Model-based inverse reinforcement learning for deterministic systems

Ryan Self [a] , Moad Abudia [a] , S M Nahid Mahmud [a] , Rushikesh Kamalapurkar [a]

[a] *School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma, 74078, USA*

Abstract

This paper focuses on the development of an online data-driven model-based inverse reinforcement learning (MBIRL) technique for linear and nonlinear deterministic systems. Input and output trajectories of an agent under observation, attempting to optimize an unknown reward function, are used to estimate the reward function and the corresponding unknown optimal value function, online and in real-time. To achieve MBIRL using limited data, a novel feedback-driven approach to MBIRL is developed. The feedback policy and the dynamic model of the agent under observation are estimated from the measured data and the estimates are used to generate synthetic data to drive MBIRL. Theoretical guarantees for ultimate boundedness of the estimation errors in general, and convergence of the estimation errors to zero in special cases, are derived using Lyapunov techniques. Proof of concept numerical experiments demonstrate the utility of the developed method to solve linear and nonlinear inverse reinforcement learning problems.

Key words: inverse reinforcement learning, inverse optimal control, system identification, state estimation

1 Introduction

Based on the premise that the most succinct representation of the behavior of an entity is its reward structure [1], this paper aims to recover the reward (or cost) function of an agent by observing the agent performing a task and monitoring its inputs and outputs. Methods to estimate the reward function using inputs and outputs fall under the umbrella of inverse reinforcement learning (IRL) (see, for example, [1] and [2]), and in a model-based context, are also referred to as inverse optimal control (IOC) [3]. The IRL method developed in this paper learns the reward function and the value function of an agent under observation, modeled as a *deterministic* nonlinear dynamical system, *online*, in the presence of modeling uncertainties, using input-output data.

While IRL in an *offline* setting has a rich history of literature [1,2,4–14], traditional IRL methods typically require a large amount of training data and iterative, computationally intensive training algorithms. As such, offline methods are ill-suited for applications such as real-time intent monitoring and real-time adaptation from expert demonstrations, which present challenges such as changing task objectives and uncertainties/changes in the dynamics of the demonstrator. To address such applications, this paper focuses on online, real-time IRL, for a limited class of systems.

IRL techniques that facilitate reward function estimation in real-time have recently started gaining attention [15–28]. In [19–22], the authors formulate the online IRL problem as a lifelong learning problem. In [15] and [17], a batch method is developed for learning cost functions of demonstrators behaving according to policies that solve linear and nonlinear infinite horizon optimal control problems, while a recursive technique for linear systems is proposed in [16]. In [29, 30], the authors develop sequential and batch processing methods for solving the IOC problem for deterministic discrete-time nonlinear systems with both infinite and finite horizons. In [18], the authors study IRL in the presence of

⋆ This research was supported, in part, by the National Science Foundation (NSF) under award number 1925147 and the Air Force Research Laboratories under award number FA8651-19-2-0009. Any opinions, findings, conclusions, or recommendations detailed in this article are those of the author(s), and do not necessarily reflect the views of the sponsoring agencies.

Email addresses: rself@okstate.edu (Ryan Self), abudia@okstate.edu (Moad Abudia), nahid.mahmud@okstate.edu (S M Nahid Mahmud), rushikesh.kamalapurkar@okstate.edu (Rushikesh Kamalapurkar).

unknown disturbances affecting both the learner and the demonstrator, and in [27], the authors utilize IRL to learn the unknown reward function for tracking control. However, results such as [15, 16, 27] only focus on linear systems, and results such as [17, 18, 29, 30] require either full state measurements or exact knowledge of the system dynamics. In addition, a majority of the IRL/IOC methods require trajectories that are sufficiently exciting in order to estimate the reward function online. The techniques developed in this paper aim to alleviate the aforementioned limitations of existing IRL/IOC methods.

In this paper, a model-based IRL (MBIRL) approach is developed for deterministic systems in continuous time based on the preliminary results in [15], [17], and [31]. The key contribution of this paper is the development of a novel method for reward function estimation for linear and nonlinear systems, using a model-based recursive IRL technique, in an online setting, using input-output measurements (as opposed to input-state measurements used in results such as [17, 18], and [31]), while compensating for uncertainties in the agent dynamics. Using Lyapunov theory, the developed MBIRL technique is shown to result in ultimate boundedness of the reward function estimation error.

Another contribution of this paper, is a novel feedback-driven approach to MBIRL for the case where the measured data does not provide sufficient information for direct reward function estimation. Since a majority of existing IRL methods are trajectory-driven and model-free, the measured trajectories need to be sufficiently information-rich for reward function estimation. The technique developed in this paper is model-based, and as a result, once a model is learned, it can utilize arbitrary state-action pairs for IRL as long as the action is the optimal action corresponding to that state. The key idea in the feedback-driven method is to estimate the optimal feedback policy of the agent online using the measured output-action pairs, and to use that estimate to artificially create additional state-action pairs to drive reward function estimation.

The paper is organized as follows: Section 2 details the notation, Section 3 introduces the problem, Section 4 presents an overview of the MBIRL approach, Section 5 solves the IRL problem using sufficiently exciting observed trajectories, Section 6 develops a feedback-driven MBIRL approach for the case where the observed trajectories are insufficient for direct reward function estimation, Section 7 demonstrates the effectiveness of the developed method through simulations, and Section 8 concludes the paper.

2 Notation

The notation $\mathbb{R}^n$ represents the $n-$dimensional Euclidean space, and the elements of $\mathbb{R}^n$ are interpreted as column vectors, where $(\cdot)^T$ denotes the vector transpose operator. The set of positive integers excluding 0 is denoted by $\mathbb{N}$. For $a \in \mathbb{R}$, $\mathbb{R}_{\geq a}$ denotes the interval $[a, \infty)$, and $\mathbb{R}_{>a}$ denotes the interval (a, ∞). The notations I_n and 0_n (or I and 0 if the dimension is clear from the context) denote the $n \times n$ identity matrix and the zero element of $\mathbb{R}^n$, respectively. The notation $a_{(i)}$ is used to denote the $i-$th component of a vector a.

3 Problem Formulation

Consider an agent under observation with the dynamics

$$\begin{aligned} \dot{x} &= f^o(x, u) + g(x, u), \\ y &= h(x, u), \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control, $f^o : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents a nominal model, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the uncertainty, $y \in \mathbb{R}^l$ is the output, and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ denotes the measurement model.

The agent under observation uses a controller $u(\cdot)$ that minimizes the performance index

$$J(x_0, u(\cdot)) = \int_0^\infty Q(x(t; x_0, u_{[0,t)})) + \big(u(t)\big)^T R u(t)\, \mathrm{d}t, \tag{2}$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite (P.D.) matrix, $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive semi-definite (P.S.D.) continuously differentiable function with a locally Lipschitz continuous gradient, and $x(\cdot; x_0, u_{[0,t)})$ is the trajectory of the agent, in response to the control signal $u(\cdot)$, restricted to the time interval $[0, t)$, starting from the initial condition x_0. Since R can be selected to be symmetric without loss of generality, only the elements of R that are on and above the main diagonal are estimated. The following assumptions are required to facilitate the development and the analysis of the MBIRL method.

Assumption 1 *The state and control trajectories are bounded, such that $x(t) \in \mathcal{X}$ and $u(t) \in \mathcal{U}$, for all $t \in \mathbb{R}_{\geq 0}$ and for some compact sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$.*

Assumption 2 *The partial derivatives of f^o and g in (1) with respect to x and u are locally Lipschitz continuous.*

Assumption 3 *The optimal control problem defined by (1) and (2) admits a twice continuously differentiable optimal value function.*

The class of affine systems is large, as it includes linear systems and Euler Lagrange systems with invertible inertia matrices. While Lipschitz continuity of the partial

derivatives of the dynamics and twice continuous differentiability of the value function are strict requirements, many optimal control problems of interest, such as linear quadratic problems and nonlinear problems similar to those used for demonstration in Section 7.1, meet these requirements.

The main objective of this paper is to estimate Q and R using measurements of the input and the output, under the assumption that $u(t)$ is the optimal action in response to the state $x(t, x_0, u_{[0,t)})$. While the estimated reward function can be used in a forward RL method to generate optimal policies that imitate the demonstrator, to focus the discussion on reward function estimation, this paper does not consider policy synthesis. In the following, the input and the output signals available for measurement will be denoted by $t \mapsto u(t)$ and $t \mapsto y(t)$, respectively, the corresponding unknown true state will be denoted by $t \mapsto x(t)$, and x and u will be used to denote generic elements of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

4 Overview of the Developed Approach

MBIRL utilizes an indirect error metric called the inverse Bellman error (IBE) to learn the unknown reward function. The IBE is a model-based error metric that is a function of both the unmeasureable system states and the agent's uncertain dynamics. In addition, the IBE is also dependent on the unknown optimal value function. In this work, estimates of the states, dynamics, and value function are utilized to estimate the IBE, and the estimated IBE is utilized to realize MBIRL.

MBIRL integrates a state observer that produces state estimates, a parameter estimator that produces estimates of unknown parameters in the system dynamics, and a new algorithm that produces estimates of the reward function and the value function. For the case where information extracted from the agent's trajectory is insufficient for direct reward function estimation, a feedback-driven MBIRL method is developed, where a policy estimator is utilized to generate artificial training data.

See Fig. 1 for a block diagram that summarizes the MBIRL method and Fig. 2 for block diagram that summarizes the feedback-driven MBIRL method.

4.1 The Inverse Bellman Error

Under the premise that the observed agent makes optimal decisions, the state and control trajectories, $x(\cdot)$ and $u(\cdot)$, satisfy the Hamilton-Jacobi-Bellman equation [32]

$$H\left(x\left(t\right), \nabla_x \left(V^*\left(x\left(t\right)\right)\right)^T, u\left(t\right)\right) = 0, \forall t \in \mathbb{R}_{\geq 0}, \quad (3)$$

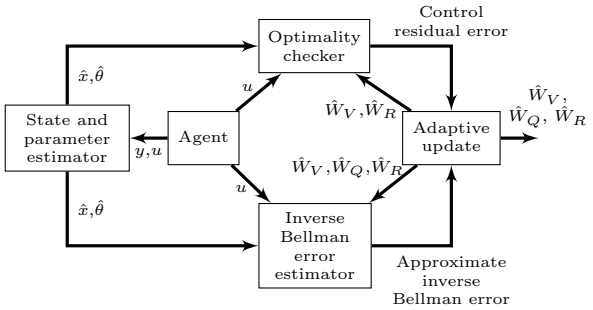

Fig. 1. Block diagram of the developed MBIRL method.

where $V^* : \mathbb{R}^n \to \mathbb{R}$ is the unknown optimal value function and $H : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the Hamiltonian, defined as $H(x, p, u) \coloneqq p^T \left(f^o(x, u) + g(x, u)\right) + Q(x) + u^T R u$.

The functions V^* and Q can be approximated using a linear combination of $P \in \mathbb{N}$ and $L \in \mathbb{N}$ basis functions, respectively, as $V^*(x) = (W_V^*)^T \sigma_V(x) + \epsilon_V(x)$ and $Q(x) = (W_Q^*)^T \sigma_Q(x) + \epsilon_Q(x)$. The vectors $W_V^* \in \mathbb{R}^P$ and $W_Q^* \in \mathbb{R}^L$ denote the ideal weights, $\sigma_V : \mathbb{R}^n \to \mathbb{R}^P$ and $\sigma_Q : \mathbb{R}^n \to \mathbb{R}^L$ denote continuously differentiable known features with locally Lipschitz continuous gradients, and $\epsilon_V : \mathbb{R}^n \to \mathbb{R}$ and $\epsilon_Q : \mathbb{R}^n \to \mathbb{R}$ denote the approximation errors. The basis functions are selected such that approximation of the functions and their derivatives is uniform over the compact set $\mathcal{X}$, so that given any constants $\overline{\epsilon}_V, \overline{\epsilon}_Q \in \mathbb{R}_{>0}$, there exist $P, L \in \mathbb{N}$ such that ϵ_V and ϵ_Q satisfy $\sup_{x \in \mathcal{X}} \left\|\epsilon_V\left(x\right)\right\| < \overline{\epsilon}_V$, $\sup_{x \in \mathcal{X}} \left\|\nabla \epsilon_V\left(x\right)\right\| < \overline{\epsilon}_V$, $\sup_{x \in \mathcal{X}} \left\|\epsilon_Q\left(x\right)\right\| < \overline{\epsilon}_Q$, and $\sup_{x \in \mathcal{X}} \left\|\nabla \epsilon_Q\left(x\right)\right\| < \overline{\epsilon}_Q$ (see, e.g., [33, Theorem 1.5]).

