NONUNIQUENESS AND EQUIVALENCE IN ONLINE INVERSE REINFORCEMENT LEARNING WITH APPLICATIONS TO PILOT PERFORMANCE MODELING

By

JARED CURTIS TOWN

Bachelor of Science in Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma
2021

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2023

NONUNIQUENESS AND EQUIVALENCE IN ONLINE INVERSE REINFORCEMENT LEARNING WITH APPLICATIONS TO PILOT PERFORMANCE MODELING

Thesis Approved:

Dr. Rushikesh Kamalapurkar

Thesis Advisor

Dr. He Bai

Dr. Imraan Faruque

ACKNOWLEDGMENTS

I would not be the person I am today without the love and support of those around me. I would like to thank all of the individuals that believed in me and helped me get to where I am today.

Thank you, Dr. Kamalapurkar, for your never-ending patience, perseverance and wisdom, you have sparked a flame in me to achieve more than I thought I originally could. Thank you, Dr. Faruque, for your informative conversations and hands on experience provided through your autopilot course, without it I would not have been able to accomplish this endeavor. Thank you Dr. Bai, as the support I received from your graduate students has been invaluable.

To my lab mates, you have my gratitude for providing a positive and fun learning environment.

To my parents, Timothy and Tiffany: without your love and support I would not have been able to pursue my dreams, for that you will always have my gratitude.

To my brother TJ: despite the challenges we may have faced along the way, your presence has been a constant inspiration, and I am forever thankful for the bond we share.

Finally, to my fiancée Autumn: you are the driving force and inspiration for the pursuit of this degree. I am forever inspired by your dedication and commitment to your apirations.

This research was supported, in part, by the National Science Foundation (NSF) under award number 1925147 and the College of Engineering, Architecture and Technology (CEAT) at Oklahoma State University.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: JARED TOWN

Date of Degree: MAY, 2023

Title of Study: NONUNIQUENESS AND EQUIVALENCE IN ONLINE INVERSE REINFORCEMENT LEARNING WITH APPLICATIONS TO PILOT PERFORMANCE MODELING

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: The focus of this thesis is behavior modeling for pilots of unmanned aerial vehicles. The pilot is assumed to make decisions that optimize an unknown cost functional, which is estimated from observed trajectories using a novel inverse reinforcement learning (IRL) framework. The resulting IRL problem often admits multiple solutions. Nonuniqueness necessitates the study of the notion of equivalent solutions, i.e., solutions that result in a different cost function but same feedback matrix, and convergence to such solutions. While offline algorithms that result in convergence to equivalent solutions have been developed in the literature, online, real-time techniques that address nonuniqueness are not available. In this thesis, a regularized history stack observer that converges to approximately equivalent solutions of the IRL problem is developed. Novel data-richness conditions are developed to facilitate the analysis and simulation results are provided to demonstrate the effectiveness of the developed technique.

The novel IRL observer is then adapted to the pilot modeling problem. The observer is shown to converge to one of the equivalent solutions of the IRL problem. The developed technique is implemented on a quadcopter where the pilot is modeled as a linear quadratic regulator. Experimental results demonstrate the robustness of the method and its ability to learn an equivalent cost functional.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

1.1 Motivation

Given the widespread use of small unmanned aerial systems (sUAS), quadcopters in particular, the need to manage flights efficiently in low altitude settings arises as that airspace is cluttered and turbulent. Cooperative piloting will be necessary for the guidance of these quadcopters to prevent air-to-air and air-to-obstacle collisions. Since piloting a small quadcopter in a windy obstacle-laden environment is a difficult task for pilots to do without assistance, modeling pilot performance for cooperative piloting is imperative. We envision a pilot-assist system that recommends paths to the pilots that are personalized to suit their preferences and skill levels. This study focuses on the learning component of the recommendation system that continually learns the pilot's performance by analyzing their behavior.

Taking inspiration from [1,37], we hypothesize that the pilot's skill level and preferences can be encoded in a cost functional. We then model the pilot-aircraft system as an optimal control problem where the natural tendencies and skill level [26] of the pilot are encoded into a cost functional that the pilot is assumed to optimize. We aim to recover the said cost function using flight data.

Inverse reinforcement learning (IRL) is the process of recovering the cost function of an *expert* whose trajectories are consistent with a given dynamic model [25]. In this thesis, the expert is assumed to be deterministic, behaving optimally with respect to some unknown cost functional. The objective is to estimate the cost functional from observations of the expert's performance. While the estimated cost functionals are typically utilized for behavior

imitation via (forward) reinforcement learning, the scope of this thesis is limited to the cost functional estimation.

