2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Online inverse reinforcement learning with limited data

Ryan Self, S M Nahid Mahmud, Katrine Hareland and Rushikesh Kamalapurkar

Abstract— **This paper addresses the problem of online inverse reinforcement learning for systems with limited data and uncertain dynamics. In the developed approach, the state and control trajectories are recorded online by observing an agent perform a task, and reward function estimation is performed in real-time using a novel inverse reinforcement learning approach. Parameter estimation is performed concurrently to help compensate for uncertainties in the agent's dynamics. Data insufficiency is resolved by developing a data-driven update law to estimate the optimal feedback controller. The estimated controller can then be queried to artificially create additional data to drive reward function estimation.**

I. 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 a demonstrator by monitoring its state and control trajectories. Reward function estimation is performed in the presence of modeling uncertainties for situations with limited data via inverse reinforcement learning (IRL) [1], [2].

While IRL in an *offline* setting has a rich history of literature [1]–[11], little work has been done to address IRL in an *online* setting. The limited data provided by a single demonstration is a significant challenge that has hampered the development of online IRL.

Preliminary results on online IRL are available for linear systems, in results such as [12] and [13], and for nonlinear systems, in results such as [14] and [15]. However, [12] and [14] exploit access to demonstrator's feedback policy, [13] requires exact model knowledge, and [15] exploits identical disturbances to provide sufficient excitation. The main contribution of this paper is the development of a novel method for reward function estimation for an agent in situations where estimation of the demonstrator's optimal feedback law is less data-intensive than direct estimation of its reward function.

The novelty in the technique developed in this paper is a recursive model-based IRL approach which facilitates the use of off-trajectory state-action pairs. A majority of IRL methods are trajectory-driven and model-free. As a result, the 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, arbitrary state-action pairs can be used for IRL as long as the action is the optimal action for that state. In [12] and [14], the off-trajectory state-action pairs are generated under the assumption that the learner either knows the demonstrator's optimal feedback law or can query the demonstrator to find out what the optimal action would be at a given off-trajectory state. In this paper, we develop a novel IRL approach that relaxes the aforementioned assumption.

The key idea in this paper is to estimate the optimal feedback controller of the agent online, and use that estimate to artificially create off-trajectory data to drive reward function estimation. In the authors' previous work [14], reward function estimation is performed directly using the agent's observed trajectories. Instead, in this paper, the trajectory information is used to estimate the optimal feedback controller. This controller is parameterized as a neural network and estimated using a concurrent learning update law. The estimated controller is simultaneously queried to create off-trajectory data which is then used for reward function estimation via IRL. Since the optimal controller is estimated using a neural network, the controller can be estimated independent of the modeling uncertainty. In the developed approach, a parameter estimator and two update laws for estimation of the optimal feedback controller and reward function are utilized simultaneously to achieve convergence of the unknown reward function weights to a neighborhood of their true values.

The paper is organized as follows: Section II explains the notation used throughout the paper. Section III details the problem formulation. Section IV shows how to estimate the optimal controller. Section V explains the IRL algorithm. Section VI shows a simulation example and Section VII concludes the paper.

II. Notation

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, ∞). If $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$, then $[a; b]$ denotes the concatenated vector $\begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^{m+n}$. The notations I_n and 0_n denote the $n \times n$ identity matrix and the zero element of $\mathbb{R}^n$, respectively. Whenever it is clear from the context, the subscript n is suppressed.

III. Problem Formulation

Consider an agent with the dynamics

$$\dot{x} = f(x, u), \tag{1}$$

The authors are with the School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK, USA. {rself,nahid.mahmud,katrine.hareland, rushikesh.kamalapurkar}@okstate.edu. This research was supported, in part, by the National Science Foundation (NSF) under award number 1925147. 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.

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

where $x : \mathbb{R}_{\geq T_0} \to \mathbb{R}^n$ is the state, $u : \mathbb{R}_{\geq T_0} \to \mathbb{R}^m$ is the control, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a continuously differentiable function, and $T_0 \in \mathbb{R}_{\geq 0}$ denotes the initial time.

Assume that the agent under observation is using a policy which minimizes the performance index

$$J(x_0, u(\cdot)) = \int_{T_0}^{\infty} r(x(t; x_0, u(\cdot)), u(t))\, \mathrm{d}t, \tag{2}$$

where $x(\cdot; x_0, u(\cdot))$ is the trajectory of the agent generated by the optimal control signal $u(\cdot)$, starting from the initial condition x_0 and initial time T_0. The main objective of the paper is to estimate the unknown reward function, r, using state-action pairs.