Let $\hat{V} : \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}$, defined as $\hat{V}(x, \hat{W}_V) \coloneqq \hat{W}_V^T \sigma_V\left(x\right)$ and $\hat{Q} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$, defined as $\hat{Q}(x, \hat{W}_Q) \coloneqq \hat{W}_Q^T \sigma_Q(x)$ be parameterized estimates of V^* and Q, respectively, where $\hat{W}_V$ and $\hat{W}_Q$ are estimates of W_V^* and W_Q^*, respectively. Furthermore, let $u^T R u$ be parameterized as $u^T R u = \left(W_R^*\right)^T \sigma_{R1}(u)$ where $\sigma_{R1} : \mathbb{R}^m \to \mathbb{R}^M$, is defined as $\sigma_{R1}(u) \coloneqq \left[u_{(1)}^2, 2u_{(1)}u_{(2)}, \ldots, 2u_{(1)}u_{(m)}, u_{(2)}^2, 2u_{(2)}u_{(3)}, \ldots, 2u_{(2)}u_{(m)}, u_{(3)}^2, \ldots, u_{(m-1)}^2, 2u_{(m-1)}u_{(m)}, u_{(m)}^2\right]^T$, and $W_R^* \in \mathbb{R}^M$ are the ideal weights, given by $W_R^* = \left[R_{11}, R_1^{(-1)}, R_{22}, R_2^{(-2)}, \ldots, R_{m-1,m-1}, R_{m-1}^{-(m-1)}, R_{mm}\right]^T$, where, for a given matrix $R \in \mathbb{R}^{m \times m}$, R_{ij} denotes the element in the i-th row and the j-th column, and $R_i^{(-j)}$ denotes the i-th row of the matrix R with the first j elements removed, e.g., $R_3^{(-3)} \coloneqq \left[R_{34}, R_{35}, \ldots, R_{3(m-1)}, R_{3m}\right]$.

Using the estimates $\hat{W}_V$, $\hat{W}_Q$, and $\hat{W}_R$ in (3) yields the

IBE $\delta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+M} \to \mathbb{R}$, given by

$$\delta\left(x, u, \hat{W}'\right) = \hat{W}_V^T \nabla_x \sigma_V(x)\left(f^o(x,u) + g(x,u)\right) + \hat{W}_Q^T \sigma_Q(x) + \hat{W}_R^T \sigma_{R1}(u), \quad (4)$$

where $\hat{W}' = \left[\hat{W}_V^T, \hat{W}_Q^T, \hat{W}_R^T\right]^T$.

Since (4) utilizes the dynamic model of the agent under observation, the IRL technique developed in this paper is model-based, and as such, an accurate model of the agent under observation is required to estimate the unknown reward function. To facilitate estimation under modeling uncertainty, a system identifier is utilized that estimates the unknown model parameters. Similarly, to remove the dependence of the inverse Bellman error on the full state x, a state estimator is also utilized. To keep the discussion focused on IRL, we assume that a state and parameter estimator that satisfies properties outlined in the subsequent Assumption 4 is available. Examples of such state and parameter estimators can be found in results such as [34–37].

4.2 State and Parameter Estimation

The unknown function g in (1) can be represented as

$$g(x,u) = \theta^T \sigma(x,u) + \epsilon(x,u), \quad (5)$$

where $\sigma : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $\epsilon : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denote the vector of continuously differentiable basis functions and the approximation error, respectively, and $\theta \in \mathbb{R}^{p \times n}$ is a constant matrix of unknown parameters. The basis functions are selected such that the approximation of g and its derivatives is uniform over the compact set $\mathcal{X} \times \mathcal{U}$ so that given any constant $\overline{\epsilon}$, there exist $p \in \mathbb{N}$ and $\overline{\sigma}, \overline{\theta} \in \mathbb{R}_{>0}$ such that $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\sigma(x,u)\right\| < \overline{\sigma}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\nabla\sigma(x,u)\right\| < \overline{\sigma}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\epsilon(x,u)\right\| < \overline{\epsilon}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\nabla\epsilon(x,u)\right\| < \overline{\epsilon}$, and $\|\theta\| < \overline{\theta}$ (see, e.g., [33, Theorem 1.5]).

To focus the discussion on IRL, it is assumed that estimates $\hat{\theta}$ and $\hat{x}$, of the parameters θ and the state x, respectively, are generated using a state and parameter estimator that satisfies the following property.

Assumption 4 *The state and parameter estimation errors, $\tilde{x} := x - \hat{x}$ and $\tilde{\theta} := \theta - \hat{\theta}$, satisfy $\left\|\tilde{x}(t)\right\| \leq \beta_{\tilde{x}}\left(\left\|\tilde{x}(0)\right\|, t\right) + \overline{X}/2$ and $\left\|\tilde{\theta}(t)\right\| \leq \beta_{\tilde{\theta}}\left(\left\|\tilde{\theta}(0)\right\|, t\right) + \overline{\Theta}/2$, for some real numbers $\overline{X} \geq 0$ and $\overline{\Theta} \geq 0$ and class $\mathcal{KL}$ functions $\beta_{\tilde{x}} : \mathbb{R} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $\beta_{\tilde{\theta}} : \mathbb{R} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$.*

Assumption 4 implies the existence of compact sets $\hat{\mathcal{X}} \subseteq \mathbb{R}^n$ and $\hat{\Theta} \subseteq \mathbb{R}^p$, such that $\hat{x}(t) \in \hat{\mathcal{X}}$ and $\hat{\theta}(t) \in \hat{\Theta}$, $\forall t \in \mathbb{R}_{\geq 0}$, and the existence of $\overline{T} \geq 0$ such that $\left\|\tilde{\theta}(t)\right\| \leq \overline{\Theta}$, and $\left\|\tilde{x}(t)\right\| \leq \overline{X}$, for all $t \geq \overline{T}$. The state and parameter estimator is implemented synchronously with inverse reinforcement learning, and in real-time.

5 Inverse Reinforcement Learning Utilizing Trajectory Information

5.1 Approximation of the Inverse Bellman Error

In this section, the state and parameter estimates are utilized to formulate an indirect error metric, called the approximate IBE, to facilitate IRL. Utilizing $\hat{x}$ and $\hat{\theta}$ from Assumption 4, and the parametric dynamics from (5), the IBE from (4) can be approximated as $\hat{\delta}\left(\hat{x}, u, \hat{W}', \hat{\theta}\right) = \hat{W}_V^T \nabla_x \sigma_V(\hat{x}) \hat{Y}(\hat{x}, u, \hat{\theta}) + \hat{W}_Q^T \sigma_Q(\hat{x}) + \hat{W}_R^T \sigma_{R1}(u)$, where $\hat{Y}(\hat{x}, u, \hat{\theta}) = f^o(\hat{x}, u) + \hat{\theta}^T \sigma(\hat{x}, u)$. Rearranging, we get $\hat{\delta}\left(\hat{x}, u, \hat{W}', \hat{\theta}\right) = \left(\hat{W}'\right)^T \sigma'\left(\hat{x}, u, \hat{\theta}\right)$, where $\sigma'\left(\hat{x}, u, \hat{\theta}\right) := \left[\left(\nabla_x \sigma_V(\hat{x}) \hat{Y}(\hat{x}, u, \hat{\theta})\right)^T, \left(\sigma_Q(\hat{x})\right)^T, \left(\sigma_{R1}(u)\right)^T\right]^T$. In the *ideal case*, i.e., if $x(\cdot)$ and $u(\cdot)$ are optimal with respect to the reward function in (2), the approximation of V^*, Q and g is exact (i.e., $\epsilon_V = \epsilon_Q = \epsilon = 0$), and $\tilde{x}$ and $\tilde{\theta}$ are equal to zero, then the IBE is equal to zero whenever $\hat{W}' = \left[\left(W_V^*\right)^T, \left(W_Q^*\right)^T, \left(W_R^*\right)^T\right]^T$. Therefore, the IBE is an indirect metric for the quality of a given set of weight estimates.

Candidate solutions to the IRL problem can thus be generated by minimizing the approximate IBE. It can be seen that in the ideal case, $\hat{W}' = 0$ trivially minimizes approximate IBE. Existence of the trivial solution is expected because minimization of any positive constant multiple of a reward function generates identical optimal trajectories, and as such, the IRL problem can only be solved up to a scaling factor. As a result, there is no loss of generality in arbitrarily assigning a value to one of the reward function weights. In this paper, it is assumed that the first element, R_{11}, of $\hat{W}_R$ is selected arbitrarily.

The approximate IBE can then be expressed as $\hat{\delta}'\left(\hat{x}, u, \hat{W}, \hat{\theta}\right) = \hat{W}^T \sigma''\left(\hat{x}, u, \hat{\theta}\right) + R_{11} u_1^2$, where $\hat{W} := \left[\hat{W}_V^T, \hat{W}_Q^T, \left(\hat{W}_R^{(-1)}\right)^T\right]$, and $\sigma''\left(\hat{x}, u, \hat{\theta}\right) := \left[\left(\nabla_x \sigma_V(\hat{x}) \hat{Y}(\hat{x}, u, \hat{\theta})\right)^T, \left(\sigma_Q(\hat{x})\right)^T, \left(\sigma_{R1}^{(-1)}(u)\right)^T\right]^T$.

To estimate the unknown weights using the approximate IBE, one could update the weight estimates using

$$\dot{\hat{W}} = -K\sigma''\left(\hat{x},u,\hat{\theta}\right)\left(\left(\sigma''\left(\hat{x},u,\hat{\theta}\right)\right)^T\hat{W}+R_{11}u_1^2\right),$$

where K is a gain matrix. The dynamics of the state estimation error can then be expressed as a perturbed linear time-varying system with $\sigma''\left(\hat{x},u,\hat{\theta}\right)\left(\sigma''\left(\hat{x},u,\hat{\theta}\right)\right)^T$ as the system matrix. However, such a formulation requires persistence of excitation for boundedness and convergence of the estimation error [38–41]. The features σ'' of the IBE are nonlinear, and as such, ensuring persistence of excitation a priori and monitoring persistence of excitation online are generally difficult.

