance inputs, and outputs from $k-N$ to $k-1$ as*

$$x_k = \begin{bmatrix} G_\omega & G_u & G_y \end{bmatrix} \begin{bmatrix} (\Omega_{k-1}^{k-N})^T & (U_{k-1}^{k-N})^T & (Y_{k-1}^{k-N})^T \end{bmatrix}, \tag{18}$$

where $N \geq n$ is an upper bound, and $\Omega_{k-1}^{k-N} \in \mathbb{R}^{lN}$, $U_{k-1}^{k-N} \in \mathbb{R}^{mN}$ and $Y_{k-1}^{k-N} \in \mathbb{R}^{pN}$ are the control input, disturbance input, and output data vectors

$$\Omega_{k-1}^{k-N} = \begin{bmatrix} \omega_{k-1} \\ \vdots \\ \omega_{k-N} \end{bmatrix}, U_{k-1}^{k-N} = \begin{bmatrix} u_{k-1} \\ \vdots \\ u_{k-N} \end{bmatrix}, Y_{k-1}^{k-N} = \begin{bmatrix} y_{k-1} \\ \vdots \\ y_{k-N} \end{bmatrix}. \tag{19}$$

The coupling matrices are

$$G_\omega = \Omega_N - A^N(L_N^T L_N)^{-1} L_N^T V_{N1}, G_u = T_N - A^N(L_N^T L_N)^{-1} L_N^T V_{N2}, G_y = A^N(L_N^T L_N)^{-1} L_N^T,$$

with the observability matrix L_N, the controllability matrix pair (Ω_N, T_N), and the Toeplitz matrix pair (V_{N1}, V_{N2}) given by

$$\Omega_N = \begin{bmatrix} D & \dots & A^{N-1}D \end{bmatrix}, T_N = \begin{bmatrix} B & \dots & A^{N-1}B \end{bmatrix}, L_N = \begin{bmatrix} (CA^{N-1})^T & \dots & (CA)^T & C^T \end{bmatrix},$$

$$V_{N1} = \begin{bmatrix} \boldsymbol{0} & CD & CAD & \dots & CA^{N-2}D \\ \boldsymbol{0} & \boldsymbol{0} & CD & \dots & CA^{N-3}D \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \dots & \boldsymbol{0} & CD \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, V_{N2} = \begin{bmatrix} \boldsymbol{0} & CB & CAB & \dots & CA^{N-2}B \\ \boldsymbol{0} & \boldsymbol{0} & CB & \dots & CA^{N-3}B \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \dots & \boldsymbol{0} & CB \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Since (A, C) is observable, there exists a constant $N_0 > 0$ that L_N has full column rank when $N \geq N_0$ (see Lemma 1 in Lewis and Vamvoudakis (2010) and Lemma 1 in Rizvi and Lin (2018)).

We now leverage Lemma 11 to replace states in (17) with historical inputs and outputs. Define $z_k = \begin{bmatrix} (\Omega_{k-1}^{k-N})^{\mathrm{T}} & (U_{k-1}^{k-N})^{\mathrm{T}} & (Y_{k-1}^{k-N})^{\mathrm{T}} & \omega_k^{\mathrm{T}} & u_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Substituting (18) into (17) yields an input-output Q-function, denoted by

$$\mathbf{Q}(z_k, \mathbf{H}^i) = z_k^{\mathrm{T}} \mathbf{H}^i z_k, \tag{20}$$

where

$$\mathbf{H}^i \equiv \begin{bmatrix} H_{\Omega\Omega}^i & H_{\Omega U}^i & H_{\Omega Y}^i & H_{\Omega\omega}^i & H_{\omega u}^i \\ * & H_{UU}^i & H_{UY}^i & H_{U\omega}^i & H_{Uu}^i \\ * & * & H_{YY}^i & H_{Y\omega}^i & H_{Yu}^i \\ * & * & * & H_{\omega\omega}^i & H_{\omega u}^i \\ * & * & * & * & H_{uu}^i \end{bmatrix} \in \mathbb{R}^{((m+l+p)N+m+l)\times((m+l+p)N+m+l)},$$