The following assumptions are used throughout the rest of this paper.

Assumption 1. *The unknown reward function r is quadratic in the control, i.e.,*

$$r(x, u) = Q(x) + u^T R u, \tag{3}$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix, such that $R = \mathrm{diag}([r_1, \cdots, r_m])$.

Assumption 2. *The state and control trajectories are bounded such that $x(t) \in \mathcal{X}$, $u(t) \in \mathcal{U}$ for some compact sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$.*

The continuous function Q can be represented using $L \in \mathbb{N}$ basis functions as $Q(x) = (W_Q^*)^T \sigma_Q(x) + \epsilon_Q(x)$, where $W_Q^* \coloneqq [q_1, \cdots, q_L]^T$ are ideal reward function weights, $\sigma_Q : \mathbb{R}^n \to \mathbb{R}^L$ are known continuously differentiable features, and $\epsilon_Q : \mathbb{R}^n \to \mathbb{R}$ is the approximation error. Given any constant $\bar{\epsilon}_Q \in \mathbb{R}_{>0}$, there exists $L \in \mathbb{N}$ such that ϵ_Q satisfies $\sup_{x \in \mathcal{X}} \|\epsilon_Q(x)\| < \bar{\epsilon}_Q$, and $\sup_{x \in \mathcal{X}} \|\nabla_x \epsilon_Q(x)\| < \bar{\epsilon}_Q$ [16], [17].

Assumption 3. *The dynamics for the agent are affine in control.*

The dynamics can be represented using $P \in \mathbb{N}$ basis functions as

$$\dot{x} = f^o(x, u) + \theta^T \sigma(x, u) + \epsilon(x, u), \tag{4}$$

where $f^o : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the continuously differentiable nominal dynamics, $\theta^T \sigma$ is a parameterized estimate of the uncertain part of the dynamics, where $\theta \in \mathbb{R}^{p \times n}$ is a matrix of unknown constant parameters and $\sigma : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ are known continuously differentiable features, and $\epsilon : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the function approximation error. Given any constant $\bar{\epsilon} \in \mathbb{R}_{>0}$, there exist $p \in \mathbb{N}$ such that $\sup_{(x,u) \in (\mathcal{X} \times \mathcal{U})} \|\epsilon(x, u)\| < \bar{\epsilon}$, and $\sup_{(x,u) \in (\mathcal{X} \times \mathcal{U})} \|\nabla \epsilon(x, u)\| < \bar{\epsilon}$.

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 (HJB) [18] equation

$$H\left(x(t), \left([\nabla_x V^*](x(t))\right)^T, u(t)\right) = 0, \forall t \in \mathbb{R}_{\geq T_0}, \tag{5}$$

where the unknown optimal value function is $V^* : \mathbb{R}^n \to \mathbb{R}$ and $H : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the Hamiltonian, defined as $H(x, p, u) \coloneqq p^T f(x, u) + r(x, u)$. The goal of IRL is to estimate the reward function, r.

To aid in the estimation of the reward function, let $\hat{V} : \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}$, $\left(x, \hat{W}_V\right) \mapsto \hat{W}_V^T \sigma_V(x)$ be a parameterized estimate of the optimal value function V^*, where $\hat{W}_V \in \mathbb{R}^P$ are the estimates of the ideal value function weights W_V^* and $\sigma_V : \mathbb{R}^n \to \mathbb{R}^P$ are known continuously differentiable features. Let $\epsilon_V : \mathbb{R}^n \to \mathbb{R}$, defined as $\epsilon_V(x) = V^*(x) - (W_V^*)^T \sigma_V(x)$, be the resulting approximation error. Given any constant $\bar{\epsilon}_V \in \mathbb{R}_{>0}$, there exists $P \in \mathbb{N}$ such that ϵ_V satisfies $\sup_{x \in \mathcal{X}} \|\epsilon_V(x)\| < \bar{\epsilon}_V$, and $\sup_{x \in \mathcal{X}} \|\nabla_x \epsilon_V(x)\| < \bar{\epsilon}_V$. Using $\hat{W}_V$, $\hat{W}_Q$, and $\hat{W}_R$, which are the estimates of W_V^*, W_Q^*, and $W_R^* \coloneqq [r_1, \cdots, r_m]^T$, respectively, in (5), the inverse Bellman error $\delta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+m} \to \mathbb{R}$ is obtained as

$$\begin{aligned}\delta\left(x, u, \hat{W}\right) =& \hat{W}_V^T \Big([\nabla_x \sigma_V](x)\Big) f(x, u) + \hat{W}_Q^T \sigma_Q(x) \\ &+ \hat{W}_R^T \sigma_u(u),\end{aligned} \tag{6}$$

where $\sigma_u(u) \coloneqq \left[u_1^2, \cdots, u_m^2\right]$.

