Pilot Performance modeling via observer-based inverse reinforcement learning

Jared Town, Zachary Morrison, and Rushikesh Kamalapurkar

***Abstract*—The focus of this paper is behavior modeling for pilots of unmanned aerial systems. The pilot is assumed to make decisions that optimize an unknown cost functional. The cost functional is estimated from observed trajectories using a novel inverse reinforcement learning (IRL) framework. The resulting IRL problem often admits multiple solutions. In this paper, a recently developed novel IRL observer is adapted to the pilot behavior modeling problem. The observer is shown to converge to one of the equivalent solutions of the corresponding IRL problem. The developed technique is implemented on a quadcopter where the pilot is a surrogate linear quadratic controller that generates velocity commands for set-point regulation of the quadcopter. Experimental results demonstrate the ability of the developed method to learn equivalent cost functionals.**

***Index Terms*—Inverse Reinforcement Learning, Inverse Optimal Control, Pilot Behavior Modeling**

I. Introduction

Given the widespread use of small unmanned aerial systems (sUAS), quadcopters in particular, the need to manage flights efficiently at low altitudes arises as that airspace is cluttered and turbulent. Cooperative piloting is necessary for the guidance of these quadcopters to prevent air-to-air and air-to-obstacle collisions. Piloting a small quadcopter in a windy and obstacle-laden environment is a difficult task for pilots to manage without assistance. We envision a pilot-assist system that recommends paths to the pilots that are personalized to suit their preferences and skill levels. To develop such a system, pilot performance is modeled in terms of a cost functional that is learned by analyzing the control inputs of the pilot. The learned cost functional can then be paired with existing optimal control techniques to generate personalized path/trajectory recommendations. Such optimization is not discussed in this paper, this study exclusively focuses on the cost functional estimation component of the envisoned recommendation system.

Taking inspiration from [1]–[3], we hypothesize that the skill level and the preferences of a quadcopter pilot are encoded in a cost functional. We then model the pilot-aircraft system as an optimal control problem and aim to recover the said cost functional using flight logs that record the commands of the pilot and the resulting trajectories of the quadcopter.

The authors are with the School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK, USA. {jared.town, zachmor, rushikesh.kamalapurkar}@okstate.edu.
This research was supported, in part, by the National Science Foundation (NSF) under award numbers 1925147 and 2027999 and the Air Force Office of Scientific Research under award number FA9550-20-1-0127. 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.

Inverse reinforcement learning (IRL) is a popular tool for obtaining the cost functional of an expert by measuring their input commands and the resulting behavior of the controlled system. IRL methods such as [4]–[17] are developed under the assumption that the decisions of the said expert are optimal or near-optimal with respect to the unknown cost functional. A general characteristic of such methods is that they require multiple trajectories and are computationally complex, making them unsuitable for online, real-time implementation. To address the IRL problem in a real-time and online setting, methods such as [18]–[21] have been developed. These methods are typically model-based and use a single continuous trajectory to learn the cost functional of an expert. A notable result is obtained in [22], where an online and model-free approach is developed that utilizes a neural network to solve the IRL problem in the presence of adversarial . However, this method only identifies the state penalty matrix and is unable to identify the control penalty matrix.

This paper is focused on the development of an IRL formulation of the pilot modeling problem and an adaptation of the of the regularized history stack observer (RHSO) developed in [23] to solve the resulting IRL problem. It is shown in [24] that IRL problems that have a product structure have multiple linearly independent solutions. Since the linearized model of a quadcopter decouples lateral and longitudinal dynamics, it has a product structure. As a result, implementation of IRL to estimate cost functionals of quadcopter pilots requires methods such as [23] that are suited for IRL problems with multiple linearly independent solutions.

The method developed in [23] is an online IRL method that is capable of identifying the true cost functional of the pilot, up to a scaling factor, if the IRL problem has a unique (up to a scaling factor) solution, and an equivalent solution (that is, a cost functional that results in the same feedback matrix as the expert), if the IRL problem admits multiple linearly independent solutions. The key contribution of this paper is a reformulation of the pilot behavior modeling problem in the framework of IRL, where the control inputs of the pilot are velocity commands that are executed by an onboard autopilot. The reformulation allows for the use of the IRL method developed in [23], with minimal modification, to estimate a cost functional that models the performance of the pilot.

II. Inverse Reinforcement Learning

This section describes the IRL algorithm used to estimate a cost functional that is equivalent to the cost functional of the pilot. The algorithm is similar to [23], with minor modifications to account for availability of full state measurements.

The pilot-controlled system is assumed to be a linear time-invariant system of the form

$$\dot{X}(t) = AX + BU, \tag{1}$$

where $X \in \mathbb{R}^{12}$ is the state, $U \in \mathbb{R}^4$ is the control input, $A \in \mathbb{R}^{12\times 12}$ is the system matrix and $B \in \mathbb{R}^{12\times 4}$ is the control effectiveness matrix. Motivated by [1]–[3], the pilot is assumed to employ an optimal controller that minimizes the cost functional

$$J(X_0, U(\cdot)) = \int_0^\infty \left(X(t)^\top Q X(t) + U(t)^\top R U(t)\right) dt, \tag{2}$$

where $X(\cdot)$ denotes the system trajectory under the control signal $U(\cdot)$, starting from the initial condition X_0, $Q \in \mathbb{R}^{12\times 12}$ is an unknown positive semidefinite matrix, and $R \in \mathbb{R}^{4\times 4}$ is an unknown positive definite matrix.

Assumption 1. The pair (A, B) is stabilizable and $(A, \sqrt{Q})$ is detectable.

Stabilizability of (A, B) and detectability of $(A, \sqrt{Q})$ is needed for the optimal controller to exist. Linearized models of quadrotors, including the one used in the experiments presented in Section IV, are stabilizable. In particular, the Popov–Belevitch–Hautus (PBH) test in Theorem 14.3 of [25], can be used to show that the pilot model developed in Section III satisfies the stabilizability condition in Assumption 1. In the experiments, the pilot is assumed to penalize translational position errors and heading errors, and the resulting pair $(A, \sqrt{Q})$ is shown to satisfy the detectability condition.