$$H_{\Omega\Omega}^i \equiv G_\omega^{\mathrm{T}} H_{xx}^i G_\omega \in \mathbb{R}^{lN\times lN}, \quad H_{\Omega U}^i \equiv G_\omega^{\mathrm{T}} H_{xx}^i G_U \in \mathbb{R}^{lN\times mN}, \quad H_{\Omega Y}^i \equiv G_\omega^{\mathrm{T}} H_{xx}^i G_Y \in \mathbb{R}^{lN\times pN},$$
$$H_{\Omega\omega}^i \equiv G_\omega^{\mathrm{T}} H_{x\omega}^i \in \mathbb{R}^{lN\times p}, \quad H_{\Omega u}^i \equiv G_\omega^{\mathrm{T}} H_{xu}^i \in \mathbb{R}^{lN\times m}, \quad H_{UU}^i \equiv G_u^{\mathrm{T}} H_{xx}^i G_u \in \mathbb{R}^{mN\times mN},$$
$$H_{UY}^i \equiv G_u^{\mathrm{T}} H_{xx}^i G_y \in \mathbb{R}^{mN\times pN}, \quad H_{U\omega}^i \equiv G_u^{\mathrm{T}} H_{x\omega}^i \in \mathbb{R}^{mN\times l}, \quad H_{Uu}^i \equiv G_u^{\mathrm{T}} H_{xu}^i \in \mathbb{R}^{mN\times m},$$
$$H_{YY}^i \equiv G_y^{\mathrm{T}} H_{xx}^i G_y \in \mathbb{R}^{pN\times pN}, \quad H_{Y\omega}^i \equiv G_y^{\mathrm{T}} H_{x\omega}^i \in \mathbb{R}^{pN\times l}, \quad H_{Yu}^i \equiv G_y^{\mathrm{T}} H_{xu}^i \in \mathbb{R}^{pN\times m},$$

Combining (17) and (20) yields

$$z_k^{\mathrm{T}} \mathbf{H}^i z_k = z_{k+1}^{\mathrm{T}} \mathbf{H}^i z_{k+1} + (u_k^i)^{\mathrm{T}} R u_k^i - \gamma^2 (\omega_k^i)^{\mathrm{T}} \omega_k^i + y_k^{\mathrm{T}} Q^i y_k. \tag{21}$$

Phase 2: Design a model-free, data-driven equation that replaces (11) for solving Q^{i+1} by using expert's data (y_{ek}, u_{ek}). Multiplying the both sides of (13) by C^{T} and C yields

$$C^{\mathrm{T}} Q^{i+1} C = \alpha (K^i C - K_e C)^{\mathrm{T}} R (K^i C - K_e C) + C^{\mathrm{T}} Q^i C. \tag{22}$$

Multiplying x_{ek}^{T} and x_{ek} to (22) yields

$$\begin{aligned} y_{ek}^{\mathrm{T}} Q^{i+1} y_{ek} &= \alpha (K^i C x_{ek} + u_{ek})^{\mathrm{T}} R (K^i C x_{ek} + u_{ek}) + y_{ek}^{\mathrm{T}} Q^i y_{ek} \\ &= \alpha (K^i y_{ek} + u_{ek})^{\mathrm{T}} R (K^i y_{ek} + u_{ek}) + y_{ek}^{\mathrm{T}} Q^i y_{ek}. \end{aligned} \tag{23}$$

Phase 3: Design model-free, data-driven equations that replace (12a)-(12c). To replace (12a), we multiply the right side of (12a) by x_k and yield

$$\begin{aligned} u_k^{i+1} &= -K^{i+1}Cx_k = -K^{i+1}y_k \\ &= -(R+B^{\mathrm{T}}P^iB+B^{\mathrm{T}}P^iD(\gamma^2I-D^{\mathrm{T}}P^iD)^{-1}D^{\mathrm{T}}P^iB)^{-1} \\ &\quad \times(-B^{\mathrm{T}}P^iA-B^{\mathrm{T}}P^iD(\gamma^2I-D^{\mathrm{T}}P^iD)^{-1}D^{\mathrm{T}}P^iA+M^i)x_k \\ &= -(H_{uu}^i-H_{u\omega}^i(H_{\omega\omega}^i)^{-1}H_{\omega u}^i)\big[-H_{u\Omega}^i\Omega-H_{uU}^iU-H_{uY}^iY \\ &\quad +H_{u\omega}^i(H_{\omega\omega}^i)^{-1}(H_{\omega\Omega}^i\Omega+H_{\omega U}^iU+H_{\omega Y}^iY)+M^i(x_k)\big], \end{aligned} \tag{24}$$