For brevity of presentation, it is assumed that a parameter estimator that satisfies the following properties is available. For examples of such parameter estimators, see [14], [19].

Assumption 4. *[20, Assumption 2] A compact set $\Theta \subset \mathbb{R}^p$ such that $\theta \in \Theta$ is known a priori. The estimate $\hat{\theta} : \mathbb{R}_{\geq T_0} \to \mathbb{R}^p$ are updated based on a switched update law of the form*

$$\dot{\hat{\theta}} = f_{\theta_s}(\hat{\theta}(t), t),$$

$\hat{\theta}(T_0) = \hat{\theta}_0 \in \Theta$, where $s \in \mathbb{N}$ denotes the switching index and $\{f_{\theta_s} : \mathbb{R}^p \times \mathbb{R}_{\geq T_0} \to \mathbb{R}^p\}_{s \in \mathbb{N}}$ denotes the family of continuously differentiable functions. The dynamics of the parameter estimation error $\tilde{\theta} : \mathbb{R}_{\geq T_0} \to \mathbb{R}^p$, defined as $\tilde{\theta}(t) \coloneqq \theta - \hat{\theta}(t)$, can be expressed as $\dot{\tilde{\theta}}(t) = f_{\theta_s}\left(\theta - \tilde{\theta}(t), t\right)$. Furthermore, there exists a continuously differentiable function $V_\theta : \mathbb{R}^p \times \mathbb{R}_{\geq T_0} \to \mathbb{R}_{\geq 0}$ that satisfies

$$\underline{\nu}_\theta\left(\left\|\tilde{\theta}\right\|\right) \leq V_\theta\left(\tilde{\theta}, t\right) \leq \overline{\nu}_\theta\left(\left\|\tilde{\theta}\right\|\right),$$

and

$$\begin{aligned}\left(\left[\nabla_{\tilde{\theta}} V_\theta\right]\left(\tilde{\theta}, t\right)\right)\left(-f_{\theta_s}\left(\theta - \tilde{\theta}, t\right)\right) + \frac{\partial V_\theta\left(\tilde{\theta}, t\right)}{\partial t} \\ \leq -K\left\|\tilde{\theta}\right\|^2 + D\left\|\tilde{\theta}\right\|,\end{aligned}$$

for all $s \in \mathbb{N}, t \in \mathbb{R}_{\geq T_0}$, and $\tilde{\theta} \in \mathbb{R}^p$, where $\underline{\nu}_\theta, \overline{\nu}_\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions, $K \in \mathbb{R}_{>0}$ is an adjustable parameter, and $D \in \mathbb{R}_{>0}$ is a positive constant.

Utilizing the parameter estimates $\hat{\theta}$, the inverse Bellman error in (6) can be approximated as

$$\begin{aligned}\delta'\left(x, u, \hat{W}, \hat{\theta}\right) =& \hat{W}_V^T \Big([\nabla_x \sigma_V](x)\Big) \hat{Y}(x, u, \hat{\theta}) + \hat{W}_Q^T \sigma_Q(x) \\ &+ \hat{W}_R^T \sigma_u(u),\end{aligned} \tag{7}$$

where $\hat{Y}(x,u,\hat{\theta}) := f^o(x,u)+\hat{\theta}^T\sigma(x,u)$ and $\hat{\theta}$ are estimates of unknown parameters. Rearranging, (7) becomes

$$\delta'\left(x,u,\hat{W}',\hat{\theta}\right) = \left(\hat{W}'\right)^T \sigma'\left(x,u,\hat{\theta}\right), \tag{8}$$

where $\hat{W}' := \left[\hat{W}_V;\hat{W}_Q;\hat{W}_R\right]$ and $\sigma'\left(x,u,\hat{\theta}\right) := \left[\left(\left[\nabla_x\sigma_V\right](x)\right)\hat{Y}(x,u,\hat{\theta});\sigma_Q(x);\sigma_u(u)\right]$.

In the following, the parameter estimator is executed synchronously with IRL and in real-time.