The algebraic Riccati equation (ARE),

$$A^\top S + SA - SBR^{-1}B^\top S + Q = 0, \tag{3}$$

of the optimal control problem described by (1) and (2) can be solved to yield the optimal policy of the pilot, given by $u = -K_{EP}x$, where $K_{EP} = R^{-1}B^\top S$. The objective is to estimate the unknown matrices Q and R online and in real-time using the known system matrices, A and B, and measurements of X and U. The IRL problem, as formulated above, is ill-posed in general. That is, given A, B, and measurements of X and U, there can be infinitely many linearly independent triplets (Q, R, S), with respect to which the measured state and control signals are optimal. In particular, the linear system in the pilot modeling problem (see (24)) is comprised of two decoupled systems. If the penalty matrices Q and R are also decoupled, for example, diagonal, then the corresponding IRL problem can be shown to admit multiple linearly independent solutions [24].

To formulate a well-posed problem, an equivalent solution is sought according to the following definition.

Definition 1. ([23]) Given $\varpi \geq 0$, a triplet $(\hat{Q}, \hat{S}, \hat{R})$ is called an $\varpi-$equivalent solution of the IRL problem if

$$\left\| A^\top \hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^\top \hat{S} + \hat{Q} \right\| \leq \varpi,$$

and optimization of the performance index J, with $Q = \hat{Q}$ and $R = \hat{R}$, results in a feedback matrix, $\hat{K}_p := \hat{R}^{-1}B^\top \hat{S}$, that satisfies

$$\left\| \hat{K}_p - K_{EP} \right\| \leq \varpi.$$

A. The Regularized History Stack Observer

The following development is a special case of the RHSO developed in [23], where the system state is measurable. In the experiment, state estimates generated by an onboard Kalman filter are utilized.

If Assumption 1 is met and if the state and control trajectories, $X(\cdot)$ and $U(\cdot)$ respectively, of the quadcopter, are optimal with respect to the cost functional in (2), then there exists a matrix S such that the matrices Q, R, A, B, and S satisfy the Hamilton-Jacobi-Bellman (HJB) equation

$$X^\top(t)\left(A^\top S + SA - SBR^{-1}B^\top S + Q\right)X(t) = 0, \tag{4}$$

and the optimal control equation

$$U(t) = -R^{-1}B^\top S X(t), \tag{5}$$

for all $t \in \mathbb{R}_{\geq 0}$.

Given measurements of the the state, X, and control signal, U, and estimates $\hat{Q}$, $\hat{R}$, and $\hat{S}$ of Q, R, and S, respectively, (4) and (5) can be used to develop an equivalence metric that evaluates to zero if the estimates constitute an equivalent solution. Furthermore, if a collection of measurements of X and U meet the data-sufficiency conditions outlined in Definition 2, then satisfaction of (4) and (5) for all measurements can be shown to result in a $\varpi-$equivalent solution with $\varpi = 0$ (see the proof of Corollary 1).

Since scaling of a cost functional results in another equivalent cost functional, equivalent cost functionals can only be identified up to a scaling factor. To fix the scale, the (1,1) element of $\hat{R}$, denoted by r_1, is selected to be equal to one. In particular, the RHSO generates an equivalent solution $(\hat{Q}, \hat{R}, \hat{S})$ of the IRL problem using

$$\dot{\hat{W}} = K_W \Sigma^\top \left(\Sigma_u - \Sigma \hat{W}\right), \tag{6}$$

where K_W is a symmetric positive definite learning gain matrix. To facilitate a comparison with the HSO in [18], we select $K_W = (\Sigma^\top\Sigma + \epsilon I)^{-1}$. When $\epsilon = 0$, the RHSO reduces to the HSO. In (6), $\hat{W} = \left[\hat{W}_S^\top, \quad \hat{W}_Q^\top, \quad (\hat{W}_R^-)^\top\right]^\top$, where $\hat{W}_S \in \mathbb{R}^{78}$, $\hat{W}_Q \in \mathbb{R}^{78}$, and $\hat{W}_R \in \mathbb{R}^{10}$ are weights that satisfy $(\hat{W}_S)^\top \sigma_S(X) = X^\top \hat{S} X$, $(\hat{W}_Q)^\top \sigma_Q(X) = X^\top \hat{Q} X$, $(\hat{W}_R)^\top \sigma_{R1}(U) = U^\top \hat{R} U$, and $\sigma_{R2}(U)\hat{W}_R = \hat{R}U$, respectively, and the vector $\hat{W}_R^-$ is a copy of $\hat{W}_R$ with the first element, r_1, removed. The basis functions are given by

$$\sigma_S(X) = \sigma_Q(X) := [X_1^2, 2X_1X_2, 2X_1X_3, \ldots, 2X_1X_{12}, X_2^2, 2X_2X_3, 2X_2X_4, \ldots, X_{11}^2, \ldots, 2X_{11}X_{12}, X_{12}^2]^\top, \tag{7}$$

$$\sigma_{R1}(U) := [U_1^2, 2U_1U_2, 2U_1U_3, 2U_1U_4, U_2^2, 2U_2U_3, 2U_2U_4, U_3^2, 2U_3U_4, U_4^2]^\top, \tag{8}$$

and

$$\sigma_{R2}(U) = \begin{bmatrix} U^\top & 0_{1\times 3} & 0_{1\times 2} & 0 \\ U_1 e_{2,4} & \left(U^\top\right)^{(-1)} & 0_{1\times 2} & 0 \\ U_1 e_{3,4} & U_2 e_{2,3} & \left(U^\top\right)^{(-2)} & 0 \\ U_1 e_{4,4} & U_2 e_{3,3} & U_3 e_{2,2} & U_4 \end{bmatrix}, \tag{9}$$

where $U^{(-j)}$ denotes the vector U with the first j elements removed, and $e_{i,j}$ denotes a row vector of size j, with a one in the $i-$th position and zeros everywhere else. The matrices $\Sigma \in \mathbb{R}^{165N\times 165}$ and $\Sigma_u \in \mathbb{R}^{165N}$, referred to collectively as *the history stack*, are constructed as

$$\Sigma \coloneqq \begin{bmatrix} \sigma_\delta\left(X(t_1),U(t_1)\right) \\ \sigma_{\Delta_u}\left(X(t_1),U(t_1)\right) \\ \vdots \\ \sigma_\delta\left(X(t_N),U(t_N)\right) \\ \sigma_{\Delta_u}\left(X(t_N),U(t_N)\right) \end{bmatrix}, \quad \Sigma_u \coloneqq \begin{bmatrix} -U_1^2(t_1)r_1 \\ -2U_1(t_1)r_1 \\ 0_{m-1\times 1} \\ \vdots \\ -U_1^2(t_N)r_1 \\ -2U_1(t_N)r_1 \\ 0_{m-1\times 1} \end{bmatrix},$$

where N is the number of time instances selected for storage and the functions σ_δ and σ_{Δ_u} are given by

$$\sigma_\delta\left(X,U\right) = \left[(AX+BU)^\top(\nabla_X\sigma_S(X))^\top \quad (\sigma_Q(X))^\top \quad (\sigma_{R1}^-(U))^\top\right] \tag{10}$$

and

$$\sigma_{\Delta_u}\left(X,U\right) = \left[B^\top(\nabla_x\sigma_S(X))^\top \quad 0_{4\times 78} \quad 2\sigma_{R2}^-(U)\right], \tag{11}$$

where σ_{R2}^- is a copy of σ_{R2} with the first column removed and σ_{R1}^- is a copy of σ_{R1} with the first element removed.

