Online inverse reinforcement learning for systems with disturbances

Ryan Self, Moad Abudia and Rushikesh Kamalapurkar

Abstract— **This paper addresses the problem of online inverse reinforcement learning for nonlinear systems with modeling uncertainties and additive disturbances. In the developed approach, the learner measures state and input trajectories of the demonstrator and identifies its unknown reward function online. Sub-optimality introduced in the measured trajectories by the unknown external disturbance is compensated for using a novel model-based inverse reinforcement learning approach. The learner estimates the external disturbances and uses them to identify the dynamic model of the demonstrator. The learned model along with the observed sub-optimal trajectories are used for 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. The reward function estimation is performed in the presence of modeling uncertainties and unknown disturbances via inverse reinforcement learning (IRL) [1], [2].

IRL [1]–[12] and inverse optimal control (IOC) methods [13] are extensively utilized to teach autonomous machines to perform specific tasks in an *offline* setting. However, these *offline* approaches are, in general, too computationally intensive or require more data than online situations provide. Inspired by the success of model-based real-time reinforcement learning methods in, e.g., [14] and [15], and the online IRL/IOC results for linear systems in [16] and [17], this paper presents an online IRL technique for nonlinear systems.

The main contribution of this paper is the development of a novel method for reward function estimation for an agent with unknown dynamics in the presence of disturbances. The developed technique in this paper builds on the previous work in [18] where a batch IRL method is utilized that relies on optimal demonstrations, and as such, does not consider external disturbances affecting the agent being observed. Addressing the complexities resulting for disturbance-induced sub-optimality of the demonstrations is one of the major technical contributions of this paper. In addition, the novel continuous IRL results in smoother weight estimates and admits Lyapunov-based performance guarantees.

Model-free IRL methods, in general, are entirely trajectory driven, and require either optimal/near-optimal trajectories or requires sub-optimal trajectories to be rare occurrences [6]. However, if the agent under observation is experiencing external disturbances, then not all trajectories are optimal with respect to the same cost function, which makes model-free IRL difficult. Even if the unknown disturbances can be estimated, removing the effects of these disturbances is nontrivial in a model-free IRL setting.

The novelty in the technique developed in this paper is the use of model-based IRL to compensate for the disturbance-induced sub-optimality. If a dynamic model of the demonstrator is unavailable, it needs to be identified from the data. However, the disturbances make system identification challenging, and the resulting models are typically poor. To overcome this challenge, it is assumed that the learner and demonstrator are co-located and as a result, experience the same disturbance. One can then estimate the disturbance using its effects on the learner and use the resulting estimates to identify the dynamic model of the demonstrator. A model-based IRL method can then be deployed to learn the unknown reward function.

The paper is organized as follows: Section II explains the notation used throughout the paper. Section III details the problem formulation and how the additional challenges related to disturbances are addressed. Section IV details the disturbance estimator. Section V shows the developed parameter estimator. Section VI explains the IRL algorithm. Section VII shows a simulation example for the proposed method and Section VIII concludes the paper.

II. 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, ∞). 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 two agents, Agent 1 and Agent 2, where Agent 1 is monitoring the behavior of Agent 2. Agent 1 has the following dynamics

$$\dot{x}_1 = f_1(x_1, u_1) + d_1, \tag{1}$$

where $x_1 : \mathbb{R}_{\geq T_0} \to \mathbb{R}^n$ is the state, $u_1 : \mathbb{R}_{\geq T_0} \to \mathbb{R}^m$ is the control, $f_1 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are the dynamics, d_1 :

The authors are with the School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK, USA. {rself, abudia, 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.

$\mathbb{R}_{\geq T_0} \to \mathbb{R}^n$ is a disturbance acting on Agent 1, and T_0 is the initial time. The dynamics for Agent 2 are

$$\dot{x}_2 = f_2(x_2, u_2) + d_2, \tag{2}$$

where $x_2 : \mathbb{R}_{\geq T_0} \to \mathbb{R}^n$ is the state, $u_2 : \mathbb{R}_{\geq T_0} \to \mathbb{R}^m$ is the control, $f_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are the dynamics, and $d_2 : \mathbb{R}_{\geq T_0} \to \mathbb{R}^n$ is a disturbance acting on the Agent 2.