To relax the PE requirement and help ensure boundedness of the weight estimation errors under loss of excitation, the IRL method developed in this paper borrows the idea of history stacks from concurrent learning (CL) adaptive control [42–44]. A history stack at time t, denoted by $\mathcal{H}^{IRL}(t)$, is a collection of values of $\hat{x}(\cdot)$ and $u(\cdot)$, recorded at judiciously selected time instances $t_1(t) < t_2(t) < \ldots < t_N(t) \leq t$. Using the history stack, the approximate IBE, evaluated along the trajectories $\hat{x}(\cdot)$ and $u(\cdot)$, at time instances $t_1(t), t_2(t), \ldots, t_N(t)$, can be compiled in the matrix form

$$\Delta'(t,\hat{W}) = \hat{\Sigma}'(t)\hat{W} + R_{11}\left[u_{(1)}^2(t_1(t)),\ldots,u_{(1)}^2(t_N(t))\right]^T, \tag{6}$$

where

$$\Delta'(t,\hat{W}) \coloneqq \begin{bmatrix} \hat{\delta}'\left(\hat{x}\left(t_1(t)\right),u\left(t_1(t)\right),\hat{W},\hat{\theta}\left(t_1(t)\right)\right) \\ \vdots \\ \hat{\delta}'\left(\hat{x}\left(t_N(t)\right),u\left(t_N(t)\right),\hat{W},\hat{\theta}\left(t_N(t)\right)\right) \end{bmatrix}$$

and

$$\hat{\Sigma}'(t) \coloneqq \begin{bmatrix} \left(\sigma''\left(\hat{x}\left(t_1(t)\right),u\left(t_1(t)\right),\hat{\theta}\left(t_1(t)\right)\right)\right)^T \\ \vdots \\ \left(\sigma''\left(\hat{x}\left(t_N(t)\right),u\left(t_N(t)\right),\hat{\theta}\left(t_N(t)\right)\right)\right)^T \end{bmatrix}.$$

Further information about the weight estimates is gained by leveraging the optimal policy.

5.2 Optimality Check and the Control Residual Error

If u is the optimal action in response to the state x, then [1] $u = -\frac{1}{2}R^{-1}\left(\nabla_u f(x)\right)^T\left(\nabla_x V^*(x)\right)^T$. That is,

$$\begin{aligned} -2Ru = &\left(\nabla_u f^o(x) + \theta^T\nabla_u\sigma(x)\right)^T\left(\nabla_x\sigma_V(x)\right)^T W_V^* \\ &+ \left(\nabla_u f^o(x) + \theta^T\nabla_u\sigma(x)\right)^T\left(\nabla_x\epsilon_V(x)\right)^T \\ &+ \left(\nabla_u\epsilon(x)\right)^T\left(\left(\nabla_x\sigma_V(x)\right)^T W_V^* + \left(\nabla_x\epsilon_V(x)\right)^T\right). \end{aligned} \tag{7}$$

The product Ru can be linearly parameterized as $Ru = \sigma_{R2}(u)W_R^*$, with $\sigma_{R2}:\mathbb{R}^m \to \mathbb{R}^{m\times M}$ given by $\sigma_{R2}(u) =$

$$\begin{bmatrix} u^T & 0_{1\times m-1} & 0_{1\times m-2} & \cdots & 0 \\ u_{(1)}e_{2,m} & \left(u^{(-1)}\right)^T & 0_{1\times m-2} & \cdots & 0 \\ u_{(1)}e_{3,m} & u_{(2)}e_{2,m-1} & \left(u^{(-2)}\right)^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{(1)}e_{m,m} & u_{(2)}e_{m-1,m-1} & u_{(3)}e_{m-2,m-2} & \cdots & \left(u^{-(m-1)}\right)^T \end{bmatrix},$$

where $e_{i,j}$ denotes a row vector of size j, with a one in the $i-$th position and zeros everywhere else.

Using the estimates $\hat{W}_R$ and $\hat{W}_V$ in (7) for W_R^* and W_V^*, respectively, a control residual error $\Delta_u' : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+M} \to \mathbb{R}^m$ is obtained as

$$\begin{aligned} \Delta_u'(x,u,\hat{W}') = &2\sigma_{R2}(u)\hat{W}_R \\ &+ \left(\nabla_u f^o(x) + \theta^T\nabla_u\sigma(x)\right)^T\left(\nabla_x\sigma_V(x)\right)^T\hat{W}_V. \end{aligned} \tag{8}$$

Utilizing the history stack $\mathcal{H}^{IRL}(t)$ to subtract $0 = H\left(x(t_i(t)),\left(\nabla_x, V(x(t_i(t)))\right)^T, u(t_i(t))\right)$ from (6), substituting estimates $\hat{x}$ and $\hat{\theta}$ in (8), and appending (8) evaluated at $t_1(t),\ldots,t_N(t)$ to (6), with the arbitrarily assigned weight R_{11} removed, results in the linear system of equations

$$-\Sigma_{R1}(t) - \hat{\Sigma}(t)\hat{W} = \hat{\Sigma}(t)\tilde{W} + \Delta(t), \tag{9}$$

where the weight estimation error is defined as $\tilde{W} = W^* - \hat{W}$ with

$$W^* \coloneqq \left[(W_V^*)^T,\left(W_Q^*\right)^T,\left((W_R^*)^{(-1)}\right)^T\right]^T,$$

$$\hat{\Sigma}(t) \coloneqq \begin{bmatrix} \left(\sigma'''\left(\hat{x}\left(t_1(t)\right),u\left(t_1(t)\right),\hat{\theta}\left(t_1(t)\right)\right)\right)^T \\ \vdots \\ \left(\sigma'''\left(\hat{x}\left(t_N(t)\right),u\left(t_N(t)\right),\hat{\theta}\left(t_N(t)\right)\right)\right)^T \end{bmatrix},$$

$$\Sigma_{R1}(t) \coloneqq \Big[R_{11}\left(u_{(1)}(t_1(t))\right)^2, 2R_{11}u_{(1)}(t_1(t)), 0_{1\times(m-1)},\cdots,R_{11}\left(u_{(1)}(t_N(t))\right)^2, 2R_{11}u_{(1)}(t_N(t)),0_{1\times(m-1)}\Big]^T,$$

[1] Since f,σ, and ϵ are assumed to be affine in control, their partial derivatives with respect to u are independent of u.

$$\sigma'''\left(\hat{x}\left(t_i(t)\right),u\left(t_i(t)\right),\hat{\theta}\left(t_i(t)\right)\right) \coloneqq$$
$$\begin{bmatrix} \left(\sigma''\left(\hat{x}\left(t_i(t)\right),u\left(t_i(t)\right),\hat{\theta}\left(t_i(t)\right)\right)\right)^T \\ \left[G\left(\hat{x}\left(t_i(t)\right),\hat{\theta}\left(t_i(t)\right)\right),\ 0_{m\times L},\ 2\sigma_{R2}^{(-1)}\left(u(t_i(t))\right)\right] \end{bmatrix}^T,$$
$$G\left(\hat{x}\left(t_i(t)\right),\hat{\theta}\left(t_i(t)\right)\right) \coloneqq$$
$$\left(\nabla_u f^o(\hat{x}(t_i(t)))\right)^T\left(\nabla_x\sigma_V(\hat{x}(t_i(t)))\right)^T +$$
$$\left(\left(\hat{\theta}(t_i(t))\right)^T\nabla_u\sigma(\hat{x}(t_i(t)))\right)^T\left(\nabla_x\sigma_V(\hat{x}(t_i(t)))\right)^T,$$

and the residual Δ is independent of $\tilde{W}$.

Using the fact that the gradients of $(x,u)\mapsto f^o(x,u)$, $(x,u)\mapsto\sigma(x,u)$, $x\mapsto\sigma_V(x)$, and $x\mapsto\sigma_Q(x)$ are locally Lipschitz, the residual Δ can be bounded above by

$$\|\Delta(t)\| \le \overline{\Delta}_\epsilon + \bar{\tilde{x}}(t)\overline{\Delta}_{\tilde{x}} + \bar{\tilde{\theta}}(t)\overline{\Delta}_{\tilde{\theta}}, \tag{10}$$

where $\bar{\tilde{x}}(t) \coloneqq \max_{i=1,2,\ldots,N}\|\tilde{x}(t_i(t))\|$ and $\bar{\tilde{\theta}}(t) \coloneqq \max_{i=1,2,\ldots,N}\|\tilde{\theta}(t_i(t))\|$. Since $t\mapsto x(t), t\mapsto\hat{x}(t), t\mapsto\hat{\theta}(t)$, and $t\mapsto u(t)$ are bounded by Assumption 4, the bounds $\bar{\Delta}_\epsilon, \bar{\Delta}_{\tilde{x}}$, and $\overline{\Delta}_{\tilde{\theta}}$ can be selected independent of t_i and the specific trajectories $x(\cdot)$, $u(\cdot)$, and $\hat{x}(\cdot)$ currently stored in the history stack. The relationship in (9) is used in the following to derive adaptive update laws for estimation of the unknown weights.

5.3 Adaptive Update Laws

The unknown weights are estimated using the update law

$$\dot{\hat{W}} = \alpha\Gamma(t)\left(\hat{\Sigma}(t)\right)^T\left(-\hat{\Sigma}(t)\hat{W} - \Sigma_{R1}(t)\right), \tag{11}$$

where $\alpha\in\mathbb{R}_{>0}$ is a constant adaptation gain and $\Gamma:\mathbb{R}_{\ge 0}\to\mathbb{R}^{(L+P+m-1)\times(L+P+m-1)}$ is the least-squares gain updated using the update law

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)\Gamma, \tag{12}$$

and $\beta\in\mathbb{R}_{>0}$ is the forgetting factor. The update law in (11) is motivated by the fact that the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}} = -\alpha\Gamma(t)\left(\hat{\Sigma}(t)\right)^T\left(\hat{\Sigma}(t)\tilde{W} + \Delta(t)\right), \tag{13}$$

which can be shown to be a perturbed stable linear time-varying system under conditions detailed in the following section.

Analyzing (13), it can be seen that the rate of decay for the weight estimation errors is proportional to the minimum eigenvalue of the matrix $\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)$. To yield faster convergence, a minimum eigenvalue maximization algorithm (see Algorithm 1) is utilized to select the time instances $t_1(t),\ldots,t_N(t)$. Specifically, a new data point $(\hat{x}(t^*(t)),u(t^*(t)))$ replaces an existing data point $(\hat{x}(t_i(t)),u(t_i(t)))$, for some $i\in\{1,\ldots,N\}$, if the replacement results in the largest increase in the minimum eigenvalue of $\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)$ among all N possible replacements, i.e.,

$$\lambda_{\min}\left(\sum_{i\ne j}\hat{\Sigma}_i^T\hat{\Sigma}_i + \hat{\Sigma}_j^T\hat{\Sigma}_j\right) < \psi\lambda_{\min}\left(\sum_{i\ne j}\hat{\Sigma}_i^T\hat{\Sigma}_i + \left(\hat{\Sigma}^*\right)^T\hat{\Sigma}^*\right), \tag{14}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a matrix, $\hat{\Sigma}_i \coloneqq \hat{\Sigma}(t_i(t))$, $\hat{\Sigma}_j \coloneqq \hat{\Sigma}(t_j(t))$, $\hat{\Sigma}^* \coloneqq \hat{\Sigma}(t^*(t))$, and $\psi\in(0,1]$ is a threshold for replacement.

Algorithm 1 Algorithm for history stack purging with dwell time. At each time instance t, $\eta(t)$ stores the last time instance $\mathcal{H}$ was purged, $\Omega(t)$ stores the highest minimum eigenvalue encountered so far, τ denotes the dwell time, and $\xi\in(0,1]$ is a threshold for purging.

$\eta(0)\leftarrow 0$, $\Omega(0)\leftarrow 0$
if $\mathcal{G}^{IRL}$ is not full **then**
  add the data point to $\mathcal{G}^{IRL}$
else
  add the data point to $\mathcal{G}^{IRL}$ if (14) holds
end if
if $\lambda_{\min}\left(\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)\right) \ge \xi\Omega(t)$ **then**
  if $t-\eta(t)\ge\tau$ **then**
    $\mathcal{H}^{IRL}\leftarrow\mathcal{G}^{IRL}$ and $\mathcal{G}^{IRL}\leftarrow 0$ ▷ purge and replace $\mathcal{H}^{IRL}$
    $\eta(t)\leftarrow t$
    if $\Omega(t) < \lambda_{\min}\left(\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)\right)$ **then**
      $\Omega(t)\leftarrow\lambda_{\min}\left(\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)\right)$
    end if
  end if
end if

Since the size of the perturbation $\Delta(t)$ in (13) depends on the quality of the state and parameter estimates in the history stack $\mathcal{H}^{IRL}(t)$, a time-based purging algorithm is utilized to purge poor estimates $\hat{x}$ and $\hat{\theta}$ from the history stack. Since the state and parameter estimation errors are assumed to be bounded by a class of $\mathcal{KL}$ functions, newer estimates of $\hat{x}$ and $\hat{\theta}$ are expected to be better, and

are utilized when available by purging older estimates from the history stack.