A key goal for this thesis is the development of an IRL formulation of the pilot modeling problem [36] and application of an IRL method to solve the resulting IRL problem. The pilot is assumed to be flying a quadcopter by minimizing a quadratic cost functional. This cost functional has multiple equivalent solutions. As the quadcopter model later developed in Section 3.1 and similar linearized systems follow a product structure, at is known from [15], they will admit non-unique solutions to the corresponding IRL problem. A method for obtaining an equivalent cost function is necessary.

The IRL method of choice in this thesis is the history stack observer (HSO) for IRL developed in [33]. However, it was derived under the implicit assumption that the IRL problem admits a unique solution. Since IRL problems, and the pilot modeling problem, generally admit multiple linearly independent solutions [14,15], the uniqueness assumption is restrictive. Non-uniqueness is studied in results such as [14], where procedures to determine equivalent cost functionals are developed. It is also shown that IRL problems with non-unique solutions arise naturally in state space models that have a product structure (see [15]). Many real-world systems have a product structure, either in the original model or in the linearized model. For example, linearized dynamics of aerospace vehicles [15] have a product structure due to separation of longitudinal and lateral dynamics. The study of IRL problems that admit nonunique solutions is thus indispensable in real-world applications.

Motivated by [38], a novel online implementation for learning non-unique reward weights using the HSO formulation, called the Non-unique History Stack Observer (NHSO), is developed in this thesis. While the modification made to the HSO resembles ridge regression, the resulting convergence guarantees are surprising and require novel analysis tools and data richness conditions. The analysis shows that if the IRL problem has non-unique solutions, then Q and R converges to an equivalent solution. The developed method is assessed using numerical experiments that utilize an academic example whose corresponding IRL problem

admits non-unique solutions. To provide a comprehensive extension of the study reported in [33], we conduct additional testing on a problem that has a unique solution. As the original HSO methodology was developed to be used with a Kalman gain, we have therefore included simulations that incorporate noise. In application to the pilot modeling problem, the NHSO is modified for identification of the weight estimates without state estimates. Additionally, a linearized quadcopter model that assumes a pilot's control inputs as velocity commands is developed. Finally, a cost function that recognizes a pilot's preferred performance penalties is produced.

1.2 Literature Review

Inverse reinforcement learning (IRL) is the process of measuring an "experts"' inputs and subsequent behavior over time and obtaining their cost function where said "expert" generates trajectories that are consistent with a given dynamic model [25]. This "expert" is assumed to be behaving optimally with respect to some unknown cost function. IRL methods such as [2,3,19–21,23–25,28,30,32,34,39,40] are utilized to uncover the true cost function. 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, online setting, methods such as [4, 8, 29, 31, 33] have been developed. These methods are typically model-based and use a single continuous trajectory to learn the cost function of an expert. A notable result is obtained in [22] where an online and model-free approach that utilizes a neural network to solve the IRL problem in the presence of adversarial attacks is developed. However, this method only identifies the state penalty matrix, Q, and is unable to identify the control penalty matrix, R.

The IRL methods recently developed in [7, 22] and [38] study nonuniqueness of solutions and guarantee convergence to the set of equivalent solutions. In [7, 38] the problem is solved in an offline setting as opposed to the online and real-time problem under consideration in this thesis and demonstrated in [22]. [7, 22] only identify equivalent solutions of the state

penalty matrix, Q, by knowing the expert's control input, u. These methods can further identify a unique solution of Q if the expert's control penalty matrix, R, is known.

The method developed in this article identifies equivalent Q and R matrices given u and measurements of the state, x, are obtainable. Unique solutions of the corresponding IRL problem can be obtained, up to a scaling factor, provided they exist. Similarly, a new method to solve the unique IRL problem up to a scaling factor through non-cooperative linear quadratic differential games is developed in [11], it is then expanded to an online IRL method using differential games in [12] and compared to, and converges faster than, the method in [16], which this article draws inspiration from. However, [12] does not address non-unique solutions.

1.3 Contributions

The key contributions of this thesis are as follows:

1. This thesis extends the IRL HSO in [33] to problems where the observed trajectories can be optimal with respect to multiple cost functions. A learner with access to the state space model, controller input, and measurement data reconstructs an equivalent cost function of an expert.

2. The proposed modification to the HSO is inspired by ridge regression, but has a surprising convergence property. Under ideal conditions (no noise and persistently exciting regressor), the convergence is exact, as opposed to ridge regression, where the solutions are off by a factor proportional to the regularization coefficient. As demonstrated by the simulation results in this thesis, offline implementations of ridge regression techniques on the same dataset do not converge, but the developed HSO does.

3. A novel analysis approach that guarantees convergence of the learned solution to a neighborhood of an equivalent solution is developed. The analysis makes use of the invariance principle and a novel data informativity condition.

4. The utility of the developed model-based IRL algorithm is demonstrated in simulation by applying it to different academic examples that admit unique and nonunique solutions.