where $M^i(x_k)\equiv M^ix_k$ and for ease of notation, denote Ω, U, Y for Ω_{k-1}^{k-N}, U_{k-1}^{k-N}, Y_{k-1}^{k-N}. To replace (12b), we multiply x_k on the right side of (12b) and yield

$$\begin{aligned} M^{i+1}(x_k) &= (R+B^{\mathrm{T}}P^iB+B^{\mathrm{T}}P^iD(\gamma^2I-D^{\mathrm{T}}P^iD)^{-1}D^{\mathrm{T}}P^iB)K^{i+1}Cx_k \\ &\quad +B^{\mathrm{T}}P^iD(\gamma^2I-D^{\mathrm{T}}P^iD)^{-1}D^{\mathrm{T}}P^iAx_k+B^{\mathrm{T}}P^iAx_k \\ &= H_{uu}^i-H_{u\omega}^i(H_{\omega\omega}^i)^{-1}H_{\omega U}^iK^{i+1}y_k-H_{u\omega}^i(H_{\omega\omega}^i)^{-1}(H_{\omega U}^iU+H_{\omega\Omega}^i\Omega+H_{\omega Y}^iY) \\ &\quad +H_{uU}^iU+H_{u\Omega}^i\Omega+H_{uY}^iY. \end{aligned} \tag{25}$$

To replace (12c), we multiply the right side of (12c) by x_k and yield

$$\begin{aligned} \omega_k^{i+1} &= (D^{\mathrm{T}}P^iD-\gamma^2I-D^{\mathrm{T}}P^iB(R+B^{\mathrm{T}}P^iB)^{-1}B^{\mathrm{T}}P^iD)^{-1} \\ &\quad \times(-D^{\mathrm{T}}P^iA+D^{\mathrm{T}}P^iB(R+B^{\mathrm{T}}P^iB)^{-1}B^{\mathrm{T}}P^iA)x_k \\ &= (H_{\omega\omega}^i-H_{\omega u}^i(H_{uu}^i)^{-1}H_{u\omega}^i)^{-1}\big[-H_{\omega\Omega}^i\Omega-H_{\omega U}^iU-H_{\omega Y}^iY \\ &\quad +H_{\omega u}^i(H_{uu}^i)^{-1}(H_{u\Omega}^i\Omega+H_{uU}^iU+H_{uY}^iY)\big]. \end{aligned} \tag{26}$$

The above derivations are summarized as the model-free, data-driven IRL Algorithm 2.

Algorithm 2 Model-Free Robust IRL of DT Input-output Zero-Sum Games

1: Select initial $Q^0>\mathbf{0}$, $Q^0\in\mathbb{R}^{p\times p}$, $R>\mathbf{0}$, $R\in\mathbb{R}^{m\times m}$, large $\gamma>0$, tiny threshold $\epsilon>0$, initial stabilizing u^0 and ω^0, and $\alpha\in(0,1]$. Set $i=0$ and $M^0(x_k)=0$.

2: Solve $\mathbf{H}^i$ via (21).

3: Improve Q^{i+1} via (23).

4: Improve u_k^{i+1}, $M^{i+1}(x_k)$, and ω_k^{i+1} via (24), (25), (26), respectively.

5: Stop if $\|Q^{i+1}-Q^i\|\le\epsilon$, or set $i\to i+1$ and go to Step 2.

Remark 12 *Algorithm 2 will be implemented online through the Batch Least Squares. The procedure can be obtained by extending that in Lian et al. (2023a) to this context. It is omitted here due to the page limitation. To ensure the learner system is persistently excited (PE), small probing noises will be added to control input u_k^i in (21) and the unbiased solutions are obtained by extending that in Rizvi and Lin (2018). To solve $\boldsymbol{H}^i$, at least $((m+l+p)N+m+l)((m+l+p)N+m+l+1)/2$ data tuples are needed. To solve Q^{i+1}, at least $p(p+1)/2$ data tuples are needed during $k\in[\underline{k},\bar{k}]$.*

Remark 13 *The advantages of using the learner's data in Phases 1 and 3 and the expert's data in Phase 2 are that 1) the learner's disturbance is more accessible than the expert's; 2) the update of Q^i in Phase 2 does not require the expert system to be PE, reducing requirement on the expert; 3) Phases 1 and 2 can obtain unbiased solutions according to Rizvi and Lin (2018).*