The developed purging technique maintains an auxiliary transient history stack, labeled $\mathcal{G}^{IRL}$ populated using minimum eigenvalue maximization. When the transient history stack is full rank according to (15), $\mathcal{H}^{IRL}$ is emptied and $\mathcal{G}^{IRL}$ is copied into $\mathcal{H}^{IRL}$. The history stack $\mathcal{H}^{IRL}$ is kept constant in between purging instances. To avoid chattering, a constant $\tau \in \mathbb{R}_{>0}$ is selected so that a new purging event occurs only after τ seconds have passed since the previous purging event. Due to purging, the time instances $\{t_1, \cdots t_N\}$, the matrices $\hat{\Sigma}$ and Σ_{R1}, and consequently, the history stack $\mathcal{H}^{IRL}$, are piecewise constant in time.

5.4 Analysis of the Direct MBIRL Technique

Convergence of the estimation error to a neighborhood of the origin follows if the data is sufficiently informative according to the following definitions.

Definition 1 *The history stack $\mathcal{H}^{IRL}$, is called* full rank, uniformly in t, *if there exists $\underline{\sigma} \in \mathbb{R}_{>0}$ such that*[2] $\forall t \in \mathbb{R}_{\geq 0}$,

$$\underline{\sigma} < \lambda_{\min} \left\{ \left(\hat{\Sigma}(t) \right)^T \hat{\Sigma}(t) \right\}. \tag{15}$$

Definition 2 *The signal $(\hat{x}, u)$ is called* finitely informative *(FI) if there exist time instances $0 \leq t_1 < t_2 < \cdots < t_N$ such that the resulting history stack is full rank and* persistently informative *(PI) if for any $T \geq 0$, there exist time instances $T \leq t_1 < t_2 < \cdots < t_N$ such that the resulting history stack is full rank.*

The stability result is summarized in the following theorem.

Theorem 1 *If the unknown state variables and model parameters are estimated using a state and parameter estimator that satisfies Assumption 4, the signal $(\hat{x}, u)$ is FI, $\mathcal{H}^{IRL}$ is populated using Algorithm 1, and the weights are estimated using the update laws in* (11) *and* (12) *then $t \mapsto \tilde{W}(t)$ is ultimately bounded (UB).*[3]

[2] The history stack $\mathcal{H}^{IRL}(0)$ can be initialized using arbitrarily selected trajectories $\left(x(\cdot), \hat{x}(\cdot), u(\cdot)\right)$ to ensure that the history stack is full rank at $t = 0$.

[3] In the ideal case, if the history stack is full rank, then the linear system of equations in (9), with $\hat{W} = W^*$, admits a unique solution, and therefore, so does the IRL problem (once R_{11} is selected). As a result, the finite informativity requirement in Theorem 1 implicitly restricts the results in this paper to IRL problems that admit unique solutions up to a scaling factor.

PROOF. Consider the candidate Lyapunov function

$$U(\tilde{W}, t) = \frac{1}{2} \tilde{W}^T \Gamma^{-1}(t) \tilde{W}.$$

Using arguments similar to [41, Corollary 4.3.2], it can be shown that provided $\lambda_{\min}\left\{\Gamma^{-1}(0)\right\} > 0$, the least squares gain matrix satisfies

$$\underline{\Gamma} \mathrm{I}_{L+P+m-1} \leq \Gamma(t) \leq \overline{\Gamma} \mathrm{I}_{L+P+m-1}, \forall\, t \geq 0, \tag{16}$$

where $\underline{\Gamma}$ and $\overline{\Gamma}$ are positive constants. Using the bounds in (16), the candidate Lyapunov function can be shown to satisfy

$$\frac{1}{2\overline{\Gamma}} \left\| \tilde{W} \right\|^2 \leq U\left(\tilde{W}, t\right) \leq \frac{1}{2\underline{\Gamma}} \left\| \tilde{W} \right\|^2. \tag{17}$$

Using (12), (13), (15), and (16), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, and the Cauchy-Schwartz inequality, the orbital derivative of the candidate Lyapunov function along the solutions of (11) and (12) can be bounded by $\dot{U}(\tilde{W}, t) \leq -\frac{1}{2}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta\right) \left\| \tilde{W} \right\|^2 + \alpha \left\| \tilde{W} \right\| \left\| \hat{\Sigma}(t) \right\| \left\| \Delta(t) \right\|$. Using (10), $\dot{U}$ can be bounded as

$$\dot{U}(\tilde{W}, t) \leq -\frac{1}{4} \left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta \right) \left\| \tilde{W} \right\|^2, \ \forall \|\tilde{W}\| \geq \rho\left(\|\mu\|\right), \tag{18}$$

where $\mu = \left[\sqrt{\overline{\Delta}_\epsilon}, \sqrt{\overline{\bar{x}}}, \sqrt{\overline{\bar{\theta}}} \right]^T$,

$$\rho(\|\mu\|) = \left(\frac{4\alpha\overline{\Sigma} \max\{1, \overline{\Delta}_x, \overline{\Delta}_\theta\}}{\alpha\underline{\sigma} + \beta/\overline{\Gamma}} \right) \|\mu\|^2,$$

and $\overline{\Sigma}$ satisfies $\left\| \hat{\Sigma}(t) \right\| \leq \overline{\Sigma},\ \forall t \geq 0$. Since $t \mapsto x(t), t \mapsto \hat{x}(t), t \mapsto \hat{\theta}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 4, the bound $\overline{\Sigma}$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{x}$ currently stored in the history stack. Using (17) and (18), [45, Theorem 4.19] can be invoked to conclude that (13) is input-to-state stable (ISS) with state $\tilde{W}$ and input μ.

If Algorithm 1 is implemented and if the signal $(\hat{x}, u)$ is FI, then there exists a time instance T_s, such that for all $t \geq T_s$, the history stack $\mathcal{H}^{IRL}(t)$ remains unchanged. As a result, using Exercise 4.58 from [45], it can be concluded that the ultimate bound on $\tilde{W}$ can be expressed as

$$\limsup_{t \to \infty} \|\tilde{W}(t)\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}} \left(\frac{4\alpha\overline{\Sigma} \max\{1, \overline{\Delta}_x, \overline{\Delta}_\theta\}}{\alpha\underline{\sigma} + \beta/\overline{\Gamma}} \right) \overline{\Delta}_\epsilon$$

$$+\sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+\beta/\overline{\Gamma}}\right)\left(\overline{\tilde{x}}(T_s)+\overline{\tilde{\theta}}(T_s)\right), \tag{19}$$

where $\overline{\tilde{x}}(T_s)$ and $\overline{\tilde{\theta}}(T_s)$ denote bounds on the state and parameter estimation errors, respectively, in the history stack $\mathcal{H}^{IRL}(t)$ for all $t \geq T_s$.

Furthermore, if $(\hat{x}, u)$ is PI, then the limits $\limsup_{t\to\infty} \overline{\tilde{x}}(t) \to \overline{X}$ and $\limsup_{t\to\infty} \overline{\tilde{\theta}}(t) \to \overline{\Theta}$ imply that (19) reduces to

$$\limsup_{t\to\infty}\|\tilde{W}(t)\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+\beta/\overline{\Gamma}}\right)\overline{\Delta}_\epsilon$$
$$+\sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+\beta/\overline{\Gamma}}\right)\left(\overline{X}+\overline{\Theta}\right). \qquad \square$$

The ultimate bound decreases with decreasing approximation errors for the reward function, the value function, the system state, and the model parameters. Since an exact basis for parameterization of the value function is not assumed to be known in Theorem 1, the weight estimation errors cannot be expected to decay to zero. Furthermore, if the observed dataset is richer, i.e., the minimum eigenvalue of the history stack in Algorithm 1 is larger, then the ultimate bound is smaller and the convergence rate is faster.

If the signal (x, u) is PI, the actual reward function and the optimal value function are linearly parameterizable with a known basis, and if the state and parameter estimator is exact and converges in finite time (which is possible if the system dynamics are linearly parameterizable with a known basis, either under persistence of excitation [46], or under finite excitation with full state feedback [47, 48]), then the estimated reward function converges to the true reward function. This observation is formalized in the following corollary, which follows from Theorem 1.

Corollary 1 *Under the hypothesis of Theorem 1, if $\overline{\Theta}$, $\overline{X}$, ϵ_Q, and ϵ_V are zero, and the signal $(\hat{x}, u)$ is PI, then* $\lim_{t\to\infty} \|\tilde{W}(t)\| = 0$.

Remark 1 *If the full state is measurable, the smoothness restrictions on the agent model and the basis functions can be relaxed to continuous differentiability.*

6 Feedback-Driven Inverse Reinforcement Learning

In optimal control problems that are aimed at driving the state to a set-point or an error signal to zero, information content of the state and control trajectories can quickly decay to zero. As a result, the reward function estimate may never converge. In this case, artificially generated state-action pairs can be used to drive estimation. In addition, even if sufficient excitation exists to estimate the unknown reward function directly, artificially generated state-action pairs can provide additional data, potentially resulting in faster estimation of the reward function. Motivated by the observation that knowledge of the optimal policy can be leveraged to artificially synthesize data to drive IRL, the following section develops a feedback-driven MBIRL technique that utilizes an estimate of the optimal policy.

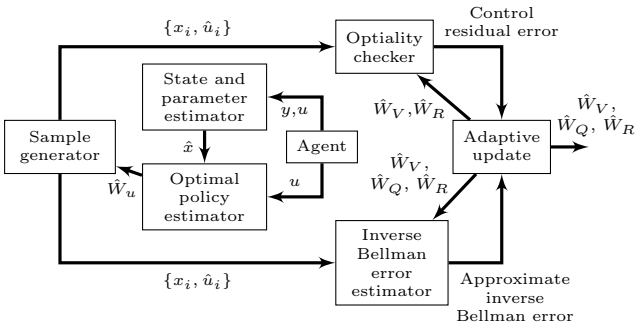

Fig. 2. Block diagram of the developed Feedback-Driven IRL method.

A block diagram that illustrates the Feedback-Driven IRL method is shown in Fig. 2.

6.1 Optimal Policy Estimation

The closed-form nonlinear optimal policy corresponding to the reward structure in (2) is

$$u^*(x) = -\frac{1}{2}R^{-1}\left(\nabla_u f(x)\right)^T\left(\nabla_x V^*(x)\right)^T. \tag{20}$$

To facilitate estimation, u^* will be represented as

$$u^*(x) = -\left(W_u^*\right)^T\sigma_u\left(x\right) + \epsilon_u(x), \tag{21}$$

where $W_u^* \in \mathbb{R}^{K\times m}$ is a matrix of unknown ideal constant parameters, $\sigma_u : \mathbb{R}^n \to \mathbb{R}^K$ are known continuously differentiable features, and $\epsilon_u : \mathbb{R}^n \to \mathbb{R}^m$ is the resulting approximation error. The basis functions are selected such that the approximation of u^* and its derivatives are uniform over the compact set $\mathcal{X}\times\mathcal{U}$ so that given any constant $\overline{\epsilon}_u$, there exist $K \in \mathbb{N}$ and $\overline{\sigma}_u \in \mathbb{R}_{>0}$ such that $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\sigma_u\left(x,u\right)\right\| < \overline{\sigma}_u$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\nabla\sigma_u\left(x,u\right)\right\| < \overline{\sigma}_u$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\epsilon_u\left(x,u\right)\right\| < \overline{\epsilon}_u$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\nabla\epsilon_u\left(x,u\right)\right\| < \overline{\epsilon}_u$ (see, e.g., [33, Theorem 1.5]).