IV. Optimal Policy Estimation

Since a large majority of optimal control problems 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 rendering them unable to provide usable data. More specifically, once the states converge, newer data points from the agent's trajectory will simply provide zero, or near-zero, values for both the states (or errors) and the controls. As a result, the reward function estimate may never converge. In addition, even if sufficient excitation exists to estimate the unknown reward function directly, artificially generated state-action pairs can help accelerate the estimation by providing additional data points. Motivated by the observation that knowledge of the optimal controller can be leveraged to artificially create additional data to drive IRL, this section develops a process for finding an estimate of the optimal controller.

A. Policy Estimator Design

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

$$u^*(x) = -\frac{1}{2}R^{-1}\left(\left[\nabla_u f\right](x)\right)^T\left(\left[\nabla_x V^*\right](x)\right)^T, \tag{9}$$

where $u^* := [u_1,u_2,\cdots,u_m]^T$. To facilitate estimation, u^* is represented as

$$u^*(x) = -\left(W_u^*\right)^T\sigma_u(x) + \epsilon_u(x), \tag{10}$$

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. Given any constant $\overline{\epsilon}_u \in \mathbb{R}_{>0}$, there exists $K \in \mathbb{N}$ such that ϵ_u satisfies $\sup_{x\in\mathcal{X}}\|\epsilon_u(x)\| < \overline{\epsilon}_u$, and $\sup_{x\in\mathcal{X}}\|\nabla_x\epsilon_u(x)\| < \overline{\epsilon}_u$.

Collecting state and control signals over time instances, $t_1,t_2,\cdots,t_M$, stored in a history stack, denoted as $\mathcal{H}^u$, (10) can be formulated into the matrix form

$$-\Sigma_u - \Sigma_\sigma\hat{W}_u = \Sigma_\sigma\tilde{W}_u - \Delta_u, \tag{11}$$

where $\Sigma_u := [u^T(t_1);u^T(t_2);\cdots;u^T(t_M)]$, $\Sigma_\sigma := [\sigma_u^T(x(t_1));\sigma_u^T(x(t_2));\cdots;\sigma_u^T(x(t_M))]$, and $\Delta_u := [\epsilon_u^T(x(t_1));\epsilon_u^T(x(t_2));\cdots;\epsilon_u^T(x(t_M))]$. The weight estimation error is defined as $\tilde{W}_u = W_u^* - \hat{W}_u$, where $\hat{W}_u$ is the estimate of W_u^*.

Using (11), a recursive least-squares update law to estimate the unknown weights is designed as

$$\dot{\hat{W}}_u = \alpha_u\Gamma_u\Sigma_\sigma^T\left(-\Sigma_u - \Sigma_\sigma\hat{W}_u\right), \tag{12}$$

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

$$\dot{\Gamma}_u = \beta_u\Gamma_u - \alpha_u\Gamma_u\Sigma_\sigma^T\Sigma_\sigma\Gamma_u, \tag{13}$$

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

B. Analysis

The time-varying history stack, $\mathcal{H}^u$, is called full rank, uniformly in t, if there exists a $\underline{k} > 0$ such that $\forall t \in \mathbb{R}_{\geq T_0}$,

$$0 < \underline{k} < \lambda_{\min}\left\{\Sigma_\sigma^T(t)\Sigma_\sigma(t)\right\}. \tag{14}$$

Using arguments similar to [21, 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

$$\underline{\Gamma}_u\mathrm{I}_K \leq \Gamma_u(t) \leq \overline{\Gamma}_u\mathrm{I}_K, \forall\, t \in \mathbb{R}_{\geq T_0}, \tag{15}$$

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

To facilitate the following analysis, using (11) and (12), the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}}_u = -\alpha_u\Gamma_u\Sigma_\sigma^T\left(\Sigma_\sigma\tilde{W}_u - \Delta_u\right). \tag{16}$$

Theorem 1. *If $\mathcal{H}^u$ is full rank, uniformly in t, then $t \mapsto \tilde{W}_u(t)$ is ultimately bounded.*

Proof. For brevity, the details of the proof has been omitted (see [22, Theorem 1]). □