Agent 2 is attempting to follow a policy that minimizes the following performance index

$$J(x_0, u(\cdot)) = \int_{T_0}^{\infty} r(x(t; x_0, u(\cdot)), u(t))\,\mathrm{d}t, \tag{3}$$

where $x\left(\cdot; x_0, u(\cdot)\right)$ is the trajectory generated by the optimal controller $u(\cdot)$ for the undisturbed dynamics that minimizes the performance index in (3) starting at x_0 and beginning at time T_0. The main objective of this paper is to estimate the unknown reward function, r, in the presence of disturbances and uncertainties in the dynamics.

The following assumptions are used in the analysis of the paper.

Assumption 1. *The disturbances affecting both agents are identical, i.e.* $d_1\left(t\right) = d_2\left(t\right) = d(t), \forall t$.

Assumption 2. *The unknown reward function* r *is quadratic in the control, i.e.,*

$$r(x, u) = Q(x) + u^T R u, \tag{4}$$

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

The continuous function Q can be represented using a neural network 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.

Assumption 3. *The dynamics for Agent 2 can be expressed as*

$$\dot{x}_2 = f_2^o(x_2, u_2) + \theta_2^T \sigma_2(x_2, u_2) + d, \tag{5}$$

where $f_2^o : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ *denotes the nominal dynamics,* $\theta_2^T \sigma_2$ *is a parameterized estimate of the uncertain part of the dynamics,* $\theta_2 \in \mathbb{R}^{p \times n}$ *is a matrix of unknown constant parameters, and* $\sigma_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ *are known features.*

If Agent 1 and Agent 2 are co-located and of similar size then the disturbances affecting them can be reasonably assumed to be equal. Assumption 2 facilitates the IRL problem formulation in Section VI, and Assumption 3 facilitates the parameter estimation in Section V.

Due to the unknown disturbance d acting on Agent 2, the trajectories of Agent 2 will no longer be optimal with respect to its unknown reward function. As a result, a purely data-driven implementation of IRL would yield incorrect reward function estimates. Instead, in this paper, the state trajectories for Agent 2 are measured and the reward function is estimated using a model-based approach that compensates for the trajectory deviations. The unknown disturbance, d, is estimated by Agent 1 using its known internal model, and Agent 1 implements a parameter estimator that incorporates the disturbance estimates to calculate the unknown parameters in the dynamics of Agent 2. Finally, both the disturbance and parameter estimates are used by Agent 1 to estimate the unknown reward function that Agent 2 is attempting to optimize. Disturbance estimation, parameter estimation, and inverse reinforcement learning, are performed in parallel and in real-time.

IV. Disturbance Estimation

While the IRL method discussed in the following can be developed using any disturbance estimator that results in uniform ultimate boundedness of the disturbance estimation error, the following exponential disturbance estimator (inspired by [19]) is used in this paper for ease of exposition. Since the disturbance estimation is performed only by Agent 1, the subscripts in the dynamics that denote the agent number will be omitted in this section for clarity.

The unknown disturbance acting on the agents is assumed to be an additive disturbance that is generated from the exogenous linear system

$$\dot{\zeta} = A\zeta, \tag{6}$$

$$d = C\zeta, \tag{7}$$

where $\zeta : \mathbb{R}_{\geq T_0} \to \mathbb{R}^N, A \in \mathbb{R}^{N \times N}, C \in \mathbb{R}^{n \times N}$, and $d : \mathbb{R}_{\geq T_0} \to \mathbb{R}^n$ is the disturbance.

The disturbance estimator is designed as

$$\dot{\hat{\zeta}} = A\hat{\zeta} + K\left(\dot{x} - \left(f\left(x, u\right) + \hat{d}\right)\right), \tag{8}$$

and

$$\hat{d} = C\hat{\zeta}, \tag{9}$$

where $K \in \mathbb{R}^{N \times n}$ is a gain matrix.