Collecting values of the state estimates and the control signals over time instances, $t_1^u(t), t_2^u(t), \cdots, t_M^u(t)$, in a

history stack, denoted as $\mathcal{H}^u(t)$, and using the fact that if $u(t)$ is the optimal action in response to the state $x(t)$, i.e., $u(t) = u^*(x(t))$, (21) can be reformulated into the matrix form

$$-\Sigma_u(t) - \hat{\Sigma}_\sigma(t)\hat{W}_u = \hat{\Sigma}_\sigma(t)\tilde{W}_u - \Delta_u(t), \tag{22}$$

where the weight estimation error is defined as $\tilde{W}_u = W_u^* - \hat{W}_u$, $\Sigma_u(t) \coloneqq [u(t_1(t)), \cdots, u(t_M(t))]^T$, $\hat{\Sigma}_\sigma(t) \coloneqq [\sigma_u(\hat{x}(t_1(t))), \cdots, \sigma_u(\hat{x}(t_M(t)))]^T$, the residual Δ_u depends on ϵ_u and $\tilde{x}$, and the time instances $t_1^u, \ldots, t_M^u$ are selected according to minimum eigenvalue maximization.

Since $x \mapsto \sigma_u(x)$ is continuously differentiable, the residual Δ_u can be bounded above by

$$\|\Delta_u(t)\| \leq \overline{\Delta}_u + L_u \bar{\tilde{x}}(t), \tag{23}$$

where $\bar{\tilde{x}}(t) = \max_{i=1,2,\ldots,M} \|\tilde{x}(t_i(t))\|$. Since $t \mapsto x(t), t \mapsto \hat{x}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 4, the bound $\overline{\Delta}_u$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{x}$ currently stored in the history stack.

The relationship in (22) suggests the following update law for estimation of the unknown weights

$$\dot{\hat{W}}_u = \alpha_u \Gamma_u(t) \left(\hat{\Sigma}_\sigma(t)\right)^T \left(-\Sigma_u(t) - \hat{\Sigma}_\sigma(t)\hat{W}_u\right), \tag{24}$$

where $\alpha_u \in \mathbb{R}_{>0}$ is a constant adaptation gain, and $\Gamma_u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{K\times K}$ is the least-squares gain updated using the update law

$$\dot{\Gamma}_u = \beta_u \Gamma_u - \alpha_u \Gamma_u \left(\hat{\Sigma}_\sigma(t)\right)^T \hat{\Sigma}_\sigma(t)\Gamma_u, \tag{25}$$

where $\beta_u \in \mathbb{R}_{>0}$ is the forgetting factor.

The update law in (22) is motivated by the fact that the dynamics of the weight estimation error can be described by

$$\dot{\tilde{W}}_u = -\alpha_u \Gamma_u(t) \left(\hat{\Sigma}_\sigma(t)\right)^T \left(\hat{\Sigma}_\sigma(t)\tilde{W}_u - \Delta_u(t)\right), \tag{26}$$

which can be shown to be a perturbed stable linear time-varying system under conditions detailed in the following section.

6.2 Analysis of the Optimal Policy Estimator

Convergence of the estimation error to a neighborhood of the origin follows if the data are sufficiently informative according to the following definitions.

Definition 3 *The time-varying history stack, $\mathcal{H}^u$, is called full rank, uniformly in t, if there exists a $\underline{k} > 0$ such that*[4] *$\forall t \in \mathbb{R}_{\geq 0}$,*

$$\underline{k} < \lambda_{\min}\left\{\left(\hat{\Sigma}_\sigma(t)\right)^T \hat{\Sigma}_\sigma(t)\right\}. \tag{27}$$

Using arguments similar to [41, Corollary 4.3.2], it can be shown that if $\lambda_{\min}\left\{\Gamma_u^{-1}(0)\right\} > 0$, and if $\mathcal{H}^u$ is full rank, uniformly in t, then the least squares gain matrix satisfies

$$\underline{\Gamma}_u \mathrm{I}_K \leq \Gamma_u(t) \leq \overline{\Gamma}_u \mathrm{I}_K, \forall\, t \geq 0, \tag{28}$$

where $\underline{\Gamma}_u$ and $\overline{\Gamma}_u$ are positive constants.

Theorem 2 *If the unknown state variables and model parameters are estimated using a state and parameter estimator that satisfies Assumption 4, the signal $(\hat{x}, u)$ is FI, $\mathcal{H}^u$ is populated using a time-based purging algorithm similar to Algorithm 1, and the weights are estimated using the update laws in* (24) *and* (25)*, then $t \mapsto \tilde{W}_u(t)$ is ultimately bounded.*

PROOF. Consider the following positive definite candidate Lyapunov function

$$U_u(\tilde{W}_u, t) = \mathrm{tr}(\tilde{W}_u^T \Gamma_u^{-1}(t)\tilde{W}_u), \tag{29}$$

Using the bounds in (28), the candidate Lyapunov function satisfies

$$\frac{1}{\overline{\Gamma}_u}\left\|\tilde{W}_u\right\|^2 \leq U_u\left(\tilde{W}_u, t\right) \leq \frac{1}{\underline{\Gamma}_u}\left\|\tilde{W}_u\right\|^2. \tag{30}$$

Taking the orbital derivative of (29), using (25), (26), (27) and (28), along with the identity $\dot{\Gamma}_u^{-1} = -\Gamma_u^{-1}\dot{\Gamma}_u\Gamma_u^{-1}$ and using the Cauchy-Schwartz inequality, $\dot{U}_u$ can be bounded by $\dot{U}_u(\tilde{W}_u, t) \leq -\left(\alpha_u \underline{k} + \frac{\beta_u}{\overline{\Gamma}_u}\right)\left\|\tilde{W}_u\right\|^2 + 2\alpha_u \left\|\tilde{W}_u\right\| \left\|\hat{\Sigma}_\sigma(t)\right\| \|\Delta_u(t)\|$. Using (23), $\dot{U}_u$ can be bounded as

$$\dot{U}_u(\tilde{W}_u, t) \leq -\frac{1}{2}\left(\alpha_u \underline{k} + \frac{\beta_u}{\overline{\Gamma}_u}\right)\left\|\tilde{W}_u\right\|^2, \forall \|\tilde{W}_u\| \geq \rho\left(\|\mu\|\right), \tag{31}$$

where $\rho\left(\|\mu\|\right) = \left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u}\right)\|\mu\|^2$, $\mu = \left[\sqrt{\overline{\Delta}_u}, \sqrt{\bar{\tilde{x}}}\right]^T$, and $\overline{\Sigma}_\sigma$ is an upper bound of $\|\hat{\Sigma}_\sigma(t)\|$,

[4] The history stack $\mathcal{H}^u(0)$ can be initialized using arbitrarily selected trajectories $\left(\hat{x}(\cdot), u(\cdot)\right) \in \hat{\mathcal{X}} \times \mathcal{U}$ to ensure that the history stack is full rank for all $t \geq 0$.

$\forall t \geq 0$. Since $t \mapsto \hat{x}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 4, the bound $\overline{\Sigma}_\sigma$ can be selected independent of t_i and the specific trajectories of u and $\hat{x}$ that are currently stored in the history stack. Using (30) and (31), [45, Theorem 4.19] can be invoked to conclude that (26) is input-to-state stable with state $\tilde{W}_u$ and input μ.

If a time-based purging algorithm, similar to Algorithm 1, is implemented and if the signal $(\hat{x}, u)$ is FI, there exists a time instance T_s, such that for all $t \geq T_s$, the history stack $\mathcal{H}^u(t)$ remains unchanged. As a result, using Exercise 4.58 from [45], it can be concluded that the ultimate bound on $\tilde{W}_u$ can be expressed as

$$\limsup_{t\to\infty} \|\tilde{W}_u(t)\| \leq \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}} \left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u} \right) \overline{\Delta}_u + \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}} \left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u} \right) \overline{\overline{x}}(T_s).$$

Furthermore, if $(\hat{x}, u)$ is PI, then the bound can be reduced to

$$\limsup_{t\to\infty} \|\tilde{W}_u(t)\| \leq \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}} \left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u} \right) \overline{\Delta}_u + \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}} \left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u} \right) \overline{X} =: \overline{\gamma}_u. \qquad \square$$

Remark 2 *Theorem 2 implies existence of a compact set $\hat{\mathcal{U}} \subseteq \mathbb{R}^m$, such that $\hat{u}(t) \in \hat{\mathcal{U}}$, $\forall t \in \mathbb{R}_{\geq 0}$.*

Remark 3 *If the full state is measurable, the optimal controller estimate converges exponentially, see [31].*

6.3 Sample Generation for Feedback-driven Inverse Reinforcement Learning

In this section, the optimal feedback estimator developed in the previous section is utilized to create a data-set of estimated near-optimal state-action pairs to drive IRL.

For each time t_i, select an arbitrary state, denoted by x_i, and let $\hat{u}_i \coloneqq -\left(\hat{W}_u(t_i)\right)^T \sigma_u(x_i)$ be the estimate of the optimal controller u_i at state x_i and time t_i. The IBE, evaluated at the arbitrarily selected state x_i and at time t_i, using estimates of the model and the optimal policy, is then given by

$$\delta''\left(x_i, \hat{u}_i, \hat{W}', \hat{\theta}(t_i)\right) = \left(\hat{W}'\right)^T \sigma'\left(x_i, \hat{u}_i, \hat{\theta}(t_i)\right), \quad (32)$$

where $\hat{W}' \coloneqq \left[\hat{W}_V^T, \hat{W}_Q^T, \hat{W}_R^T\right]^T$, and $\sigma'\left(x_i,\hat{u}_i,\hat{\theta}(t_i)\right) \coloneqq \Big[\big(\sigma(x_i,\hat{u}_i)\big)^T \hat{\theta}(t_i) \big(\nabla_x \sigma_V(x_i)\big)^T + \big(f^o(x_i,\hat{u}_i)\big)^T \big(\nabla_x \sigma_V(x_i)\big)^T, \big(\sigma_Q(x_i)\big)^T, \big(\sigma_{R1}(\hat{u}_i)\big)^T\Big]^T$. Taking the first element, R_{11}, of $\hat{W}_R$ to be known, IBE in (32) can be expressed as

$$\delta''\left(x_i, \hat{u}_i, \hat{W}, \hat{\theta}(t_i)\right) = \hat{W}^T \sigma''\left(x_i, \hat{u}_i, \hat{\theta}(t_i)\right) + R_{11}\hat{u}_{i1}^2, \quad (33)$$

where $\hat{W} \coloneqq \left[\hat{W}_V^T, \hat{W}_Q^T, \left(\hat{W}_R^{(-1)}\right)^T\right]^T$, $\hat{u}_{i1}$ denotes the first element of the vector $\hat{u}_i$, and $\sigma''\left(x_i,\hat{u}_i,\hat{\theta}(t_i)\right) \coloneqq \Big[\big(\sigma(x_i,\hat{u}_i)\big)^T \hat{\theta}(t_i) \big(\nabla_x \sigma_V(x_i)\big)^T + \big(f^o(x_i,\hat{u}_i)\big)^T \big(\nabla_x \sigma_V(x_i)\big)^T, \big(\sigma_Q(x_i)\big)^T, \Big(\sigma_{R1}^{(-1)}(\hat{u}_i)\Big)^T\Big]^T$.

In feedback-driven MBIRL, the history stack $\mathcal{H}^{IRL}$ contains a set of ordered pairs of parameter estimates, $\hat{\theta}(t_i)$, and data pairs, $(x_i, \hat{u}_i)$, collected over time instance $t_1, t_2, \ldots, t_N$ into matrices $\left(\hat{\Sigma}, \hat{\Sigma}_{R1}\right)$ (introduced in (34)). Similar to Section 5, the history stack contains potentially poor estimates of u_i and θ. Since the control estimation error and the parameter estimation error both decay exponentially to an ultimate bound, a time-based purging algorithm similar to Section 4 is needed to remove the erroneous estimates from the history stack once newer estimates become available. As a result, the data points $(x_i, \hat{u}_i)$ and the time instance t_i are time-varying.