Corollary 1 below, which guarantees convergence of (6) to an equivalent solution, relies on the error metric $\Delta \coloneqq \Sigma_u - \Sigma\hat{W}$ and its time derivative

$$\dot{\Delta} = -\Sigma K_W \Sigma^\top \Delta, \tag{12}$$

along with the following data informativity condition adopted from [23].

Definition 2. The signal (X,U) is called finitely informative (FI) if there exists a time instance $\underline{T} > 0$ such that for some $\{t_1,t_2,\ldots,t_N\} \subset [0,\underline{T}]$,

$$\text{Span}\left\{X(t_i)\right\}_{i=1}^N = \mathbb{R}^n,$$

$$\text{Span}\left\{X(t_i)X(t_i)^\top\right\}_{i=1}^N = \{\mathbb{Z}\in\mathbb{R}^{n\times n} | \mathbb{Z}=\mathbb{Z}^\top\}, \quad \text{and}$$

$$\Sigma_u \in \text{Range}(\Sigma). \tag{13}$$

In addition, for a given $\epsilon > 0$, if $\min\{\text{eig}(\chi\chi^\top)\} > \epsilon$ and $\min\{\text{eig}(ZZ^\top)\} > \epsilon$, where $\chi \coloneqq [X(t_1),\ldots,X(t_N)]$, $Z \coloneqq [\text{uvec}(X(t_1)X^\top(t_1)),\ldots,\text{uvec}(X(t_N)X^\top(t_N))] \in \mathbb{R}^{\frac{n(n+1)}{2}\times N}$, and $\text{uvec}(X(t_i)X^\top(t_i)) \in \mathbb{R}^{\frac{n(n+1)}{2}}$ denotes vectorization of the upper triangular elements of the symmetric matrix $X(t_i)X^\top(t_i) \in \mathbb{R}^{n\times n}$, then (X,U) is called $\epsilon-$*finitely informative (FI)*.

To implement the developed observer, a method to select the time instances $t_1,\ldots,t_N$ is needed. The convergence result summarized in Corollary 1 relies on the existence of a time instance $\underline{T} \geq 0$ such that the three conditions in Definition 2 are met. As such, any data selection algorithm that ensures the satisfaction of those three conditions can be used to implement the developed observer. In this paper, a data selection method that minimizes the condition number of $K_W = \Sigma^\top\Sigma + \epsilon I$ is utilized. Minimization of the condition number of $\Sigma^\top\Sigma + \epsilon I$ improves the accuracy of matrix inversion in the update law (6) and improves the convergence rate of Δ in (12).

The matrices Σ_u and Σ contained in the history stack are recorded at specific time instances according to the following procedure. Both matrices are initialized as zero matrices. Data are then added to the matrices at a user-selected sampling interval until they are filled. Then, a condition number minimization algorithm similar to [26] is used to replace old data with new data, where replacement is carried out only if the post-replacement condition number of $\Sigma^\top\Sigma+\epsilon I$ is lower than its pre-replacement condition number. Due to the replacement procedure, the time instances t_i corresponding to data stored in the history stack are piecewise constant functions of time.

Corollary 1. *If the state and control signals $X(\cdot)$ and $U(\cdot)$ are $\epsilon-$finitely informative, for some $\epsilon > 0$ and if there exist a constant $0 \leq \underline{R} < \infty$ such that the matrix $\hat{R}(t)$, extracted from $\hat{W}(t)$, is invertible with $\|\hat{R}^{-1}(t)\| \leq \underline{R}$ for all $t \geq \underline{T}$, where $\underline{T}$ is the time instance introduced in Definition 2, then the matrices $\hat{Q}$, $\hat{S}$, and $\hat{R}$, extracted from $\hat{W}$, converge to a $0-$equivalent solution of the IRL problem.*

Proof. The proof, included here for completeness, is a slight modification of the proof of Theorem 7 and Theorem 10 of [23]. Applying Theorem 7 in [23] with $K_4 = I$ it can be concluded that along the solutions of (6), $\lim_{t\to\infty}\Delta(t) = 0$. Note that the error metric Δ can be expressed using the basis functions in (9), (10), and (11) as

$$\Delta = \begin{bmatrix} \sigma'_\delta\left(X(t_1(t)),U(t_1(t))\right) \\ \sigma'_{\Delta_u}\left(X(t_1(t)),U(t_1(t))\right) \\ \vdots \\ \sigma'_\delta\left(X(t_N(t)),U(t_N(t))\right) \\ \sigma'_{\Delta_u}\left(X(t_N(t)),U(t_N(t))\right) \end{bmatrix} \hat{W}',$$

where $\hat{W}' \coloneqq \begin{bmatrix}\hat{W}_S^\top & \hat{W}_Q^\top & \hat{W}_R^\top\end{bmatrix}^\top$,

$$\sigma'_{\Delta_u}(X,U) = \begin{bmatrix} B^\top\left(\nabla_x\sigma_S\left(X\right)\right)^\top & 0_{4\times 78} & 2\sigma_{R2}(U)\end{bmatrix},$$

and

$$\sigma'_\delta(X,U) = \begin{bmatrix}(AX+BU)^\top\left(\nabla_X\sigma_S\left(X\right)\right)^\top & \left(\sigma_Q\left(X\right)\right)^\top & \left(\sigma_{R1}\left(U\right)\right)^\top\end{bmatrix}.$$