The following theorem utilizes Lyapunov-based arguments to establish exponential convergence of the disturbance estimator.

Theorem 1. *If* $(A - KC)$ *is negative definite, then the disturbance estimation error converges exponentially to zero.*

Proof. For brevity, the details of the proof has been omitted (see [20, Theorem 1]). □

V. Parameter Estimation

A parameter estimator, motivated by the authors' previous work in [21], is developed in this section. Since parameter estimation is performed only for Agent 2, the subscripts that denote the agent number in the dynamics will be omitted in this section for clarity.

A. Parameter Estimator

Integrating (5) over the interval $[t-T,t]$ for some constant $T\in\mathbb{R}_{>0}$,[1]

$$x(t)-x(t-T)=\int_{t-T}^{t}f^{o}(x(\gamma),u(\gamma))\,\mathrm{d}\gamma$$
$$+\theta^{T}\int_{t-T}^{t}\sigma(x(\gamma),u(\gamma))\,\mathrm{d}\gamma+\int_{t-T}^{t}d(\gamma)\,\mathrm{d}\gamma. \quad (10)$$

The expression in (10) can be rearranged to form the affine system

$$X(t)=F(t)+\theta^{T}S(t)+D(t),\ \forall t\in\mathbb{R}_{\geq T_0} \quad (11)$$

where

$$X(t):=\begin{cases}x(t)-x(t-T), & t\in[T_0+T,\infty),\\ 0, & t<T_0+T,\end{cases} \quad (12)$$

$$F(t):=\begin{cases}\int_{t-T}^{t}f^{o}(x(\gamma),u(\gamma))\,\mathrm{d}\gamma, & t\in[T_0+T,\infty),\\ 0, & t<T_0+T,\end{cases} \quad (13)$$

$$S(t):=\begin{cases}\int_{t-T}^{t}\sigma(x(\gamma),u(\gamma))\,\mathrm{d}\gamma, & t\in[T_0+T,\infty),\\ 0, & t<T_0+T,\end{cases} \quad (14)$$

and

$$D(t):=\begin{cases}\int_{t-T}^{t}d(\gamma)\,\mathrm{d}\gamma, & t\in[T_0+T,\infty),\\ 0, & t<T_0+T.\end{cases} \quad (15)$$

The affine error system in (11) motivates the adaptive estimation scheme that follows, in which a *concurrent learning* [22] technique is developed that utilizes recorded data stored in a history stack to drive parameter estimation.

A history stack, $\mathcal{H}^{PE}$, is a set of data points $\left\{\left(X_i,F_i,S_i,\hat{D}_i\right)\right\}_{i=1}^{M}$ such that

$$X_i=F_i+\theta^{T}S_i+\hat{D}_i+\mathcal{E}_i,\ \forall i\in\{1,\cdots,M\}, \quad (16)$$

where $\mathcal{E}_i=D_i-\hat{D}_i$, and

$$\hat{D}(t):=\begin{cases}\int_{t-T}^{t}\hat{d}(\gamma)\,\mathrm{d}\gamma, & t\in[T_0+T,\infty),\\ 0, & t<T_0+T.\end{cases} \quad (17)$$

$\mathcal{H}^{PE}$ is called *full rank* if there exists a constant $\underline{c}\in\mathbb{R}$ such that

$$0<\underline{c}<\lambda_{\min}\{\mathscr{S}\}, \quad (18)$$

where the matrix $\mathscr{S}\in\mathbb{R}^{p\times p}$ is defined as $\mathscr{S}:=\sum_{i=1}^{M}S_iS_i^{T}$. The concurrent learning update law to estimate the unknown parameters is then given by

$$\dot{\hat{\theta}}=\alpha_\theta\Gamma_\theta\sum_{i=1}^{M}S_i\left(X_i-F_i-\hat{\theta}^{T}S_i-\hat{D}_i\right)^{T}, \quad (19)$$