V. Inverse Reinforcement Learning

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

A. Approximate inverse Bellman error

Consider a time instance, t_i. For each time t_i, select an arbitrary state, denoted by x_i, and let $\hat{u}_i := \hat{W}_u^T(t_i)\sigma_u(x_i)$ be the estimate of the optimal controller u_i^* at state x_i and time t_i. The approximated inverse Bellman error, when evaluated at the arbitrarily selected state and at time t_i, using the estimates of the model and the optimal controller, is given by

$$\delta''(t_i,x_i,\hat{u}_i) = \left(\hat{W}'(t_i)\right)^T\sigma'(t_i,x_i,\hat{u}_i), \tag{17}$$

where $\hat{W}'(t_i) := \left[\hat{W}_V(t_i);\hat{W}_Q(t_i);\hat{W}_R(t_i)\right]$ and

$$\sigma'(t_i,x_i,\hat{u}_i) := \Big[\Big(\left[\nabla_x\sigma_V\right](x_i)\Big)(f^o(x_i,\hat{u}_i) + \hat{\theta}^T(t_i)\sigma(x_i,\hat{u}_i));\sigma_Q(x_i);\sigma_u(\hat{u}_i)\Big].$$

Since all positive multiples of a reward function result in the same optimal controller and optimal state trajectories, given state-action pairs, the reward function can only be identified up to a scale. As a result, one of the reward function weights can be arbitrarily assigned. In the following, the first element of $\hat{W}_R$ is assumed to be known.

The approximate inverse BE in (17) can then be expressed as

$$\delta''(t_i, x_i, \hat{u}_i) = \left(\hat{W}(t_i)\right)^T \sigma''(t_i, x_i, \hat{u}_i) + r_1 \sigma_{u1}(\hat{u}_i), \quad (18)$$

where $\hat{W}(t_i) \coloneqq \left[\hat{W}_V(t_i); \hat{W}_Q(t_i); \hat{W}_R^-(t_i)\right]$, the vector $\hat{W}_R^-$ denotes $\hat{W}_R$ with the first element removed, $\sigma_{uj}(\hat{u}_i)$ denotes the jth element of the vector $\sigma_u(\hat{u}_i)$, the vector σ_u^- denotes σ_u with the first element removed, and

$$\sigma''(t_i, x_i, \hat{u}_i) \coloneqq \Big[\Big([\nabla_x \sigma_V](x_i)\Big)(f^o(x_i, \hat{u}_i) + \hat{\theta}^T(t_i)\sigma(x_i, \hat{u}_i)); \sigma_Q(x_i); \sigma_u^-(\hat{u}_i)\Big]. \quad (19)$$

The closed-form nonlinear optimal controller corresponding to the reward structure in (2) provides the relationship

$$-2Ru^*(x_i) = \Big([\nabla_u f](x_i)\Big)^T \Big([\nabla_x \sigma_V](x_i)\Big)^T W_V^* + \Big([\nabla_u f](x_i)\Big)^T \Big([\nabla_x \epsilon_V](x_i)\Big)^T. \quad (20)$$

Utilizing estimates $\hat{\theta}(t_i)$ and data pairs $(x_i, \hat{u}_i)$ in (20), subtracting $H\left(x_i, \Big([\nabla_x V](x_i)\Big), u^*(x_i)\right)$ from (18), evaluating (18) and (20) at time instances $\{t_i\}_{i=1}^N$, and stacking the results in a matrix form, we get

$$-\hat{\Sigma}\hat{W} - \hat{\Sigma}_{u1} = \hat{\Sigma}\tilde{W} - \Delta. \quad (21)$$

In (21), the weight estimation error is defined as $\tilde{W} = W^* - \hat{W}$, $\hat{W}$ is the estimate of W^*,

$$\hat{\Sigma} \coloneqq \left[\sigma^T(t_1, x_1, \hat{u}_1); \cdots; \sigma^T(t_N, x_N, \hat{u}_N)\right],$$
$$\hat{\Sigma}_{u1} \coloneqq \left[\sigma'_{u1}(\hat{u}_1); \cdots; \sigma'_{u1}(\hat{u}_N)\right], \text{ and}$$
$$\Delta \coloneqq \left[\Delta_\delta(t_1); \Delta_m(t_1); \cdots; \Delta_\delta(t_N); \Delta_m(t_N)\right],$$