Utilizing estimates $\hat{\theta}(t_i)$ and data pairs $(x_i, \hat{u}_i)$ in (20), subtracting $0 = H\left(x_i, \left(\nabla_x V(x_i)\right)^T, u_i\right)$ from (33), where $u_i \coloneqq u^*(x_i)$ is the ideal value of $\hat{u}_i$, evaluating (33) at time instances $\{t_i\}_{i=1}^N$, and stacking the results in a matrix form, we get

$$-\hat{\Sigma}(t)\hat{W} - \hat{\Sigma}_{R1}(t) = \hat{\Sigma}(t)\tilde{W} - \Delta(t), \quad (34)$$

where the weight estimation error is defined as $\tilde{W} = W^* - \hat{W}$ with

$$W^* \coloneqq \left[\left(W_V^*\right)^T, \left(W_Q^*\right)^T, \left(\left(W_R^*\right)^{(-1)}\right)^T \right]^T,$$

$$\hat{\Sigma}(t) \coloneqq \begin{bmatrix} \left(\sigma'''\left(x_1(t), \hat{u}_1(t), \hat{\theta}\left(t_1(t)\right)\right)\right)^T \\ \vdots \\ \left(\sigma'''\left(x_N(t), \hat{u}_N(t), \hat{\theta}\left(t_N(t)\right)\right)\right)^T \end{bmatrix},$$

$$\hat{\Sigma}_{R1}(t) \coloneqq \big[R_{11}\hat{u}_{1(1)}^2(t), 2R_{11}\hat{u}_{1(1)}(t), 0_{1\times(m-1)}, \cdots, R_{11}\hat{u}_{N(1)}^2(t), 2R_{11}\hat{u}_{N(1)}(t), 0_{1\times(m-1)}\big]^T,$$

where

$$\sigma'''\left(x_i(t),\hat{u}_i(t),\hat{\theta}(t_i(t))\right) \coloneqq$$
$$\begin{bmatrix} \left(\sigma''\left(x_i(t),\hat{u}_i(t),\hat{\theta}(t_i(t))\right)\right)^T \\ \left[G\left(x_i(t),\hat{\theta}(t_i(t))\right),\ 0_{m\times L},\ 2\hat{\sigma}_{R2}^{(-1)}(\hat{u}_i(t))\right] \end{bmatrix}^T,$$
$$G\left(x_i(t),\hat{\theta}(t_i(t))\right) \coloneqq$$
$$\left(\nabla_u f^o(x_i(t)) + \left(\hat{\theta}(t_i(t))\right)^T \nabla_u \sigma(x_i(t))\right)^T \left(\nabla_x \sigma_V(x_i(t))\right)^T,$$

and the residual Δ depends on ϵ, ϵ_Q, ϵ_V, $\tilde{\theta}$, and $\tilde{u}_i \coloneqq u_i - \hat{u}_i, \forall i \in [1,\ldots,N]$.

Since $(x,u) \mapsto f(x,u)$, $(x,u) \mapsto \sigma(x,u)$, $u \mapsto \sigma_{R1}(u)$, and $u \mapsto \sigma_{R2}(u)$ are continuously differentiable, the term $\left\|\Delta(t)\right\|$ can be bounded above by

$$\left\|\Delta(t)\right\| \leq \overline{\Delta}_\epsilon + \overline{\tilde{u}}(t)\overline{\Delta}_{\tilde{u}} + \overline{\tilde{\theta}}(t)\overline{\Delta}_{\tilde{\theta}}, \tag{35}$$

where $\overline{\tilde{u}}(t) = \max_{i=1,2,\ldots,N} \|\tilde{u}_i(t)\|$. Since $t \mapsto x(t), t \mapsto \hat{u}(t), t \mapsto u(t)$ and $t \mapsto \hat{\theta}(t)$ are bounded by Assumption 4, the bounds $\overline{\Delta}_\epsilon$, $\overline{\Delta}_{\tilde{u}}$, and $\overline{\Delta}_{\tilde{\theta}}$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{u}$ currently stored in the history stack.

6.4 Feedback-driven Adaptive Update Law

The relationship in (34) suggests the following update law for estimation of the unknown reward function weights

$$\dot{\hat{W}} = \alpha\Gamma(t)\left(\hat{\Sigma}(t)\right)^T\left(-\hat{\Sigma}(t)\hat{W} - \hat{\Sigma}_{R1}(t)\right), \tag{36}$$

where $\alpha \in \mathbb{R}_{>0}$ is a constant adaptation gain and $\Gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}^{(L+P+m-1)\times(L+P+m-1)}$ is the least-squares gain updated using the update law

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\left(\hat{\Sigma}(t)\right)^T\hat{\Sigma}(t)\Gamma, \tag{37}$$

and $\beta \in \mathbb{R}_{>0}$ is the forgetting factor. The update law in (36) is motivated by the fact that the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}} = -\alpha\Gamma(t)\left(\hat{\Sigma}(t)\right)^T\left(\hat{\Sigma}(t)\tilde{W} - \Delta(t)\right), \tag{38}$$

which can be shown to be a perturbed stable linear time-varying system under conditions detailed in the following section.

6.5 Analysis of Feedback-Driven Inverse Reinforcement Learning

Using arguments similar to [41, Corollary 4.3.2], it can be shown that if $\lambda_{\min}\left\{\Gamma^{-1}(0)\right\} > 0$, and if $\mathcal{H}^{IRL}$ is full rank, uniformly in t, then the least squares gain matrix satisfies

$$\underline{\Gamma}\mathrm{I}_{L+P+m-1} \leq \Gamma(t) \leq \overline{\Gamma}\mathrm{I}_{L+P+m-1}, \forall\, t \geq 0, \tag{39}$$

where $\underline{\Gamma}$ and $\overline{\Gamma}$ are positive constants.

The stability result is summarized in the following theorem.

Theorem 3 *If the unknown state variables and model parameters are estimated using a state and parameter estimator that satisfies Assumption 4, the signal $(\hat{x},u)$ and sequence $(x_i,\hat{u}_i)$ are FI, $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$ are populated using Algorithm 1, and the weights are estimated using the update laws in* (36) *and* (37)*, then $t \mapsto \tilde{W}(t)$ is ultimately bounded.*

PROOF. Consider the positive definite candidate Lyapunov function

$$U(\tilde{W},t) = \frac{1}{2}\tilde{W}^T\Gamma^{-1}(t)\tilde{W}.$$

Using the bounds in (39), the candidate Lyapunov function satisfies

$$\frac{1}{2\overline{\Gamma}}\left\|\tilde{W}\right\|^2 \leq U\left(\tilde{W},t\right) \leq \frac{1}{2\underline{\Gamma}}\left\|\tilde{W}\right\|^2. \tag{40}$$

Using (15), (37), (39) and (38), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, and using the Cauchy-Schwartz inequality, the orbital derivative of the candidate Lyapunov function, along the trajectories of (36) and (37) can be expressed as $\dot{U}(\tilde{W},t) \leq -\frac{1}{2}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta\right)\left\|\tilde{W}\right\|^2 + \alpha\|\tilde{W}\|\|\hat{\Sigma}(t)\|\|\Delta(t)\|$. Using (35), $\dot{U}$ can be bounded as

$$\dot{U}(\tilde{W},t) \leq -\frac{1}{4}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta\right)\left\|\tilde{W}\right\|^2, \forall \|\tilde{W}\| \geq \rho\left(\|\mu\|\right), \tag{41}$$

where $\rho\left(\|\mu\|\right) = \left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_u,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+\frac{1}{\overline{\Gamma}}\beta}\right)\|\mu\|^2$, $\mu = \left[\sqrt{\overline{\Delta}_\epsilon}, \sqrt{\overline{\tilde{u}}}, \sqrt{\overline{\tilde{\theta}}}\right]^T$, and $\overline{\Sigma}$ satisfies $\|\hat{\Sigma}(t)\| \leq \overline{\Sigma},\ \forall t \geq 0$. Since $t \mapsto x(t), t \mapsto \hat{u}(t), t \mapsto u(t)$ and $t \mapsto \hat{\theta}(t)$ are bounded by Assumption 4, the bound $\overline{\Sigma}$ can be selected independent of t_i and the specific trajectories of x, u, $\hat{u}$, and $\hat{\theta}$ currently stored in the history stack. Using (40)

and (41), [45, Theorem 4.19] can be invoked to conclude that (38) is input-to-state stable with state $\tilde{W}$ and input μ.

If Algorithm 1 is implemented and if the signal $(\hat{x}, u)$ and the sequence $(x_i, \hat{u}_i)$ are FI, there exists a time instance T_s, such that for all $t \geq T_s$, the history stacks $\mathcal{H}^u(t)$ and $\mathcal{H}^{IRL}(t)$ remain unchanged. As a result, using Exercise 4.58 from [45], the ultimate bound on $\tilde{W}$ can be expressed as

$$\limsup_{t\to\infty} \|\tilde{W}(t)\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}} \left(\frac{4\alpha\overline{\Sigma} \max\{1, \overline{\Delta}_u, \overline{\Delta}_\theta}{\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta} \right) \overline{\Delta}_\epsilon + \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}} \left(\frac{4\alpha\overline{\Sigma} \max\{1, \overline{\Delta}_u, \overline{\Delta}_\theta}{\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta} \right) \left(\overline{\tilde{u}}(T_s) + \overline{\tilde{\theta}}(T_s) \right),$$

where $\overline{\tilde{u}}(T_s)$ is an upper bound on the control estimation errors corresponding to estimates stored in the history stack $\mathcal{H}^{IRL}(t)$ for all $t \geq T_s$.

If $(\hat{x}, u)$ and $(x_i, \hat{u}_i)$ are PI, then the ultimate bound on $\tilde{W}$ reduces to

$$\limsup_{t\to\infty} \|\tilde{W}(t)\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}} \left(\frac{4\alpha\overline{\Sigma} \max\{1, \overline{\Delta}_u, \overline{\Delta}_\theta}{\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta} \right) \overline{\Delta}_\epsilon + \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}} \left(\frac{4\alpha\overline{\Sigma} \max\{1, \overline{\Delta}_u, \overline{\Delta}_\theta}{\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta} \right) \left(\overline{\gamma}_u + \overline{\Theta} \right). \qquad \square$$

Similar to Section 5.4, the following corollary, immediate from Theorem 3, states the ideal case where the basis are exact and the state and the parameters are estimated exactly.

Corollary 2 *Under the hypothesis of Theorem 3, if* $\overline{\Theta}$, $\overline{X}$, ϵ_Q, ϵ_V, *and* ϵ_u *are zero and the signal* $(\hat{x}, u)$ *and the sequence* $(x_i, \hat{u}_i)$ *are PI, then* $\lim_{t\to\infty} = \|\tilde{W}(t)\|$.

Remark 4 *The rate of convergence of the approximation errors to the ultimate bound (in the case of Theorems 1, 2, and 3) or zero (in the case of Corollary 1, Remark 3, and Corollary 2) depend on the minimum eigenvalue,* $\underline{\sigma}$, *and how fast the state estimation errors,* $\tilde{x}$, *and parameter estimation errors,* $\tilde{\theta}$, *decay to their respective ultimate bounds. Assumption 4 does not impose any convergence rate constraints on the state and parameter estimator for ease of exposition. However, given the convergence rates of the state and parameter estimation errors, the minimum eigenvalue of the regressor, and a desired neighborhood of the ultimate bound, the time required for the reward function estimation errors to enter the said neighborhood can be estimated, albeit conservatively, using Lyapunov-based techniques such as [45, Theorem 4.18].*

7 Simulation

In the following, the first simulation demonstrates the IRL method detailed in Section 5 utilizing the measured data directly for reward function estimation. The second simulation demonstrates the feedback-driven IRL method detailed in Section 6 to estimate the reward function when the trajectories of the system are not exciting enough to directly estimate the reward function.