Using the fact that $\sigma'_{\Delta_u}(X(t_i(t)),U(t_i(t)))\hat{W}'(t) = \hat{R}(t)\tilde{K}_P(t)X(t_i(t))$, where $\tilde{K}_P(t) \coloneqq \hat{K}_p(t) - K_{EP}$, it can be concluded that

$$\left\|\tilde{K}_P(t)X(t_i(t))\right\| \leq \left\|\hat{R}^{-1}(t)\sigma'_{\Delta_u}\left(X(t_i(t)),U(t_i(t))\right)\hat{W}'(t)\right\|. \tag{14}$$

Given any $\varpi > 0$, if $\min\{\text{eig}(\chi(t)\chi(t)^\top)\} > \epsilon$ then there exists $c > 0$, independent of t, such that $\left\|\tilde{K}_P(t)X(t_i(t))\right\| \leq \frac{\varpi}{c}$, for all $i = 1,\ldots,N$, implies $\left\|\tilde{K}_P(t)\right\| \leq \varpi$. Select $\overline{T}$ large enough such that for all $t \geq \overline{T}$, $\|\Delta(t)\| \leq \frac{\varpi}{2c\underline{R}}$. Then, for all $i = 1,\ldots,N$, $\left\|\sigma'_{\Delta_u}\left(X(t_i(t)),U(t_i(t))\right)\hat{W}'(t)\right\| \leq \frac{\varpi}{c\underline{R}}$, which

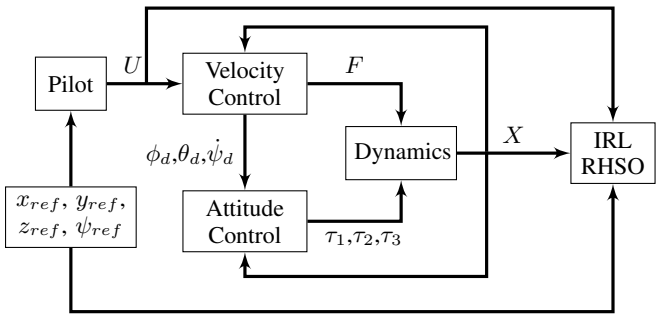

Fig. 1. A block diagram that summarizes the developed RHSO framework for pilot behavior modeling.

implies $\left\|\hat{R}^{-1}(t)\sigma'_{\Delta_u}\left(X(t_i(t)),U(t_i(t))\right)\hat{W}'(t)\right\| \leq \frac{\varpi}{2c}$. From (14), it follows that $\left\|\tilde{K}_P(t)X(t_i(t))\right\| \leq \frac{\varpi}{c}$, and as a result, $\left\|\tilde{K}_P(t)\right\| \leq \varpi$. Since ϖ was arbitrary, $\lim_{t\to\infty}\hat{K}_p(t) = K_{EP}$.

The function σ'_δ can be expressed as

$$\sigma'_\delta\left(X(t_i(t)),U(t_i(t))\right)\hat{W}'(t) = X^\top(t_i(t))\hat{M}X(t_i(t)) + g\left(\hat{K}_p(t),K_{EP}\right), \quad (15)$$

where the function g satisfies[1] $g = O\left(\left\|\tilde{K}_P(t)\right\|\right)$ and

$$\hat{M}(t) = \left(A^\top\hat{S}(t) + \hat{S}(t)A - \hat{S}(t)B\hat{R}^{-1}(t)B^\top\hat{S}(t) + \hat{Q}(t)\right).$$

Using the triangle inequality,

$$\left|X^\top(t_i(t))\hat{M}(t)X(t_i(t))\right| \leq \left|\sigma'_\delta\left(X(t_i(t)),U(t_i(t))\right)\hat{W}'\right| + \left|g\left(\hat{K}_p(t),K_{EP}\right)\right| \quad (16)$$

Since $\lim_{t\to\infty}\hat{K}_p(t) = K_{EP}$, $\lim_{t\to\infty}\Delta(t) = 0$, and $\left|\sigma'_\delta\left(X(t_i(t)),U(t_i(t))\right)\hat{W}'\right| \leq \|\Delta(t)\|$, given any $\varepsilon > 0$, the bound in (16) implies that there exists $\overline{T} \geq 0$ such that for all $t \geq \overline{T}$ and for all $i = 1,\cdots,N$, $\left|X^\top(t_i(t))\hat{M}(t)X(t_i(t))\right| \leq \varepsilon$.

Similar to the proof of Corollary 10 in [23], if $\min\{\text{eig}(Z(t)(Z(t))^\top)\} > \epsilon, \forall t \geq \underline{T}$, then given $\varpi > 0$, one can construct a $\varepsilon > 0$ such that $\left|X^\top(t_i(t))\hat{M}(t)X(t_i(t))\right| \leq \varepsilon$, for all $i = 1,\cdots,N$, implies that $\left\|\hat{M}(t)\right\| \leq \varpi$. Therefore, $\lim_{t\to\infty}\left\|\hat{M}(t)\right\| = 0$, which completes the proof of the corollary. □

III. Formulation of the Pilot Performance Modeling Problem in an IRL Framework

A. Problem Statement

This study concerns a quadcopter sUAS with an onboard autopilot being flown by a human pilot via desired velocity commands. That is, from the perspective of the human pilot, the control input is the desired linear velocities of the quadcopter and the desired yaw rate. The human pilot is asked to regulate the aircraft to the origin, starting from a non-zero initial condition. The objective is to find a best-fit cost functional such that a controller that optimizes the cost functional results in trajectories that are similar to those observed under human control.

In this proof-of-concept study, we assume that the human pilot can observe the full state of the quadcopter and the experimental study utilizes supervisory LQR controllers as surrogates in lieu of human pilots. The control commands sent to the aircraft by the LQR surrogates, along with the full state of the quadcopter, are used to learn the cost functional of the surrogate pilot using the RHSO. Since the IRL problem formulated in this section admits multiple solutions, we seek an equivalent cost functional, according to Definition 1.

B. Quadcopter Model

To implement the developed model-based IRL method, a linearized quadcopter model, with velocity commands as the input, and the actual position, velocity, orientation, and angular velocity as the output needs to be developed. Such a model depends on the autopilot being used to stabilize the aircraft, and as such, knowledge of the autopilot algorithm is required to complete the model. Note that identification of the autopilot is not the focus of this study. We assume that the autopilot is able to track the commanded velocities and aim to model the cost functional of the surrogate LQR pilot that generates velocity commands.