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

$$\dot{\Gamma}_\theta=\beta_\theta\Gamma_\theta-\alpha_\theta\Gamma_\theta\mathscr{S}\Gamma_\theta, \quad (20)$$

where $\beta_\theta\in\mathbb{R}_{>0}$ is a constant gain. Using arguments similar to [23, Corollary 4.3.2], it can be shown that provided $\lambda_{\min}\left\{\Gamma_\theta^{-1}(0)\right\}>0$, the least squares gain matrix satisfies

$$\underline{\Gamma}_\theta\mathrm{I}_p\leq\Gamma_\theta(t)\leq\overline{\Gamma}_\theta\mathrm{I}_p, \quad (21)$$

where $\underline{\Gamma}_\theta$ and $\overline{\Gamma}_\theta$ are positive constants, and I_p denotes an $p\times p$ identity matrix. If a full rank history stack that satisfies (16) is not available a priori, then the data points can be recorded online. The history stack $\mathcal{H}^{PE}$, if time-varying, is called full-rank, uniformly in t if $\underline{c}$ in (18) is independent of t.

From the Lyapunov analysis in Section V-B, it is observed that the rate of decay for the parameter estimation error is proportional to the minimum singular value of $\mathscr{S}$. Therefore, to promote faster convergence for the parameter estimates, a minimum singular value maximization algorithm is developed. At each time t, the algorithm takes the current new data point, $\left(X^*,F^*,S^*,\hat{D}^*\right)$, and checks if replacing the new data point with any data point currently in the history stack increases the minimum singular value of $\mathscr{S}$. If the new data point does increase the minimum singular value, that is,

$$\lambda_{\min}\left(\sum_{i\neq j}S_iS_i^{T}+S_jS_j^{T}\right)<\frac{\lambda_{\min}\left(\sum_{i\neq j}S_iS_i^{T}+S^*S^{*T}\right)}{(1+\psi)}, \quad (22)$$

where $\lambda_{\min}$ represents the minimum singular value of a matrix and ψ is a positive constant, then the new data point replaces the data point currently in the $\mathcal{H}^{PE}$ that results in the largest increase in the minimum singular value, if not the new point is discarded.

Using Lyapunov arguments, it can be shown (see Section V-B) that the parameter estimation error is directly related to the error $\mathcal{E}_i$ in (16). Due to the fact that newer values of $\hat{D}_i$ result in smaller $\mathcal{E}_i$ due to the exponential convergence of the disturbance estimates, a purging algorithm is developed to eliminate inaccurate data from $\mathcal{H}^{PE}$.

The algorithm maintains two history stacks, a main history stack and a transient history stack, labeled $\mathcal{H}^{PE}$ and $\mathcal{G}^{PE}$, respectively. As soon as $\mathcal{G}^{PE}$ is full and sufficient time has elapsed since the last purge (see Section V-B), $\mathcal{H}^{PE}$ is emptied and $\mathcal{G}^{PE}$ is copied into $\mathcal{H}^{PE}$.

B. Analysis

A Lyapunov based analysis, summarized in the following theorem, is performed to show convergence for the parameter estimator developed in Section V-A.

To facilitate the following Lyapunov analysis, the dynamics for the parameter estimation error can be expressed as

$$\dot{\tilde{\theta}}=-\alpha_\theta\Gamma_\theta\mathscr{S}\tilde{\theta}-\alpha_\theta\Gamma_\theta\sum_{i=1}^{M}S_i\mathcal{E}_i, \quad (23)$$

[1] If the integration interval is selected to be too short, there may not be enough information in the vector X_i to achieve accurate parameter estimation. If the integration interval is selected too long, parameter estimates may not be available during transients where they are needed the most. The development of a reasonable heuristic that guides the selection of the integration interval is a topic for future research.

by using (16) and (19), along with the error being defined as $\tilde{\theta} = \theta - \hat{\theta}$.

The stability result is summarized in the following theorem.