7.1 Direct MBIRL

To validate the performance of direct MBIRL, a nonlinear optimal control problem is selected with a known value function. The simultaneous state and parameter estimator developed in [35] is used to satisfy the conditions of Assumption 4. The agent under observation has the nonlinear dynamics

$$\begin{aligned} \dot{x}_{(1)} &= x_{(2)}, \\ \dot{x}_{(2)} &= \theta_{(1)} x_{(1)} \left(\frac{\pi}{2} + \tan^{-1}(5x_{(1)}) \right) + \frac{\theta_{(2)} x_{(1)}^2}{1 + 25x_{(1)}^2} \\ &\quad + \theta_{(3)} x_{(2)} + 3u, \end{aligned} \tag{42}$$

where the parameters $\theta_{(1)}$, $\theta_{(2)}$, and $\theta_{(3)}$ are assumed to be unknown. The ideal values of these parameters are selected to be $\theta_{(1)} = -1$, $\theta_{(2)} = -\frac{5}{2}$, and $\theta_{(3)} = 4$.

The agent attempts to minimize the cost function in (2) with $Q(x) = x_{(2)}^2$ and $R = 1$, resulting in basis functions $\sigma_Q(x) = \begin{bmatrix} x_{(1)}^2, & x_{(2)}^2 \end{bmatrix}^T$ and ideal weights $W_Q = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. The observed output and control trajectories are used to estimate the system state and the unknown parameters in the dynamics, the optimal value function, and the reward function.

The closed form optimal policy is

$$u^*(x) = -\frac{1}{2} R^{-1} \left(\nabla_u f(x) \right)^T \left(\nabla_x V(x) \right)^T = -3x_{(2)},$$

with the corresponding optimal value function

$$V^*(x) = x_{(1)}^2 \left(W_{V(1)} + W_{V(2)} \tan^{-1}(5x_{(1)}) \right) + W_{V(3)} x_{(2)}^2,$$

resulting in the ideal value function weights $W_{V(1)} = \frac{\pi}{2}$, $W_{V(2)} = 1$, and $W_{V(3)} = 1$. It is assumed that the controller that the agent under observation is utilizing is a combination of the optimal controller and a known exciting controller, that is,

$$u(t) = -3x_{(2)}(t) + 9\cos(3t) + 6\cos(2t) + 3\cos(t) + 15\cos(5t).$$

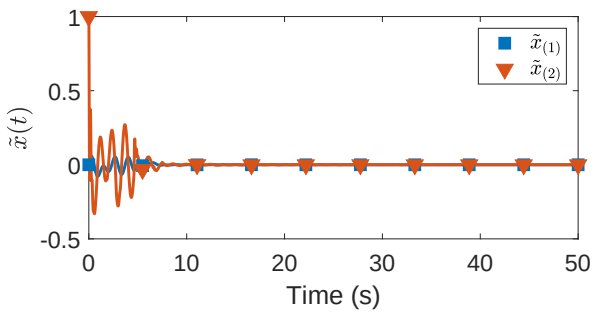

Fig. 3. State estimation errors for the system in (42).

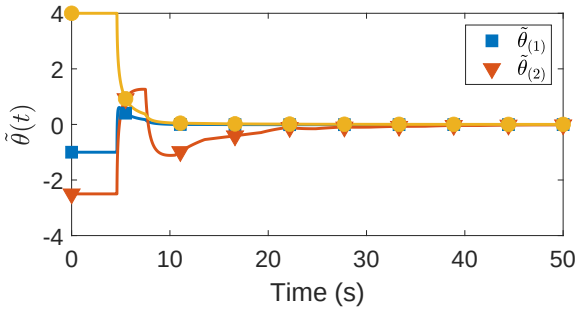

Fig. 4. Parameter estimation errors for the uncertain dynamics in (42).

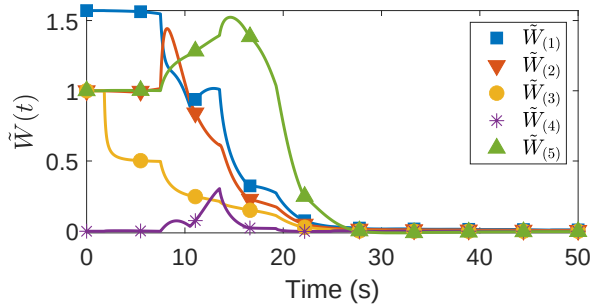

Fig. 5. Reward and value function weight estimation errors using direct MBIRL in Section 5 for the optimal control problem in (2) with $Q(x) = x_2^2$ and $R = 1$.

The history stack, $\mathcal{H}^{IRL}$ is initialized to be zero [5]. Data are added to the history stack using a minimum eigenvalue maximization algorithm. A time-based purging method is utilized with $\tau = 0.1$. Trajectories of the system and the adaptive update laws are simulated using the MathWorks® MATLAB® Simulink® implementation of the fourth order Runge-Kutta method with adaptive step size, capped to a maximum of 0.01s. The weight estimates are initialized to be zero. The learning gains are selected, through trial and error, as $N = 150$, $\alpha = {}^{0.01}/_{N}$, $\beta = 0.6$, and $\Gamma(0) = 0.001\text{diag}([1, 1, 1, 1, 0.1])$.

[5] Full rank initialization of the history stacks is a sufficient, but not necessary, condition for the analysis in Sections 6.2 and 6.5.

Figs. 3 - 5 show the performance of the developed MBIRL method. As seen in Figs. 3 and 4, the uncertain parameters and system state estimates converge to the origin. As seen in Fig. 5, the MBIRL approach is able to estimate the ideal values of the reward and value functions online.

7.2 Feedback-Driven MBIRL

In the second simulation, the unknown reward and value function weights are estimated using feedback-driven MBIRL in the case where direct MBIRL using the measured data results in large reward function estimation errors. To demonstrate the performance of feedback-driven IRL, a linear optimal trajectory tracking problem with a known value function is designed using the method developed in [49, 50]. The state and parameter estimator developed in [34] is used to satisfy the conditions of Assumption 4.

Consider an agent modeled as a linear time-invariant system $\dot{x} = Ax + Bu$, where $x \in \mathbb{R}^2$, $u \in \mathbb{R}^n$,

$$A = \begin{bmatrix} 0 & 1 \\ \theta_{(1)} & \theta_{(2)} \end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{43}$$

The parameters $\theta_{(1)}$ and $\theta_{(2)}$ are assumed to be unknown, with ideal values $\theta_{(1)} = -0.5$ and $\theta_{(2)} = -0.5$.

The trajectory the agent is attempting to follow is generated from the linear system $\dot{x} = A_d x$, where

$$A_d = \begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix}.$$

Since the agent under observation is attempting to follow a desired trajectory, the optimal control signal will likely be non-zero almost everywhere, resulting in an infinite cost. Following [49], to avoid infinite costs, it is assumed that the agent under observation solves an optimal control problem formulated to penalize an auxiliary controller, $\mu = u - u_d$, which converges to zero as the agent's controller u converges to the desired steady state controller u_d.

The error dynamics are given by

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ \theta_{(1)} & \theta_{(2)} \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu, \tag{44}$$

and the agent under observation minimizes the performance index

$$J(e_0, \mu(\cdot)) = \int_0^\infty e^T(t) Q e(t) + R\mu(t)^2 \, \mathrm{d}t, \tag{45}$$

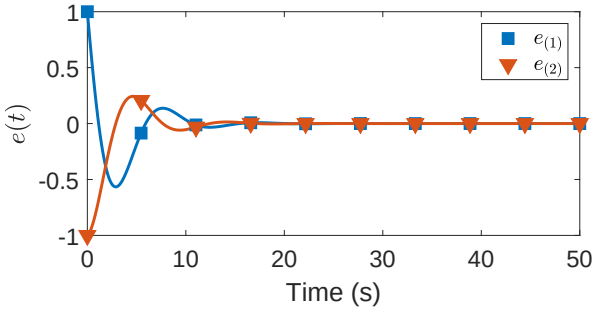

Fig. 6. Trajectory tracking error corresponding to the optimal control problem in (45).

where $t \mapsto e(t)$ denotes the solution of the error system in (44) under the controller $\mu(\cdot)$, $Q = \begin{bmatrix} 1.1 & 0 \\ 0 & 3 \end{bmatrix}$, and $R = 50$. The diagonal elements of the state penalty matrix, Q, are assumed to be unknown, resulting in the ideal reward function weights $W_Q = \begin{bmatrix} 1.1, & 3 \end{bmatrix}^T$.

The steady state controller needed to track the desired trajectory is given by $u_d = \begin{bmatrix} 0 & 1 \end{bmatrix} (A_d - A)x_d = [1.5, \ -0.5]\, x_d$. Since the learner only has access to measurements of x, x_d, and u, the steady-state controller, u_d, needs to be approximated by the learner using estimates of θ.

The optimal value function to be estimated is $V^* = W_{V(1)}e_{(1)}^2 + W_{V(2)}e_{(2)}^2 + W_{V(3)}e_{(1)}e_{(2)}$, where the ideal weights are $W_{V(1)} = 3.00, W_{V(2)} = 4.71$, and $W_{V(3)} = 2.15$. The optimal policy to be estimated is $\mu = -W_{u(1)}e_{(1)} - W_{u(2)}e_{(2)}$, where the ideal weights are $W_{u(1)} = 0.0215$ and $W_{u(2)} = 0.0942$. To generate an estimate of the optimal policy μ, the update law in (24) is used with the estimated state $\hat{x}$ and the known desired state x_d, at current time t, concatenated into $\hat{\Sigma}_\sigma$. The estimated policy is then queried with random error values e_i in the set $[-5, 5]$, which produce estimates of the optimal control command, $\hat{\mu}_i$, in response to e_i. The pairs $(e_i, \hat{\mu}_i)$ are then collected in a history stack $\mathcal{H}^{IRL}$, and utilized to implement the feedback-driven MBIRL method in Section 6.

The history stacks, $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$, are initialized to be zero. Data are added to the history stack using a minimum eigenvalue maximization algorithm. A time-based purging method is utilized with dwell time $\tau = 1$s for estimation of the value function and the reward function, and $\tau = 0.1$s for estimation of the optimal policy. The weight estimates are initialized to be zero. The learning gains are selected, through trial and error, as $\beta = 0.2$, $N = 10$, $\alpha = {}^{0.1}/_{N}$, $\beta_u = 1$, $\alpha_u = 1$, $M = 40$, $\Gamma(0) = \mathrm{I}$, and $\Gamma_u(0) - 0.002\mathrm{I}$.

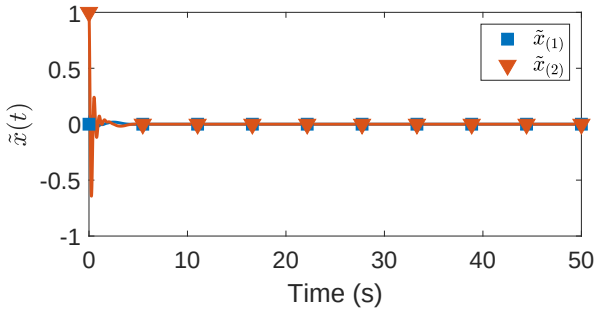

Fig. 7. State estimation errors for the system in (43).