5. The pilot performance modeling problem utilizing a surrogate LQR pilot is formulated in an IRL framework and solved using real-world experimental data.

CHAPTER II

NONUNIQUENESS AND CONVERGENCE TO EQUIVALENT SOLUTIONS IN OBSERVER-BASED INVERSE REINFORCEMENT LEARNING

A key challenge in solving the deterministic inverse reinforcement learning (IRL) problem online and in real-time is the existence of multiple solutions. Nonuniqueness necessitates the study of the notion of equivalent solutions, i.e., solutions that result in a different cost function but the same feedback matrix, and convergence to such solutions. While *offline* algorithms that result in convergence to equivalent solutions have been developed in the literature, online, real-time techniques that address nonuniqueness are not available. In this chapter, a regularized history stack observer that converges to approximately equivalent solutions of the IRL problem is developed. Novel data-richness conditions are developed to facilitate the analysis and simulation results are provided to demonstrate the effectiveness of the developed technique.

2.1 Problem Formulation

The system being controlled by the expert is assumed to be a linear system of the form

$$\dot{x}(t) = Ax + Bu, \tag{2.1.1}$$

with output

$$y' = Cx(t), \tag{2.1.2}$$

where the state is $x \in \mathbb{R}^n$ and the control input is $u \in \mathbb{R}^m$. The system matrices are given as $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and the output and output matrix are given as $y' \in \mathbb{R}^L$ and $C \in \mathbb{R}^{L\times n}$ respectively.

The expert is assumed to be an optimal controller that optimizes the cost functional

$$J(x_0, u(\cdot)) = \int_0^\infty \left(x(t)^{\mathrm{T}} Q x(t) + u(t)^{\mathrm{T}} R u(t)\right) dt, \tag{2.1.3}$$

where $x(\cdot)$ is the system trajectory under the optimal control signal $u(\cdot)$ and starting from the initial condition x_0, and $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are unknown positive semi-definite and positive definite matrices, respectfully. That is, the policy of the expert is given by $u = K_{Ep}x$, where $K_{Ep} \in \mathbb{R}^{m\times n}$ is obtained by solving the algebraic Riccati equation (ARE) corresponding to the optimal control problem described by the system in (2.1.1) and the cost functional in (2.1.3). The following assumption ensures that the IRL problem is well-posed.

Assumption 2.1.1 *The pair (A, B) is stabilizable and the pairs (A, C) and $(A, \sqrt{Q})$ are detectable.*

Stabilizability of (A, B) and detectability of $(A, \sqrt{Q})$ is needed for the optimal controller to exist and detectability of (A, C) guarantees the existence of a matrix L such that $A - LC$ is Hurwitz [9, Lemma 21.1].

The learning objective is to estimate, online and in real-time, the unknown matrices in the cost functional using knowledge of the system matrices, A, B, and C, and input-output data. Generally, for a system (A, B, C), a given set of input-output trajectories is optimal with respect to multiple cost functionals. As a result, the true cost functional cannot generally be estimated from data. Instead, an equivalent solution to the IRL problem is sought (see Definition 2.2.1 and [38]).

While the HSO in [33] is an effective technique to solve the IRL problem online and in real-time, the analysis focuses on the error between the true cost functional matrices and their estimates, and as such, implicitly assumes uniqueness of solutions. As such, the method in [33] cannot be applied to a large class of IRL problems that admit multiple solutions. In this chapter, the HSO is extended to be applicable to IRL problems that admit multiple solutions.

2.2 Nonuniqueness and the History Stack Observer

To facilitate the discussion, this section provides a brief summary of the HSO developed in [33] and highlights the key problem that is resolved in this chapter.

2.2.1 Equivalent Solutions and Equivalence Metric

If the state and control trajectories of the system are optimal with respect to the cost functional in (2.1.3) and the assumptions in Section 2.1 are met, then there exists a matrix S such that Q, R, A, B, and the optimal trajectories $x(\cdot)$ and $u(\cdot)$ satisfy the Hamilton-Jacobi-Bellman (HJB) equation

$$x(t)^{\mathrm{T}} \left(A^{\mathrm{T}}S + SA - SBR^{-1}B^{\mathrm{T}}S + Q\right) x(t) = 0 \tag{2.2.1}$$

and the optimal control equation

$$u(t) = u^*(x(t)) \coloneqq -R^{-1}B^{\mathrm{T}}Sx(t) \tag{2.2.2}$$

$\forall t \in \mathbb{R}_{\geq 0}$. The expert's feedback matrix is then given by $K_{Ep} = -R^{-1}B^{\mathrm{T}}S$. The HJB equation and the optimal control equation facilitate the definition of an equivalent solution.