where

$$\sigma'_{u1}(\hat{u}_i) \coloneqq \left[r_1 \sigma_{u1}(\hat{u}_{1i}); 2r_1 \hat{u}_{1i}; 0_{(m-1)\times 1}\right],$$
$$\sigma \coloneqq \left[\sigma''\left[\begin{bmatrix} G \\ 0_{m\times L}, \begin{bmatrix} 0_{1\times m-1} \\ 2\text{diag}\left([\hat{u}_{2i}, \cdots, \hat{u}_{mi}]\right)\end{bmatrix}\end{bmatrix}^T\right]\right],$$
$$G \coloneqq \Big([\nabla_x \sigma_V](x_i)\Big)\Big(\Big([\nabla_u f^o](x_i)\Big) + \hat{\theta}^T(t_i)\Big([\nabla_u \sigma](x_i)\Big)\Big),$$
$$\Delta_\delta(t_i) \coloneqq 2R\tilde{u}_i + \Big([\nabla_u \sigma](x_i)\Big)^T \tilde{\theta}(t_i)\Big([\nabla_u \sigma_V](x_i)\Big)^T W_V^* + \Big(\Big([\nabla_u f^o](x_i)\Big) + \theta^T(t_i)\Big([\nabla_u \sigma](x_i)\Big)\Big)^T \Big([\nabla_x \epsilon_V](x_i)\Big)^T + \Big([\nabla_u \epsilon](x_i, u_i^*)\Big)\Big([\nabla_x \sigma_V](x_i)\Big) W_V^*,$$
$$\Delta_m(t_i) \coloneqq \Big(\sigma_u(u_i^*) - \sigma_u(\hat{u}_i)\Big)^T W_R^* + \epsilon_V(x_i) + \epsilon_Q(x_i) + \Big(f^o(x_i, u_i^*) - f^o(x_i, \hat{u}_i)\Big)^T \Big([\nabla_x \sigma_V](x_i)\Big)^T W_V^* + \Big(\theta^T(\sigma(x_i, u_i^*) - \sigma(x_i, \hat{u}_i))\Big)^T \Big([\nabla_x \sigma_V](x_i)\Big)^T W_V^* + \Big(\tilde{\theta}^T(t_i)\sigma(x_i, \hat{u}_i) + \epsilon(x_i, u_i^*)\Big)^T \Big([\nabla_x \sigma_V](x_i)\Big)^T W_V^*,$$

and $\hat{u}_{ji}$ is the jth element of $\hat{u}_i$.

A history stack, denoted as $\mathcal{H}^{IRL}$, is a set of ordered pairs of parameter estimates, $\hat{\theta}(t_i)$, and data pairs, $(x_i, \hat{u}_i)$, collected over time instances $t_1, t_2, \ldots, t_N$ into matrices $\left(\hat{\Sigma}, \hat{\Sigma}_{u1}\right)$.

Due to the fact that the residual errors Δ can be decreased by improving the quality of the control and the parameter estimates stored in $\mathcal{H}^{IRL}$, a purging technique is incorporated in the following to remove poor estimates $\hat{u}$ and $\hat{\theta}$ from $\mathcal{H}^{IRL}$. During the transient phase of the control and parameter estimators, the estimates $\hat{u}$ and $\hat{\theta}$ are likely to be less accurate and the resulting values of $\hat{W}$ are likely to be poor. Purging facilitates usage of better estimates as they become available.

The developed purging technique utilizes two history stacks, a main history stack and a transient history stack, labeled $\mathcal{H}^{IRL}$ and $\mathcal{G}^{IRL}$, respectively. As soon as $\mathcal{G}^{IRL}$ is full and sufficient dwell time has elapsed since the last purge (see Section V-B in [22]), $\mathcal{H}^{IRL}$ is emptied and $\mathcal{G}^{IRL}$ is copied into $\mathcal{H}^{IRL}$.

The recursive update law for estimation of the unknown weights is then designed as

$$\dot{\hat{W}} = \alpha \Gamma \hat{\Sigma}^T \left(-\hat{\Sigma}\hat{W} - \hat{\Sigma}_{u1}\right). \quad (22)$$

In (22), $\alpha \in \mathbb{R}_{>0}$ 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 tuned using the update law

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\hat{\Sigma}^T\hat{\Sigma}\Gamma, \quad (23)$$

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

B. Analysis

The time-varying history stack, $\mathcal{H}^{IRL}$, is called full rank, uniformly in t, if there exists a $\underline{\sigma} > 0$ such that $\forall t \in \mathbb{R}_{\geq T_0}$,

$$0 < \underline{\sigma} < \lambda_{\min}\left\{\hat{\Sigma}^T(t)\hat{\Sigma}(t)\right\}. \quad (24)$$

Using arguments similar to [21, Corollary 4.3.2], it can be shown that if $\lambda_{\min}\left\{\Gamma^{-1}(T_0)\right\} > 0$, and if $\mathcal{H}^{IRL}$ is full rank, uniformly in t, then the least squares gain matrix satisfies

$$\underline{\Gamma} I_{L+P+m-1} \leq \Gamma(t) \leq \overline{\Gamma} I_{L+P+m-1}, \forall\, t \in \mathbb{R}_{\geq T_0}, \quad (25)$$

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

To facilitate the following Lyapunov analysis, using (22), the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}} = -\alpha\Gamma\hat{\Sigma}^T\left(\hat{\Sigma}\tilde{W} - \Delta\right). \quad (26)$$

The stability result is summarized in the following theorem.