The model used in this study closely follows the development in [27]–[29]. The state variables of the model are

$$X \coloneqq \left[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\right]^\top,$$

where x, y, and z are the translational positions, $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the translational velocities, ϕ, θ, and ψ are the roll, pitch, and yaw angular positions, respectively, and $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ are the roll, pitch, and yaw rates, respectively. The control input is given by

$$U \coloneqq \left[\dot{x}_d, \dot{y}_d, \dot{z}_d, \dot{\psi}_d\right]^\top,$$

where $\dot{x}_d$, $\dot{y}_d$, and $\dot{z}_d$ are the desired translational velocities and $\dot{\psi}_d$ as the desired yaw rate. The translational dynamics of a quadcopter are described in the North, East, Down (NED) coordinate frame by [27]

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\0\\mg\end{bmatrix} + R_M\begin{bmatrix}0\\0\\-F\end{bmatrix} - k_t\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}, \quad (17)$$

where k_t is the aerodynamic drag, m is the mass, g is the acceleration due to gravity, and R_M is the rotational matrix where small angle approximations result in

$$R_M = \begin{bmatrix}1 & \phi\theta - \psi & \theta + \phi\psi\\ \psi & \phi\theta\psi + 1 & \theta\psi - \phi\\ -\theta & \phi & 1\end{bmatrix}. \quad (18)$$

The thrust, F, applied by the autopilot is a proportional controller

$$F = mg + mk_{p_{13}}(\dot{z} - \dot{z}_d). \quad (19)$$

[1]For a positive function g, $f = O(g)$ if there exists a constant M such that $\|f(x)\| \leq Mg(x), \forall x$.

The rotational motion of the quadcopter is described by [28], [29]

$$\begin{aligned}\ddot{\phi}I_{xx} &= \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + l\tau_1,\\ \ddot{\theta}I_{yy} &= \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + l\tau_2,\\ \ddot{\psi}I_{zz} &= \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + \tau_3,\end{aligned} \tag{20}$$

where I_{xx}, I_{yy}, and I_{zz} are moments of inertia and τ_1, τ_2, and τ_3 are torques designed as

$$\begin{aligned}\tau_1 &= k_{p_{21}}(\phi_d - \phi) - k_{d_1}\dot{\phi},\\ \tau_2 &= k_{p_{22}}(\theta_d - \theta) - k_{d_2}\dot{\theta},\\ \tau_3 &= k_{d_3}(\dot{\psi}_d - \dot{\psi}).\end{aligned} \tag{21}$$

The desired angles ϕ_d and θ_d commanded by the autopilot are given by

$$\begin{bmatrix}\theta_d\\ \phi_d\end{bmatrix} = \begin{bmatrix}\arctan\left(\frac{k_{p_{12}}(\dot{y}_d-\dot{y})\sin\psi + k_{p_{11}}(\dot{x}_d-\dot{x})\cos\psi}{g+k_{p_{13}}(\dot{z}_d-\dot{z})}\right)\\ \arctan\left(\cos\theta_d\frac{k_{p_{11}}(\dot{x}_d-\dot{x})\sin\psi - k_{p_{12}}(\dot{y}_d-\dot{y})\cos\psi}{g+k_{p_{13}}(\dot{z}_d-\dot{z})}\right)\end{bmatrix}, \tag{22}$$

where $k_{p_{11}}$, $k_{p_{12}}$, $k_{p_{13}}$, $k_{p_{21}}$, $k_{p_{22}}$, k_{d_1}, k_{d_2}, k_{d_3} are control gains of the autopilot. The desired angles are simplified using the small angle approximation and a linear approximation of the inverse tangent function [30] to yield

$$\begin{aligned}\theta_d &= \frac{\pi}{4}\left(\frac{k_{p_{12}}(\dot{y}_d - \dot{y})\psi + k_{p_{11}}(\dot{x}_d - \dot{x})}{g + k_{p_{13}}(\dot{z}_d - \dot{z})}\right),\\ \phi_d &= \frac{\pi}{4}\left(\frac{k_{p_{11}}(\dot{x}_d - \dot{x})\psi - k_{p_{12}}(\dot{y}_d - \dot{y})}{g + k_{p_{13}}(\dot{z}_d - \dot{z})}\right).\end{aligned} \tag{23}$$

Linearizing (17) and (20) about the origin, while using (19), (21), and (23), yields the linear system

$$\begin{aligned}\ddot{x} &= -g\theta - \frac{k_t}{m}\dot{x},\\ \ddot{y} &= g\phi - \frac{k_t}{m}\dot{y},\\ \ddot{z} &= k_{p_{13}}(\dot{z}_d - \dot{z}) - \frac{k_t}{m}\dot{z},\\ \ddot{\phi} &= \frac{b_1\pi k_{p_{21}}k_{p_{12}}(\dot{y} - \dot{y}_d)}{4g} - b_1k_{d_1}\dot{\phi} - b_1k_{p_{21}}\phi,\\ \ddot{\theta} &= \frac{b_2\pi k_{p_{22}}k_{p_{11}}(\dot{x}_d - \dot{x})}{4g} - b_2k_{d_2}\dot{\theta} - b_2k_{p_{22}}\theta,\\ \ddot{\psi}, &= b_3k_{d_3}(\dot{\psi}_d - \dot{\psi}),\end{aligned} \tag{24}$$

where $b_1 = \frac{l}{I_{xx}}$, $b_2 = \frac{l}{I_{yy}}$, and $b_3 = \frac{1}{I_{zz}}$, and l is the length of the quadcopter arm.

As shown in Figure 1, given measurements of the state variables, i.e., translational positions $[x, y, z]$, translational velocities $[\dot{x}, \dot{y}, \dot{z}]$, angular positions $[\phi, \theta, \psi]$, angular velocities $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$, and the control variables, i.e., the desired velocities $[\dot{x}_d, \dot{y}_d, \dot{z}_d]$ and yaw rate $[\dot{\psi}_d]$ commanded by the surrogate LQR pilot, we aim to find an equivalent solution ($\hat{Q}$, $\hat{S}$, $\hat{R}$) of the IRL problem according to Definition 1. The developed RHSO algorithm for IRL is summarized in Algorithm 1.