Theorem 2. *If the history stack $\mathcal{H}^{PE}$ is full rank, uniformly in t, and $\tilde{d}$ converges to zero exponentially, then* $\lim_{t\to\infty} \|\tilde{\theta}(t)\| = 0$.

Proof. For brevity, the details of the proof has been omitted (see [20, Theorem 2]). □

VI. Inverse Reinforcement Learning

The formulation of IRL in the following two sections closely follows the authors' previous work in [18]. In addition, since IRL is performed only on Agent 2, the subscripts that denote the agent number in the dynamics will omitted in the next sections for clarity.

A. Inverse Bellman Error

Under the premise that Agent 2 implements a feedback controller that would be optimal in a disturbance-free environment, the state and control trajectories, $x(\cdot)$ and $u(\cdot)$, satisfy the Hamilton-Jacobi-Bellman (HJB) equation

$$H\left(x\left(t\right), \nabla_x \left(V^*\left(x\left(t\right)\right)\right)^T, u\left(t\right)\right) = 0, \forall t \in \mathbb{R}_{\geq 0}, \tag{24}$$

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 $H(x,p,u) := p^T f(x,u) + r(x,u)$. The goal of IRL is to accurately estimate the reward function, r. To aid in the estimation of the reward function, let $\hat{V} : \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}$, $\left(x, \hat{W}_V\right) \mapsto \hat{W}_V^T \sigma_V\left(x\right) + \epsilon_V(x)$ 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^*, $\sigma_V : \mathbb{R}^n \to \mathbb{R}^P$ are known continuously differentiable features, and $\epsilon_V : \mathbb{R}^n \to \mathbb{R}$ is the resulting approximation error. Using $\hat{\theta}$, $\hat{W}_V$, $\hat{W}_Q$, and $\hat{W}_R$, which are the estimates of θ, W_V^*, W_Q^*, and $W_R^* := \left[r_1, \cdots, r_m\right]^T$, respectively, in (24), the inverse Bellman error $\delta' : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+m} \times \mathbb{R}^p \to \mathbb{R}$ is obtained as

$$\begin{aligned}\delta'\left(x, u, \hat{W}, \hat{\theta}\right) =& \hat{W}_V^T \nabla_x \sigma_V\left(x\right)\ \hat{Y}(x,u,\hat{\theta}) + \hat{W}_Q^T \sigma_Q\left(x\right) \\ &+ \hat{W}_R^T \sigma_u\left(u\right),\end{aligned} \tag{25}$$

where $\sigma_u\left(u\right) := \left[u_1^2, \cdots, u_m^2\right]$ and $\hat{Y}(x,u,\hat{\theta}) = \left[f^o(x,u) + \hat{g}(x,u,\hat{\theta})\right]$ where $\hat{g}(x,u,\hat{\theta}) := \hat{\theta}^T \sigma(x,u)$ from (5). Rearranging, (25) becomes

$$\delta'\left(x, u, \hat{W}', \hat{\theta}\right) = \left(\hat{W}'\right)^T \sigma'\left(x, u, \hat{\theta}\right), \tag{26}$$

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

B. Inverse Reinforcement Learning Formulation

Using the formulation of the inverse Bellman error in Section VI-A, and control signals, trajectories, and parameter estimates stored in a history stack, denoted as $\mathcal{H}^{IRL}$, the inverse Bellman error, evaluated at time instances $t_1, t_2, \ldots, t_N$, can be formulated into the matrix form

$$\Delta' = \hat{\Sigma}' \hat{W}', \tag{27}$$

where $\Delta' := \left[\delta_t'\left(t_1\right); \cdots; \delta_t'\left(t_N\right)\right]$, $\delta_t'\left(t\right) := \delta'\left(x\left(t\right), u\left(t\right), \hat{W}', \hat{\theta}\left(t\right)\right)$, $\hat{\Sigma}' := \left[\left(\hat{\sigma}_t'\right)^T\left(t_1\right); \cdots; \left(\hat{\sigma}_t'\right)^T\left(t_N\right)\right]$, and $\hat{\sigma}_t'(t) := \sigma'\left(x(t), u(t), \hat{\theta}(t)\right)$.