Theorem 2. *If $\mathcal{H}^{IRL}$ is full rank, uniformly in t, then $t \mapsto \tilde{W}(t)$ is ultimately bounded.*

Proof. For brevity, the details of the proof has been omitted (see [22, Theorem 2]). □

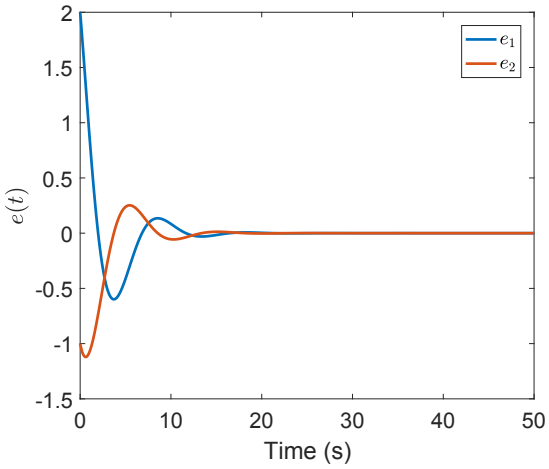

Fig. 1: Trajectory tracking error.

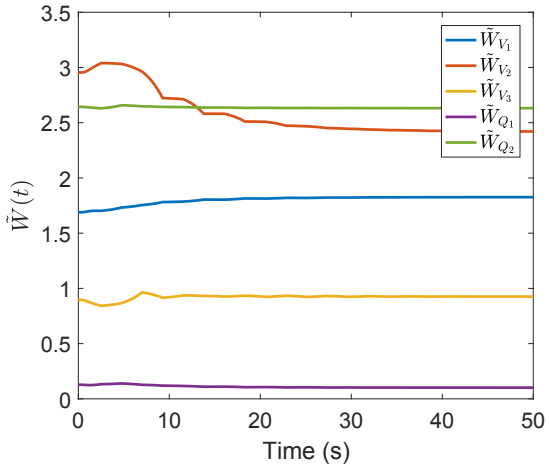

Fig. 2: Reward and value function weight estimation errors without feedback extrapolation.

VI. Simulation

To demonstrate the performance of the developed method by comparing weight estimates with their true values, a linear optimal trajectory tracking problem with known optimal controller and optimal value function is selected for the simulation study [23], [24].

Consider an agent with the linear dynamics

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \theta_1 & \theta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \theta_3 \end{bmatrix} u, \tag{27}$$

where the unknown parameters are $\theta_1 = -0.5, \theta_2 = -0.5$, and $\theta_3 = 1$. The parameter estimation technique is utilized to satisfy Assumption 4 developed in [19].

The trajectory the agent is attempting to follow is generated from the linear system

$$\dot{x}_d = \begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix} x_d. \tag{28}$$

The optimal control problem is to minimize the cost functional

$$J(e_0, \mu(\cdot)) = \int_{T_0}^{\infty} \left(e(t)^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} e(t) + 10\mu(t)^2 \right) \mathrm{d}t,$$

subject to the error dynamics

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ -0.5 & -0.5 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu, \tag{29}$$

where $e = x - x_d$, $\mu = u - u_d$, and $u_d = [-1.5,\ 0.5]\ x_d$. The resulting ideal reward function weights are $Q = \mathrm{diag}([W_{Q_1},\ W_{Q_2}]) = \mathrm{diag}([1,\ 1])$ and $R = 10$. 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 $W_{V_1} = 1.82, W_{V_2} = 2.30$, and $W_{V_3} = 1.83$. The optimal controller is given by $\mu = -[0.092,\ 0.230]e$, resulting in the ideal weights $W_{\mu_1} = -0.091$ and $W_{\mu_2} = 0.230$.

The learning gains selected for the two simulations are: $\beta = 0.5, \alpha = {}^{0.01}/_{50}, \beta_u = 2, \alpha_u = 1, M = 50, N = 50$ and a step size of $0.005s$.