Algorithm 1 One time step of the regularized history stack observer algorithm. In the algorithm, Σ^i and Σ_u^i denote the $i-$th block of 165 rows of Σ and Σ_u, respectively, $s(t) = \begin{bmatrix}\sigma_\delta\left(X(t), U(t)\right)\\ \sigma_{\Delta_u}\left(X(t), U(t)\right)\end{bmatrix}$, and $s_u(t) = \begin{bmatrix}-U_1^2(t)r_1\\ -2U_1(t)r_1\\ 0_{m-1\times 1}\end{bmatrix}$

Input: State and input measurements $X(t)$ and $U(t)$ at time t, system matrices A and B, estimate $\hat{W}$ of the weights and history stacks Σ and Σ_u from the previous time t^-

Output: Updated history stacks Σ and Σ_u and updated weight estimate $\hat{W}$

$i \leftarrow 1$
$j \leftarrow 0$
$\underline{\sigma} \leftarrow \min \text{eig}(\Sigma\Sigma^\top + \epsilon I)$
while $i \leq N$ **do**
  $\Sigma' = \Sigma^\top\Sigma - (\Sigma^i)^\top\Sigma^i + s(t)^\top s(t) + \epsilon I$
  if $\min \text{eig}(\Sigma') > \underline{\sigma}$ **then**
    $\underline{\sigma} \leftarrow \min \text{eig}(\Sigma')$
    $j \leftarrow i$
  end if
end while
if $j \neq 0$ **then**
  $\Sigma^j \leftarrow s(t)$
  $\Sigma_u^j \leftarrow s_u(t)$
end if
$\hat{W} \leftarrow (\Sigma^\top\Sigma + \epsilon I)^{-1}\Sigma^\top\left(\Sigma_u - \Sigma\hat{W}(t^-)\right)$
$\hat{W} \leftarrow \hat{W} + (t - t^-)\dot{\hat{W}}$

IV. Experiments

Experimental results obtained using the developed RHSO, implemented on a quadcopter, are presented in this section. The pilot is assumed to be a surrogate LQR controller that mimics velocity commands sent by a remote controller to a quadcopter. The velocity commands are treated as desired velocities that are executed by the onboard autopilot. The pilot behavior modeling problem is reformulated as an IRL problem and the ability of the developed IRL method to learn an equivalent solution of the IRL problem using measurements of the quadcopter state and the velocity commands sent by the surrogate LQR controller is demonstrated.

A. Hardware

A custom-built quadcopter using the PX4 flight stack is utilized for the experiments. The drone frame is built using a XILO Phreakstyle Freestyle frame kit, the flight control unit is a Holybro Kakute H7 that is connected to a ground control station through WiFi. The position and the orientation of the quadcopter is captured through a motion capture system (OptiTrack) and the angular velocity and the acceleration are measured from an onboard inertial measurement unit (IMU). Data from both sensors are fused using a Kalman filter to generate estimates of the state of the qudcopter. The model parameters for this setup are $l = 0.107642$ m, $I_{xx} = 0.002261$ kg m^2, $I_{yy} = 0.002824$ kg m^2, $I_{zz} = 0.002097$ kg m^2, $k_t = 0.01$, $g = 9.81$ m/s^2, $m = 0.579902$ kg, $k_{p_{11}} = -5.25$, $k_{p_{12}} = -5.25$, $k_{p_{13}} = 3$, $k_{p_{21}} = 3.5$, $k_{p_{22}} = 3.5$, $k_{p_{23}} = 0.35$, $k_{d_1} = 0.4$, $k_{d_2} = 0.4$, and $k_{d_3} = 0.1$.

To demonstrate the applicability of the developed framework to typical quadcopter deployment scenarios where the autopilot is proprietary and unknown, this experiment utilizes

the default PX4 autopilot, which is different from the autopilot employed in the model (i.e., (22)). While the PX4 autopilot is able to track the velocity inputs sent by the surrogate pilot, the performance of the real quadcopter employing the PX4 autopilot is substantially different from the performance of a simulated quadcopter employing the autopilot in (22).

To ensure that the closed-loop model presented in Section III fits the closed loop model of real quadcopter, the proportional and derivative gains in (22) are manually adjusted so that the trajectories of the model in Section III, employing the autopilot in (22), and the real quadcopter, employing the default PX4 autopilot, under velocity commands sent by the surrogate LQR pilot are as close to each other as possible.

B. Controller Implementation

The quadcopter is controlled via an off-board ground control station that implements the surrogate LQR pilot. The objective of the pilot is to return the quadrotor to the origin starting from a given known initial condition using velocity and yaw rate commands. The surrogate pilot implements the control policy that optimizes the cost functional in (2), assuming the linear closed-loop quadrotor model given in (24), with[2]

$$Q = \mathrm{diag}([9.57, 6.91, 2.84, 0, 0, 0, 0, 0, 11.68, 0, 0, 0]) \text{ and}$$
$$R = \mathrm{diag}([9.57, 3.48, 14.40, 0.17]). \tag{25}$$

The pairs (A, B) and $(A, \sqrt{Q})$ are confirmed to satisfy the stabilizability and detectability conditions in Assumption 1 using PBH tests in Theorems 14.3 and 16.6 in [25], respectively.

The cost functional is designed under the assumption that the surrogate LQR pilot only penalizes the state variables corresponding to the translational position and the heading. To reduce the number of unknown parameters, the sparsity structure of Q and R is assumed to be known and only the nonzero elements of Q and R are estimated. As a result, the number of unknown parameters in Q is reduced from 78 to 4 and the number of unknown parameters in R is reduced from 9 to 3, resulting in a total of 85 unknown parameters.

To satisfy the FI condition in Definition 2, the ground control station adds an excitation signal onto the velocity commands generated by the surrogate pilot before they are sent to the autopilot. As a result, the final commanded velocity is

$$U_{cmd} = U_{exc} + U, \tag{26}$$

where $U = -K_{EP}X$ is the command generated by the surrogate pilot and U_{exc} is the excitation signal. It is assumed that the excitation signal is known, and as a result, the true velocity commands generated by the surrogate pilot are also known to the learner. The fact that such excitation signals are commonly utilized in popular drone software packages during an auto-tune process for PID controllers [31] motivates their use in this work.

C. Methods

A total of 13 repeated trials are performed to gauge the performance of the developed IRL technique. In each of the 13 experiments, the quadcopter is started at a randomly generated hover point contained within the operating area. The surrogate LQR pilot then commands the quadcopter to fly to the origin with a z-offset equal to the desired flight height. To ensure that the measured costs are representative of the infinite horizon cost, the controller is run for a time horizon of 200 s, which is more than 4 times the observed time constant of the surrogate LQR controller. The excitation signal U_{exc} is composed of 4 sets of 75 sinusoidal signals. Each set spans a frequency range from 0.001 Hz to 10 Hz, with a varying frequency and a magnitude of 0.03.