The HJB equation in (24) implies that whenever $\hat{W} = W^*$, the inverse Bellman error is zero. As a result, candidate solutions of the IRL problem can be obtained by solving (27) for $\hat{W}$ so that $\Delta' = 0$. The linear system is now a homogeneous system of linear equations, and it can only be solved up to a scaling factor. Since optimal state and control trajectories are invariant with respect to scaling of the cost function, the scaling ambiguity is to be expected. Since optimal control behaviours are scale-invariant, there is no loss of generality in resolving the scale ambiguity by assigning a fixed known value to one of the reward function weights.

Taking the first element of $\hat{W}_R$ to be known, the inverse BE in (26) can then be expressed as

$$\delta'\left(x, u, \hat{W}, \hat{\theta}\right) = \hat{W}^T \sigma''\left(x, u, \hat{\theta}\right) + r_1 \sigma_{u1}\left(u\right), \tag{28}$$

where $\hat{W} := \left[\hat{W}_V; \hat{W}_Q; \hat{W}_R^-\right]$, the vector $\hat{W}_R^-$ denotes $\hat{W}_R$ with the first element removed, $\sigma_{ui}\left(u\right)$ denotes the ith element of the vector $\sigma_u\left(u\right)$, the vector σ_u^- denotes σ_u with the first element removed, and $\sigma''\left(x, u, \hat{\theta}\right) := \left[\nabla_x \sigma_V\left(x\right) \hat{Y}(x,u,\hat{\theta}); \sigma_Q\left(x\right); \sigma_u^-\left(u\right)\right]$.

The closed-form nonlinear optimal controller corresponding to the reward structure in (3) provides the relationship

$$\begin{aligned}-2Ru\left(t\right) = \left(g'(x(t))\right)^T \left(\nabla_x \sigma_V\left(x\left(t\right)\right)\right)^T W_V^* \\ + \left(g'(x(t))\right)^T \nabla_x \epsilon_V\left(x\left(t\right)\right),\end{aligned} \tag{29}$$

which can be expressed as

$$\begin{aligned}-2r_1 u_1\left(t\right) + \Delta_{u1} &= \sigma_{g1} \hat{W}_V \\ \Delta_{u^-} &= \sigma_g^- \hat{W}_V + 2\mathrm{diag}\left(u_2, \cdots, u_m\right) \hat{W}_R^-,\end{aligned}$$

where $g'(x) := \nabla_u f(x,u)$, σ_{g1} and Δ_{u_1} denote the first rows and σ_g^- and Δ_{u^-} denote all but the first rows of $\sigma_g(x) := \left(g'(x)\right)^T \left(\nabla_x \sigma_V\left(x\right)\right)^T$ and $\Delta_u(x) := \left(g'(x)\right)^T \nabla_x \epsilon_V(x)$, respectively, and $R^- := \mathrm{diag}\left(\left[r_2, \cdots, r_m\right]\right)$. For simplification, let $\sigma := \left[\sigma'', \begin{bmatrix}\sigma_g^T \\ \Theta\end{bmatrix}\right]$, where

$$\Theta := \left[0_{m\times 2n}, \begin{bmatrix} 0_{1\times m-1} \\ 2\mathrm{diag}\left(\left[u_2, \cdots, u_m\right]\right)\end{bmatrix}\right]^T.$$

Updating matrix form in (27) by removing the known reward weight results in the linear system

$$-\Sigma_{u1} = \hat{\Sigma}\hat{W} - \Delta', \tag{30}$$

where $\hat{\Sigma} := \left[\hat{\sigma}_t^T(t_1);\cdots;\hat{\sigma}_t^T(t_N)\right]$, and $\Sigma_{u1} := [\sigma'_{u1}(u(t_1));\cdots;\sigma'_{u1}(u(t_N))]$, where $\hat{\sigma}_t(\tau) := \sigma\left(x(\tau), u(\tau), \hat{\theta}(\tau)\right)$, $\sigma'_{u1}(\tau) := \left[r_1\sigma_{u1}(\tau); 2r_1u_1(\tau); 0_{(m-1)\times 1}\right]$.