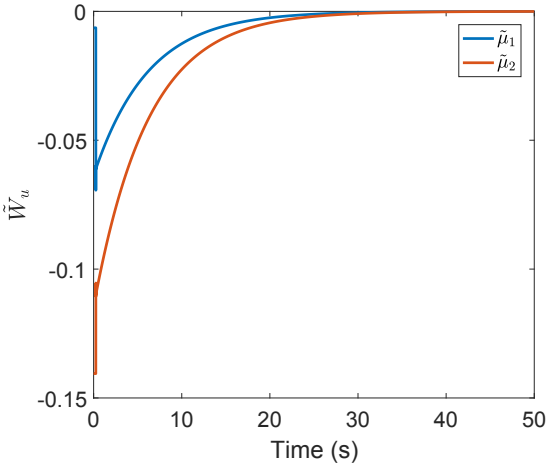

Fig. 3: Optimal feedback controller estimation error.

A. IRL without Feedback Extrapolation

The first simulation utilizes the state and control trajectories directly for IRL similar to [14], and does not estimate the optimal controller for additional data. Fig. 2 shows reward and value function estimation errors without queried data.

As seen in Fig. 2, the reward and value function estimates do not converge to the ideal values. In fact, the estimates do not change much at all. Once the trajectory tracking errors converge to zero (within $15s$ in Fig. 1), and $\mathcal{H}^{IRL}$ is purged to remove transient, erroneous parameter estimates, the remaining error trajectories, and hence the data points in $\mathcal{H}^{IRL}$, are near zero. Since the weights in Fig. 2 are estimated using $\mathcal{H}^{IRL}$, the large estimation errors demonstrated in Fig. 2 are to be expected.

B. IRL with Feedback Extrapolation

The second simulation shows the results of the novel control-estimation-based technique developed in this paper, with artificially generated state-action pairs. Utilizing the

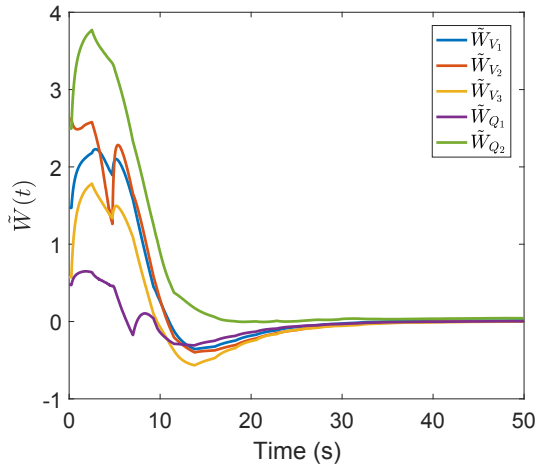

Fig. 4: Reward and value function weight estimation errors with feedback extrapolation.

estimate of the optimal policy, estimates of the optimal controller, $\hat{\mu}_i$, corresponding to states e_i, randomly selected in the set $[-1,1]\times[-1,1]$, are produced. The pairs $(e_i, \hat{\mu}_i)$ are then iteratively collected in $\mathcal{H}^{IRL}$ and IRL is implemented using the update law in (22).

Fig. 3 shows the estimation error for the optimal feedback controller, and Fig. 4 shows the reward and value function weight estimation errors.

As seen in Fig. 4, the new IRL approach estimates the ideal values of the reward and value function weights online. Though the tracking errors of the system dynamics have already converged at around $15s$ (see Fig. 1), due to the non-zero artificially generated state and control values available through feedback policy estimation, the developed IRL method is able to estimate the reward and value function weights.

VII. Conclusion

This paper develops a new approach to performing reward function estimation online in situations with limited data. The approach utilizes a concurrent learning update law to estimate the optimal feedback policy of the agent, online. This estimate is synchronously utilized to artificially create additional data to facilitate reward and value function estimation. Theoretical guarantees are provided for ultimate boundedness of the reward and value function weight estimation errors using Lyapunov theory. A simulation example is presented that demonstrates the benefit of the developed method when compared with a previous IRL technique implemented without queried data.

Future work will include an analysis of the performance of the developed approach for systems with unmeasurable states and the effect of noise on optimal control estimation.

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

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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: IEEE, Aug. 2019, pp. 296–301.

[15] 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.

[16] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1985.

[17] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, pp. 251–257, 1991.

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

[19] 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.

[20] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, Feb. 2016.

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

[22] R. Self, N. S. M. Mahmud, K. Hareland, and R. Kamalapurkar, "Online inverse reinforcement learning with limited data," arXiv:2008.08972, 2020.

[23] 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.

[24] 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.