Since the regressor Σ is a nonlinear function of the states, relationships between persistence of excitation, number of frequencies in the excitation signal, and number of unknown parameters, well-established in linear systems theory, do not apply to this problem. Using the sufficient conditions developed for linear regressors as a heuristic guideline, the number of frequencies in the excitation signal is initially selected to be roughly equal to the number of unknown parameters, and tuned using trial and error. The magnitude of the excitation signal is also selected using trial and error in simulation. A larger magnitude excitation signal typically results in a smaller condition number of $\Sigma^\top\Sigma + \epsilon I$. However, larger excitation magnitudes result in longer quadcopter trajectories, which require a larger flight arena. The excitation signal selected above was tuned using a quadcopter simulator to ensure a sufficiently small condition number for $\Sigma^\top\Sigma + \epsilon I$ while keeping the quadcopter confined within the flight arena available in the laboratory.

The RHSO is implemented with regularization parameter $\epsilon = 0.002$, and data are collected at a sampling rate of 0.08 seconds using the condition number minimization algorithm described in Section III. The initial guesses for the unknown weights are randomly generated to be normally distributed in the interval $[-5, 5]$.

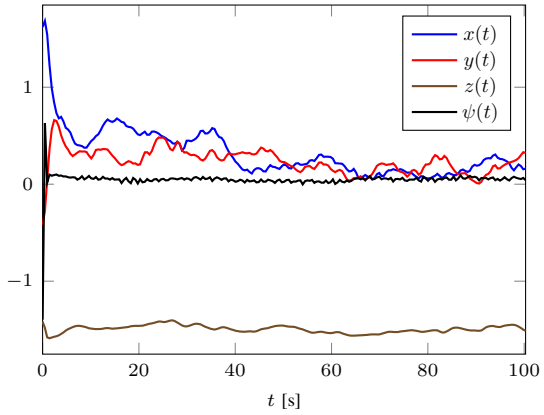

Fig. 2. Position and heading of the quadcopter in one of the 13 experiments.

D. Results and Discussion

The experimental results obtained from one of the 13 flight tests are shown in Figs. 2-6. The position of the quadcopter as a function of time is shown in Fig. 2, and the linear velocity of the quadcopter as a function of time is shown in Fig. 3.

[2]The notation $\mathrm{diag}(v)$ represents a diagonal matrix with the elements of the vector v along the diagonal.

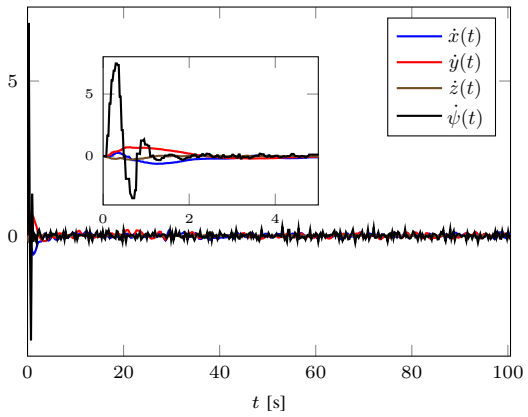

Fig. 3. Linear velocity and yaw rate of the quadcopter in one of the 13 experiments.

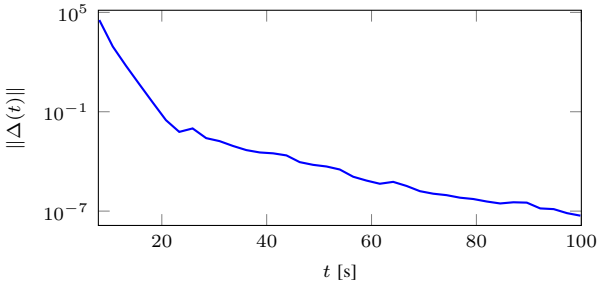

Fig. 4. A logscale plot of the norm of Δ as a function of time in one of the 13 experiments.

The quadcopter holds position at the origin with a z-offset of 1.5 m and the velocity appears noisy due to the excitation signal. The convergence of Δ to zero (Fig. 4)[3], combined with the convergence of $\hat{K}_p$ to K_{EP} (Fig. 5) indicates that the developed technique is able to obtain an equivalent solution (per Definition 1) to the IRL problem. The experimental results are thus consistent with Corollary 1.

Figs. 4 and 5 demonstrate that while the feedback policy of the surrogate LQR pilot is estimated correctly, the estimated cost functional is substantially different from the cost functional of the surrogate LQR pilot. This behavior is expected because the underlying IRL problem has multiple equivalent solutions. As indicated by Fig. 7, the cost functional recovered from the data in each of the 13 experiments converges to different equivalent solutions. The particular equivalent solution recovered in each run depends on the initial guess of the unknown weights used in that run. From the 13 experiments, it is evident that the RHSO finds equivalent solutions for the pilot modeling problem. It is observed in Table I that the convergence of the estimated solution to an equivalent solution is much faster in the quadcopter pilot modeling application than the simulation results presented in [23]. We postulate that the faster convergence can be attributed to the added excitation signal increasing the information content of the data. Furthermore, as evidenced by Table I, the original history stack

[3]The notation $\|\cdot\|$ is used to denote the euclidean norm when applied to a vector and the Frobenius norm when applied to a matrix.

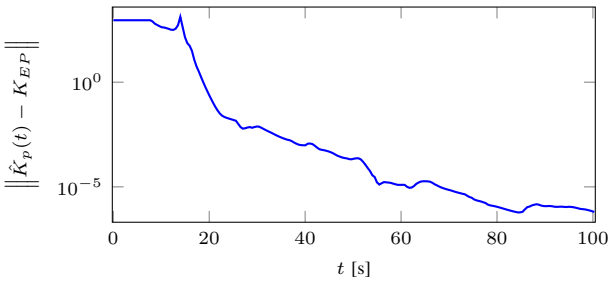

Fig. 5. A logscale plot of the induced 2−norm of the error between the estimated feedback gain and the surrogate pilot's feedback gain as a function of time in one of the 13 experiments.

	RHSO	HSO
Mean$\left(\|K_{EP} - \hat{K}_p\|\right)$	2.6997e-08	NaN
Cov$\left(\|K_{EP} - \hat{K}_p\|\right)$	8.3316e-15	NaN

TABLE I
THE RHSO AND THE HSO [18] ARE EVALUATED BY COMPARING THE MEAN AND COVARIANCE OF THE INDUCED 2− NORM OF $\hat{K}_p - K_{EP}$ FOR THE 13 TESTS.

observer (HSO) in [18] diverges in this experiment. In contrast, the RHSO converges to an equivalent solution. The divergence of the HSO can be attributed to nonuniqueness of solutions of the underlying IRL problem, which resulte in singularity of the matrix $\Sigma^\top \Sigma$.