The recursive update law is then designed as

$$\dot{\hat{W}} = \alpha\Gamma\hat{\Sigma}^T\left(-\hat{\Sigma}\hat{W} - \Sigma_{u1}\right). \tag{31}$$

In (31), $\alpha \in \mathbb{R}_{>0}$ 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

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\hat{\Sigma}^T\hat{\Sigma}\Gamma, \tag{32}$$

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

C. Analysis

A Lyapunov based analysis is performed to show convergence for the IRL method in Section VI-B.

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

$$0 < \underline{\sigma} < \lambda_{\min}\left\{\hat{\Sigma}^T(t)\hat{\Sigma}(t)\right\}. \tag{33}$$

Using arguments similar to [23, 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

$$\underline{\Gamma}\mathrm{I}_{L+P+m-1} \leq \Gamma(t) \leq \overline{\Gamma}\mathrm{I}_{L+P+m-1}, \tag{34}$$

where $\underline{\Gamma}$ and $\overline{\Gamma}$ are positive constants, and I_n denotes an $n\times n$ identity matrix.

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

$$\dot{\tilde{W}} = -\alpha\Gamma\hat{\Sigma}^T\left(\hat{\Sigma}\tilde{W} + \Delta_\theta\right), \tag{35}$$

using the fact that $\tilde{W} = W^* - \hat{W}$, along with (31) and the equation $-\Sigma_{u1} = \hat{\Sigma}W^* + \Delta_\theta$, where Δ_θ denotes the errors resulting from poor $\hat{\theta}$ estimates incorporated in $\hat{\Sigma}$.

The stability result is summarized in the following theorem.

Theorem 3. *If $\mathcal{H}^{PE}$ and $\mathcal{H}^{IRL}$ are full rank, uniformly in t, and $\tilde{d}$ converges to zero exponentially, then $t \mapsto \tilde{W}(t)$ is uniformly ultimately bounded (UUB).*

Proof. For brevity, the details of the proof has been omitted (see [20, Theorem 3]). □

VII. Simulation

To demonstrate the performance of the developed method, a nonlinear optimal control problem is constructed using [13] in order to have a known value function for comparison.

Agent 1 has the following nonlinear dynamics

$$\dot{x}_{1_1} = x_{1_2}, \qquad \dot{x}_{1_2} = x_{1_1}x_{1_2} + 3x_{1_2}^2 + 5u_1 + d.$$

Agent 2 under observation has the following nonlinear dynamics

$$\begin{aligned}\dot{x}_{2_1} &= x_{2_2},\\ \dot{x}_{2_2} &= \theta_1 x_{2_1}\left(\frac{\pi}{2} + \tan^{-1}(5x_{2_1})\right) + \frac{\theta_2 x_{2_1}^2}{1+25x_{2_1}^2}\\ &\quad + \theta_3 x_{2_2} + 3u_2 + d,\end{aligned} \tag{36}$$

where x_{i_j} denotes jth state variable for Agent i. The parameters θ_1, θ_2, and θ_3 are unknown constants to be estimated and d is the unknown disturbance. The exact values of these parameters are $\theta_1 = -1, \theta_2 = -\frac{5}{2}$, and $\theta_3 = 4$. The disturbance, d, acting on the agents is generated from the linear system in Section IV, where $A = [0, 1; -1, 0]$ and $C = [0, 0; 1, 0]$, and the chosen gain matrix is $K = [1, 0.5; 0, 5]$.