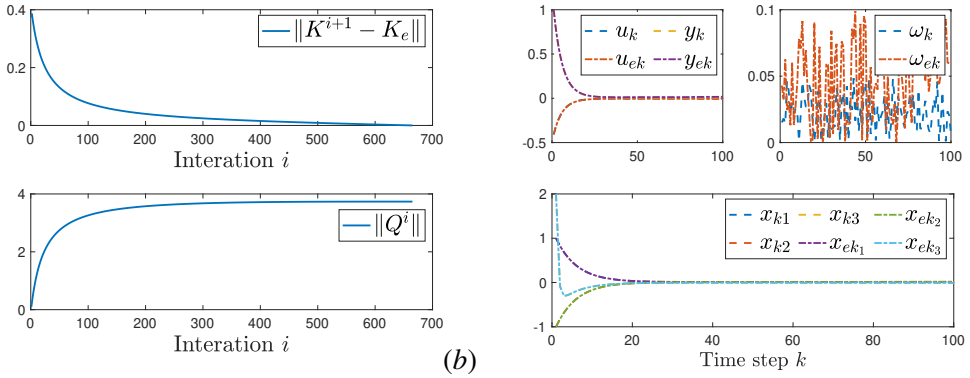
4. Simulation

We consider the learner and expert as the short-period aircraft dynamics with a three-dimensional state of $x = [\tau\ q\ \beta]^{\mathrm{T}}$ from Stevens et al. (2015). The τ, q, β denote the attack angle, pitch rate, and elevator deflection angle, respectively. The dynamics matrices from Rizvi and Lin (2018) are

$$A = \begin{bmatrix} 0.90649 & 0.08160 & -0.00050 \\ 0.07413 & 0.90121 & -0.00071 \\ 0 & 0 & 0.13266 \end{bmatrix}, B = \begin{bmatrix} -0.00151 \\ -0.00960 \\ 0.86730 \end{bmatrix}, D = \begin{bmatrix} 0.00952 \\ 0.00038 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^{\mathrm{T}}.$$

Assume the expert cost weights for the above system, in the form of (2), to be $Q_e = 8$, $R_e = 2$, and $\gamma_e = 8$. We have the static OPFB control policy matrices as $K_e = 0.4091$, $K_\omega^e = \begin{bmatrix} 0.0144 & 0.0099 & -0.0000 \end{bmatrix}$, $M_e = \begin{bmatrix} 0.0058 & -0.6737 & 0.0010 \end{bmatrix}$.

For Algorithm 2, we select $Q^0 = 0.1$, $R = 1$, $\gamma = 6$, $N = 3$, $\alpha = 1$, and $e = 0.001$. Assume the probing noises as $w_k = 0.001a_k$, the learner's disturbance as $\omega_k = 0.005b_k$, and the expert's disturbance as $\omega_k = 0.01c_k$ with a_k, b_k, c_k being randoms in [0, 1]. The learner and expert start with the same initial states $[1\ \ -1\ 2]^{\mathrm{T}}$. Figure 4 (a) shows the convergence of K^i to K_e and Q^i to an equivalent weight to Q_e. The converged values are $K^\infty = 0.4091$, $Q^\infty = 3.7358$. Note that the learned Q^∞ can differ from Q_e. Figure 4 (b) shows that using the learned K^∞, the learner trajectories of (x_k, u_k, y_k) are close to the demonstrated trajectories (x_{ek}, u_{ek}, y_{ek}) while allowing differences between relative small ω_k and ω_{ek}.



(*a*) (*b*)

Figure 1: (a) Convergence of K^{i+1} and Q^i using Algorithm 2; (b) Trajectories of states, inputs, outputs of expert and learner that is generated by using K^∞.

5. Conclusion

This paper studies IRL algorithms to reconstruct the objective function and OPFB control policy for discrete-time systems with disturbances modeled by DT zero-sum games, along with theoretical guarantees. The model-free algorithm uses the expert's control input and output data.

Acknowledgments

We thank the support of Auburn University research fund.

References

Navid Aghasadeghi and Timothy Bretl. Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, pages 1561–1566. IEEE, 2011.

Asma Al-Tamimi, Frank L Lewis, and Murad Abu-Khalaf. Model-free q-learning designs for linear discrete-time zero-sum games with application to h-infinity control. *Automatica*, 43(3):473–481, 2007.

T. Başar and G.J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1998.

Richard C Dorf Robert H Bishop. *Modern control systems*. 2011.

Kun Cao and Lihua Xie. Game-theoretic inverse reinforcement learning: A differential pontryagin's maximum principle approach. *IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2022.3148376*, 2022.