Selection of the interval used to add data to the history stacks involves important trade-offs. Longer intervals allow larger changes in two subsequent recorded data points, resulting in a lower condition number of $\Sigma^\top \Sigma + \epsilon I$; whereas, shorter intervals allow for faster population of the history stacks, which results in better utilization of excitation naturally present during the transient response of the system, especially for problems where addition of an excitation signal is not feasible. The tuning of the RHSO also requires selection of an ϵ to ensure invertibility of $\Sigma^\top \Sigma + \epsilon I$. Large values of ϵ were observed to slow down the convergence rate, a phenomenon for which the authors presently lack an explanation.

V. CONCLUSION

The experimental results demonstrate the ability of the RHSO to consistently learn an equivalent solution for the cost functional of a surrogate LQR pilot. The estimated cost functional reproduces the feedback matrix of the surrogate pilot. The robustness of the algorithm to changes in initial conditions is demonstrated through convergence obtained using randomly generated setpoints and initial guesses for unknown weights.

In solving the pilot modeling problem, the pilot is assumed to be an optimal controller that has full state information and transmits velocity commands to the quadcopter. The results of this paper indicate that this assumption is acceptable for the case where the pilot is a surrogate LQR controller. Further experimentation with human pilots will be required to establish the validity of this assumption in a real-world scenario.

The assumption that excitation signals can be designed so that they do not interrupt a human pilot from performing

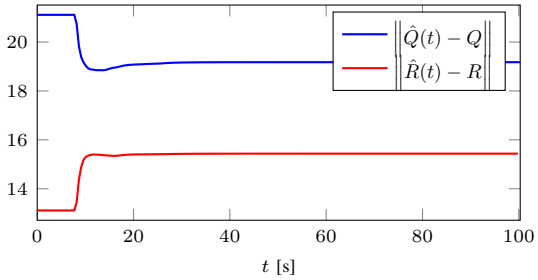

Fig. 6. A plot of the induced $2-$norm of the error between $\hat{Q}$ (blue) and Q and $\hat{R}$ (red) and R as a function of time in one of the 13 experiments.

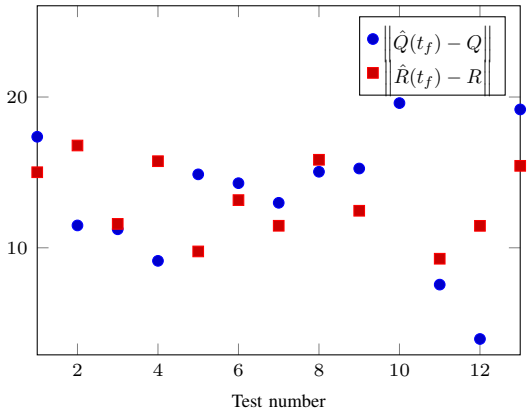

Fig. 7. Norm of the error between Q and $\hat{Q}$ and R and $\hat{R}$ obtained at $t = 200$s in the 13 experiments.

their mission is reasonable but requires careful tuning of the excitation signal so it does not become a nuisance. Validation of the assumption that a human pilot behaves like a deterministic LQR controller needs further experimentation with human pilots. Future research will focus on experimentation involving human pilots where the developed IRL method will be used to replicate their performance by learning cost functionals equivalent to the ones being minimized by them. Future work will also involve possible extensions of the developed framework to nonlinear systems and probabilistic models of pilot behavior.

REFERENCES

[1] A. Phatak, H. Weinert, I. Segall, and C. N. Day, "Identification of a modified optimal control model for the human operator," *Automatica*, vol. 12, no. 1, pp. 31–41, 1976.

[2] S. Xu, W. Tan, A. V. Efremov, L. Sun, and X. Qu, "Review of control models for human pilot behavior," *Annual Reviews in Control*, vol. 44, pp. 274–291, 2017.

[3] P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.

[4] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2000.

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

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

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

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

[9] 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.

[10] 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.

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

[12] 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.

[13] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Adv. Neural Inf. Process. Syst.*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 1449–1456.

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

[15] 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.

[16] B. Lian, V. S. Donge, F. L. Lewis, T. Chai, and A. Davoudi, "Data-driven inverse reinforcement learning control for linear multiplayer games," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[17] R. V. Self, M. Abudia, S. M. N. Mahmud, and R. Kamalapurkar, "Model-based inverse reinforcement learning for deterministic systems," *Automatica*, vol. 140, no. 110242, pp. 1–13, Jun. 2022.

[18] R. V. Self, K. Coleman, H. Bai, and R. Kamalapurkar, "Online observer-based inverse reinforcement learning," *IEEE Control Syst. Lett.*, vol. 5, no. 6, pp. 1922–1927, Dec. 2021.

[19] 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.

[20] M. Herman, V. Fischer, T. Gindele, and W. Burgard, "Inverse reinforcement learning of behavioral models for online-adapting navigation strategies," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3215–3222.

[21] 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.

[22] 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*, vol. 31, no. 14, pp. 6646–6667, 2021.

[23] J. Town, Z. Morrison, and R. Kamalapurkar, "Nonuniqueness and convergence to equivalent solutions in observer-based inverse reinforcement learning," arXiv:2210.16299, submitted to Automatica.

[24] F. Jean and S. Maslovskaya, "Inverse optimal control problem: the linear-quadratic case," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 888–893.

[25] J. P. Hespanha, *Linear systems theory*. Princeton University Press, 2009.

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

[27] M. Islam, M. Okasha, and M. M. Idres, "Trajectory tracking in quadrotor platform by using PD controller and LQR control approach," in *IOP Conf. Mater. Sci. Eng.*, vol. 260, no. 1, 2017, pp. 2451–2456.

[28] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Proc. Intell. Robot. Syst.*, 2007, pp. 153–158.

[29] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. Intell. Robot. Syst.*, vol. 3. IEEE, 2004, pp. 2451–2456.

[30] S. Rajan, S. Wang, R. Inkol, and A. Joyal, "Efficient approximations for the arctangent function," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 108–111, 2006.

[31] "Auto-tuning — px4 user guide (main)," https://docs.px4.io/main/en/config/autotune.html#background-detail, accessed: 2024-03-06.