The performance index that the agent is trying to minimize is

$$J(x_0, u_2(\cdot)) = \int_0^\infty (x_{2_2}^2 + u_2^2)dt,$$

resulting in the ideal reward function weights $Q = \mathrm{diag}(q_1, q_2) = \mathrm{diag}(0, 1)$ and $R = 1$. The observed state and control trajectories, and the disturbance estimates are used in the estimation of unknown parameters in the dynamics, along with the optimal value function parameters and the reward function weights. The optimal controller is $u_2^* = -3x_{2_2}$, while the optimal value function is $V^* = x_{2_1}^2(v_1 + v_2\tan^{-1}(5x_{2_1})) + v_3x_{2_2}^2$, resulting in the ideal function parameters $v_1 = \frac{\pi}{2}$, $v_2 = 1$, and $v_3 = 1$. Figs. 1 and 2 show the performance of the proposed method. Fig. 1 shows convergence of the unknown part of Agent 2's dynamics, and Fig. 2 shows convergence of the unknown reward function. Fig. 3 shows the convergence of the disturbance estimates. The parameters used for the simulation are: $T = 1.2s$, $N = 100$, $M = 150$, $\beta = \beta_\theta = 0.5$, $\alpha = \alpha_\theta = {}^1\!/_N$, and a time step of $0.0005s$.

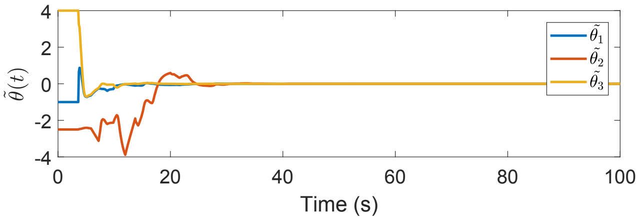

Fig. 1. Estimation error for the unknown parameters in Agent 2's dynamics.

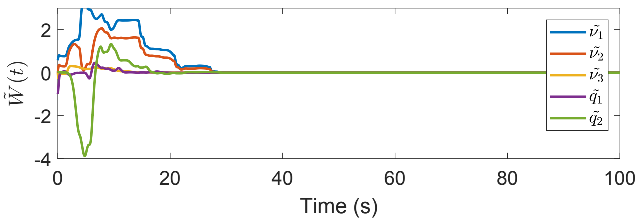

Fig. 2. Estimation error for the unknown parameters in the reward function for Agent 2.

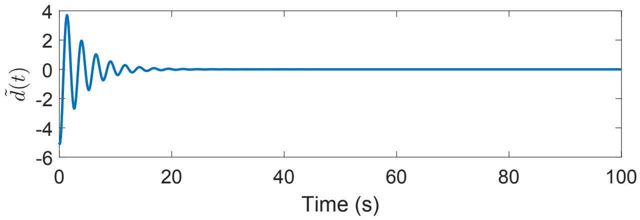

Fig. 3. Estimation error for the unknown disturbance acting on the two agents.

VIII. Conclusion

A novel IRL framework is developed in this paper for reward function estimation in the presence of modeling errors and additive disturbances. To compensate for disturbance-induced sub-optimality of observed trajectories, a model-based approach is developed that relies on a disturbance estimator.

Future work will focus on the development of output feedback IRL methods that utilize both state and parameter estimation methods, and extensions of the developed method for disturbances that affect the agents through a control effectiveness matrix. The authors will additionally explore the use of implicit disturbance estimation techniques that would result in bounded, but nonzero disturbance estimation errors.

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 *Proceedings of the eleventh annual conference on Computational learning 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] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Auton. Robot.*, vol. 28, no. 3, pp. 369–383, 2010.

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

[14] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[15] D. Wang, D. Liu, H. Li, B. Luo, and H. Ma, "An approximate optimal control approach for robust stabilization of a class of discrete-time nonlinear systems with uncertainties," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 46, no. 5, pp. 713–717, 2016.

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

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

[18] 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.

[19] W.-H. Chen, "Disturbance observer based control for nonlinear systems," *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 4, pp. 706–710, 2004.

[20] R. Self, M. Abudia, and R. Kamalapurkar, "Online inverse reinforcement learning with unknown disturbances," arXiv:2003.03912, 2020.

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

[22] G. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, Georgia Institute of Technology, Dec. 2010.

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