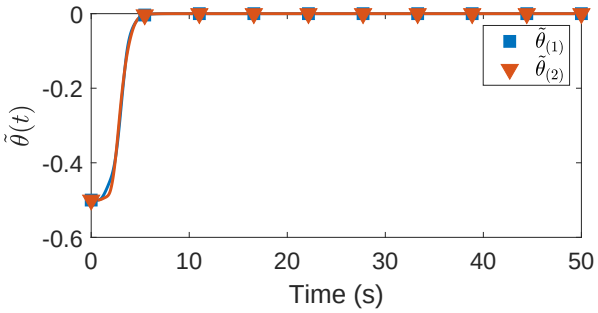

Fig. 8. Parameter estimation errors for the uncertain dynamics in (43).

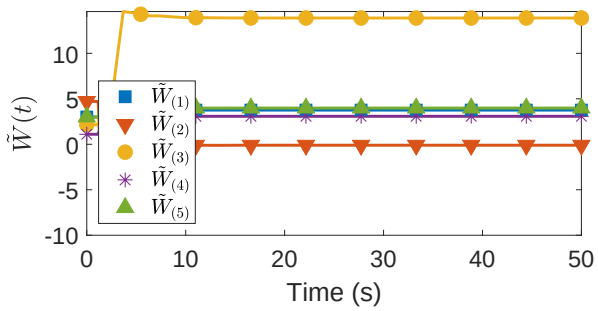

Fig. 9. Reward and value function weight estimation errors using direct MBIRL in Section 5 for the optimal control problem in (45).

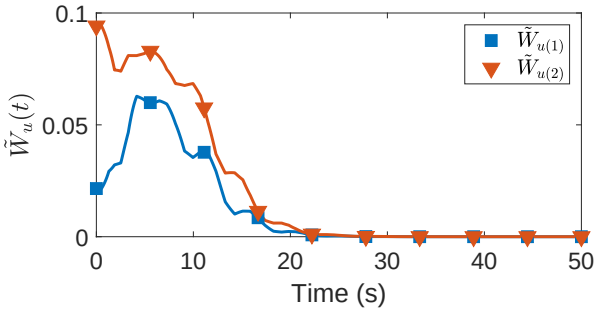

Fig. 10. Control weight estimation errors for the auxiliary controller μ for the optimal control problem in (45).

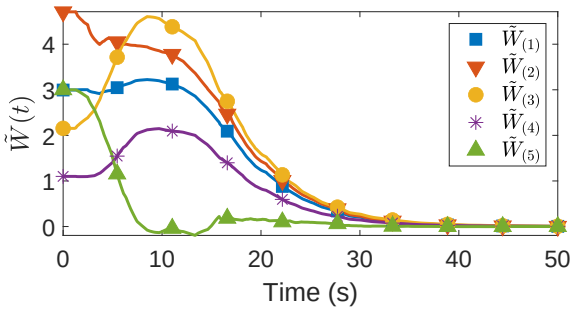

Fig. 11. Reward and value function weight estimation errors using feedback-driven MBIRL in Section 6 for the optimal control problem in (45).

The tracking errors and corresponding auxiliary controller trajectories converge to the origin within 20 seconds (see Fig. 6). It is clear from Fig. 9 that the direct MBIRL method Section 5, implemented using these trajectories, results in large estimation errors. The errors are attributed, heuristically, to lack of sufficient information regarding the reward function in the recorded data. The lack of information can be compensated for in feedback-driven MBIRL via artificially synthesized expert response $\hat{\mu}_i$ to randomly selected tracking errors e_i. Since the estimation errors $\tilde{W}_u$ corresponding to the unknown weights in the optimal policy go to zero (see Fig. 10), the artificially synthesized demonstrator responses asymptotically approach true expert responses, facilitating accurate reward function estimation, as observed in Fig. 11.

8 Conclusion

In this paper, an online MBIRL method is developed that facilitate reward function estimation utilizing a single demonstration. Since a large majority of optimal control problems are aimed at driving a state to a set-point or an error signal to zero, single demonstrations may not provide sufficient excitation to directly estimate the reward function from only measured data. Therefore, a second method is developed that utilizes an estimated policy to synthesize additional data that mimics the control policy of the agent under observation.

As stated in Footnote 3, the finite informativity requirement implicitly restricts the results in this paper to IRL problems that admit unique solutions up to a scaling factor. A detailed examination of IRL problems with solutions that are nonunique beyond a scaling factor is outside the scope of this paper. However, numerical experiments indicate that, when applied to such problems, the developed method converges to different equivalent solutions of the IRL problem depending on the initial guess of $\hat{W}$, where two reward functions are called equivalent if the corresponding optimal policies are identical.

References

[1] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn.* Morgan Kaufmann, 2000, pp. 663–670.

[2] S. Russell, "Learning agents for uncertain environments (extended abstract)," in *Proc. Conf. Comput. Learn. Theory*, 1998.

[3] R. E. Kalman, "When is a linear control system optimal?" *J. Basic Eng.*, vol. 86, no. 1, pp. 51–60, 1964.

[4] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2004.

[5] P. Abbeel and Y. Ng, Andrew, "Exploration and apprenticeship learning in reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2005.

[6] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proc. Int. Conf. Mach. Learn.*, 2006.

[7] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI Conf. Artif. Intel.*, 2008, pp. 1433–1438.

[8] Z. Zhou, M. Bloem, and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," *IEEE Trans. Autom. Control*, vol. 63, no. 9, pp. 2787–2802, 2018.

[9] S. Levine, Z. Popovic, and V. Koltun, "Feature construction for inverse reinforcement learning," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1342–1350.

[10] G. Neu and C. Szepesvari, "Apprenticeship learning using inverse reinforcement learning and gradient methods," in *Proc. Anu. Conf. Uncertain. Artif. Intell.* Corvallis, Oregon: AUAI Press, 2007, pp. 295–302.

[11] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 1449–1456.

[12] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," arXiv:1507.04888, 2015.

[13] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 19–27.

[14] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Human behavior modeling with maximum entropy inverse optimal control." in *AAAI Hum. Behav. Model.*, 2009, p. 92.

[15] R. Kamalapurkar, "Linear inverse reinforcement learning in continuous time and space," in *Proc. Am. Control Conf.*, Milwaukee, WI, USA, Jun. 2018, pp. 1683–1688.

[16] T. Molloy, J. Ford, and T. Perez, "Online inverse optimal control on infinite horizons," in *Proc. IEEE Conf. Decis. Control.* IEEE, 2018, pp. 1663–1668.

[17] R. V. Self, M. Harlan, and R. Kamalapurkar, "Online inverse reinforcement learning for nonlinear systems," in *Proc. IEEE Conf. Control Technol. Appl.*, Hong Kong, China, Aug. 2019, pp. 296–301.

[18] R. V. Self, M. Abudia, and R. Kamalapurkar, "Online inverse reinforcement learning for systems with disturbances," in *Proc. Am. Control Conf.*, Jul. 2020, pp. 1118–1123.

[19] N. Rhinehart and K. M. Kitani, "Online semantic activity forecasting with darko," arXiv:1612.07796, 2016.

[20] ——, "First-person activity forecasting with online inverse reinforcement learning," in *Proc. IEEE Conf. Comput. Vis.*, 2017, pp. 3696–3705.

[21] N. Rhinehart and K. Kitani, "First-person activity forecasting from video with online inverse reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 304–317, 2018.

[22] J. Mendez, S. Shivkumar, and E. Eaton, "Lifelong inverse reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 4502–4513.

[23] S. Arora, P. Doshi, and B. Banerjee, "Online inverse reinforcement learning under occlusion," in *Proc. Conf. Auton. Agents MultiAgent Syst.* International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1170–1178.

[24] ——, "A framework and method for online inverse reinforcement learning," arXiv:1805.07871, 2018.

[25] Z. Jin, H. Qian, S. Chen, and M. Zhu, "Convergence analysis of an incremental approach to online inverse reinforcement learning," *J. Zhejiang Univ. - Sci. C*, vol. 12, no. 1, pp. 17–24, 2011.

[26] K. Li and J. Burdick, "Online inverse reinforcement learning via bellman gradient iteration," arXiv:1707.09393, 2017.

[27] W. Xue, P. Kolaric, J. Fan, B. Lian, T. Chai, and F. L. Lewis, "Inverse reinforcement learning in tracking control based on inverse optimal control," *IEEE Trans. Cybern.*, 2021.

[28] B. Lian, W. Xue, F. L. Lewis, and T. Chai, "Online inverse reinforcement learning for nonlinear systems with adversarial attacks," *Int. J. Robust Nonlinear Control*, 2021.

[29] T. L. Molloy, J. J. Ford, and T. Perez, "Online inverse optimal control for control-constrained discrete-time systems on finite and infinite horizons," *Automatica*, vol. 120, 2020.

[30] ——, "Finite-horizon inverse optimal control for discrete-time nonlinear systems," *Automatica*, vol. 87, pp. 442–446, 2018.

[31] R. V. Self, S. M. N. Mahmud, K. Hareland, and R. Kamalapurkar, "Online inverse reinforcement learning with limited data," in *Proc. IEEE Conf. Decis. Control*, Jeju Island, Republic of Korea, Dec. 2020, pp. 603–608.

[32] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction.* Princeton University Press, 2012.

[33] F. Sauvigny, *Partial Differential Equations 1.* Springer, 2012.

[34] R. Kamalapurkar, "Online output-feedback parameter and state estimation for second order linear systems," in *Proc. Am. Control Conf.*, Seattle, WA, USA, May 2017, pp. 5672–5677.

[35] ——, "Simultaneous state and parameter estimation for second-order nonlinear systems," in *Proc. IEEE Conf. Decis. Control*, Melbourne, VIC, Australia, Dec. 2017, pp. 2164–2169.

[36] R. Morino and P. Tomie, "Adaptive observers with arbitrary exponential rate of convergence for nonlinear systems via filtered transformations," *IEEE Trans. Autom. Control*, vol. 40, pp. 1300–1304, 1995.

[37] P. Li, F. Boem, G. Pin, and T. Parisini, "Kernel-based simultaneous parameter-state estimation for continuous-time systems," *IEEE Trans. Auto. Control*, vol. 65, no. 7, pp. 3053–3059, 2019.

[38] A. Morgan and K. S. Narendra, "On the uniform asymptotic stability of certain linear nonautonomous differential equations," *SIAM J. Control Optim.*, vol. 15, no. 1, pp. 5–24, 1977.

[39] S. T. Glad and L. Ljung, "Model structure identifiability and persistence of excitation," in *Proc. IEEE Conf. Decis. Control*, 1990, pp. 3236–3240.

[40] S. S. Sastry and M. Bodson, *Adaptive control: stability, convergence, and robustness.* Upper Saddle River, NJ: Prentice-Hall, 1989.

[41] P. Ioannou and J. Sun, *Robust adaptive control.* Prentice Hall, 1996.

[42] G. Chowdhary and E. Johnson, "A singular value maximizing data recording algorithm for concurrent learning," in *Proc. Am. Control Conf.*, 2011, pp. 3547–3552.

[43] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.

[44] S. Kersting and M. Buss, "Concurrent learning adaptive identification of piecewise affine systems," in *Proc. IEEE Conf. Decis. Control*, Dec. 2014, pp. 3930–3935.

[45] H. K. Khalil, *Nonlinear systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[46] V. Adetola and M. Guay, "Finite-time parameter estimation in adaptive control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 807–811, 2008.

[47] J. Zhi, X. Dong, Y. Chen, Z. Liu, and C. Shi, "Robust adaptive finite time parameter estimation with relaxed persistence of excitation," in *Asian Control Conf.*, 2017, pp. 1384–1388.

[48] J. Na, M. N. Mahyuddin, G. Herrmann, and X. Ren, "Robust adaptive finite-time parameter estimation for linearly parameterized nonlinear systems," in *Proc. Chinese Control Conf.*, 2013, pp. 1735–1741.

[49] R. Kamalapurkar, H. T. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, Jan. 2015.

[50] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for infinite-horizon approximate optimal tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 753–758, Mar. 2017.