Vrushabh S Donge, Bosen Lian, Frank L Lewis, and Ali Davoudi. Multiagent graphical games with inverse reinforcement learning. *IEEE Transactions on Control of Network Systems*, 10(2): 841–852, 2022.

Fatemeh Mahdavi Golmisheh and Saeed Shamaghdari. Optimal robust formation of multi-agent systems as adversarial graphical apprentice games with inverse reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 2024.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Frédéric Jean and Sofya Maslovskaya. Inverse optimal control problem: the linear-quadratic case. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 888–893, 2018. doi: 10.1109/CDC.2018.8619204.

Wanxin Jin, Dana Kulić, Shaoshuai Mou, and Sandra Hirche. Inverse optimal control from incomplete trajectory observations. *The International Journal of Robotics Research*, 40(6-7):848–865, 2021.

David Kleinman. On an iterative technique for riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1):114–115, 1968.

P Lancaster. *Algebraic Riccati Equations*. Oxford Science Publications/The Clarendon Press, Oxford University Press, 1995.

Frank L Lewis and Kyriakos G Vamvoudakis. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):14–25, 2010.

Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

Jie Li, Shengbo Eben Li, Jingliang Duan, Yao Lyu, Wenjun Zou, Yang Guan, and Yuming Yin. Relaxed policy iteration algorithm for nonlinear zero-sum games with application to h-infinity control. *IEEE Transactions on Automatic Control*, 69(1):426–433, 2023.

Bosen Lian, Wenqian Xue, Frank L Lewis, and Tianyou Chai. Inverse reinforcement learning for multi-player noncooperative apprentice games. *Automatica*, 145:110524, 2022.

Bosen Lian, Wenqian Xue, Frank L Lewis, and Ali Davoudi. Inverse q-learning using input–output data. *IEEE Transactions on Cybernetics*, 2023a.

Bosen Lian, Wenqian Xue, Yijing Xie, Frank L Lewis, and Ali Davoudi. Off-policy inverse q-learning for discrete-time antagonistic unknown systems. *Automatica*, 155:111171, 2023b.

Bosen Lian, Wenqian Xue, Frank L Lewis, Hamidreza Modares, and Bahare Kiumarsi. *Integral and Inverse Reinforcement Learning for Optimal Control Systems and Games*. Springer, 2024.

Emin Martirosyan and Ming Cao. Reinforcement learning for inverse linear-quadratic dynamic non-cooperative games. *Systems & Control Letters*, 191:105883, 2024.

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 *International Conference on Machine Learning*, volume 1, page 2, 2000.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

Syed Ali Asad Rizvi and Zongli Lin. Output feedback q-learning for discrete-time linear zero-sum games with application to the h-infinity control. *Automatica*, 95:213–221, 2018.

Brian L Stevens, Frank L Lewis, and Eric N Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

Samuel Tesfazgi, Armin Lederer, and Sandra Hirche. Inverse reinforcement learning: A control lyapunov approach. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3627–3632. IEEE, 2021.

Jared Town, Zachary Morrison, and Rushikesh Kamalapurkar. Nonuniqueness and convergence to equivalent solutions in observer-based inverse reinforcement learning. *Automatica*, 171:111977, 2025.

Amir Parviz Valadbeigi, Ali Khaki Sedigh, and Frank L Lewis. H∞ static output-feedback control design for discrete-time systems using reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2):396–406, 2019.

Huai-Ning Wu. Online learning human behavior for a class of human-in-the-loop systems via adaptive inverse optimal control. *IEEE Transactions on Human-Machine Systems*, 52(5):1004–1014, 2022.

Markus Wulfmeier, Dushyant Rao, Dominic Zeng Wang, Peter Ondruska, and Ingmar Posner. Large-scale cost function learning for path planning using deep inverse reinforcement learning. *The International Journal of Robotics Research*, 36(10):1073–1087, 2017.

Wenqian Xue, Bosen Lian, Jialu Fan, Patrik Kolaric, Tianyou Chai, and Frank L Lewis. Inverse reinforcement q-learning through expert imitation for discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, 34(5):2386–2399, 2021.

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, 2022.

Wenqian Xue, Bosen Lian, Jialu Fan, Tianyou Chai, and Frank L. Lewis. Inverse reinforcement learning for trajectory imitation using static output feedback control. *IEEE Transactions on Cybernetics*, 54(3):1695–1707, 2024. doi: 10.1109/TCYB.2023.3241015.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.