Definition 2.2.1 *A solution ($\hat{Q}$, $\hat{S}$, $\hat{R}$) is called an equivalent solution of the IRL problem if it satisfies the ARE*

$$A^{\mathrm{T}}\hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^{\mathrm{T}}\hat{S} + \hat{Q} = 0 \tag{2.2.3}$$

and optimization of the performance index J, with $Q = \hat{Q}$ and $R = \hat{R}$, results in the same feedback matrix as the one utilized by the expert, that is,

$$\hat{K}_P \coloneqq \hat{R}^{-1}B^{\mathrm{T}}\hat{S} = K_{EP}.$$

Given an estimate $\hat{x}$ of the state x, a measurement of the control signal, u, and estimates $\hat{Q}$, $\hat{R}$, and $\hat{S}$ of Q, R, and S, respectively, (2.2.1) and (2.2.2) can be evaluated to develop an observation error that evaluates to zero if the state estimates and estimates of the matrices Q,

R, and S are correct. In the following, the observation error is used to improve the estimates by framing the IRL problem as a state estimation problem. To facilitate the observer design, equations (2.2.1) and (2.2.2) are linearly parameterized as

$$0 = 2\sigma_{R2}(u)W_R^* + B^{\mathrm{T}} \left(\nabla_x \sigma_S(x)\right)^{\mathrm{T}} W_S^*, \tag{2.2.4}$$

$$0 = \nabla_x \left((W_S^*)^{\mathrm{T}} \sigma_S(x)\right) \left(Ax(t) + Bu(t)\right) + (W_Q^*)^{\mathrm{T}} \sigma_Q(x) + (W_R^*)^{\mathrm{T}} \sigma_{R1}(u), \tag{2.2.5}$$

where $x^{\mathrm{T}} S x = (W_S^*)^{\mathrm{T}} \sigma_S(x)$, $x^{\mathrm{T}} Q x = (W_Q^*)^{\mathrm{T}} \sigma_Q(x)$, $u^{\mathrm{T}} R u = (W_R^*)^{\mathrm{T}} \sigma_{R1}(u)$, and $Ru = \sigma_{R2}(u) W_R^*$, and $W_S^* \in \mathbb{R}^{P_S}$, $W_Q^* \in \mathbb{R}^{P_Q}$, $W_R^* \in \mathbb{R}^M$ are the ideal weights with P_S, P_Q, and M being the number of basis functions in the respective linear parameterizations.

Motivated by (2.2.4), and using the estimates $\hat{W}_S$, $\hat{W}_Q$, and $\hat{W}_R$ for W_S^*, W_Q^*, and W_R^* respectively, (2.2.4) a control residual error is defined as

$$\Delta_u' \coloneqq 2\sigma_{R2}(u)\hat{W}_R + B^{\mathrm{T}} \left(\nabla_x \sigma_S(x)\right)^{\mathrm{T}} \hat{W}_S. \tag{2.2.6}$$

Similarly, from (2.2.5), the inverse Bellman error is defined as

$$\delta' \coloneqq \nabla_x \left((\hat{W}_S)^{\mathrm{T}} \sigma_S(x)\right) \left(Ax(t) + Bu(t)\right) + (\hat{W}_Q)^{\mathrm{T}} \sigma_Q(x) + (\hat{W}_R)^{\mathrm{T}} \sigma_{R1}(u). \tag{2.2.7}$$

Separating out $\hat{W}' = \left[\hat{W}_S, \hat{W}_Q, \hat{W}_R\right]^{\mathrm{T}}$ yields

$$\begin{bmatrix} \delta'\left(x, u, \hat{W}'\right) \\ \Delta_u'\left(x, u, \hat{W}'\right) \end{bmatrix} = \begin{bmatrix} \sigma_{\delta'}(x, u) \\ \sigma_{\Delta_u'}(x, u) \end{bmatrix} \begin{bmatrix} \hat{W}_S \\ \hat{W}_Q \\ \hat{W}_R \end{bmatrix}, \tag{2.2.8}$$

where

$$\sigma_{\delta'}(x, u) = \begin{bmatrix} (Ax + Bu)^{\mathrm{T}} (\nabla_x \sigma_S(x))^{\mathrm{T}} & \sigma_Q(x)^{\mathrm{T}} & \sigma_{R1}(u)^{\mathrm{T}} \end{bmatrix} \tag{2.2.9}$$

and

$$\sigma_{\Delta_u'}(x, u) = \begin{bmatrix} B^{\mathrm{T}} (\nabla_x \sigma_S(x))^{\mathrm{T}} & 0_{m \times P_S + P_Q} & 2\sigma_{R2}(u) \end{bmatrix}. \tag{2.2.10}$$

The scaling ambiguity inherent in linear quadratic optimal control, which is apparent in the fact that $\hat{W}' = 0$ is a solution of (2.2.4) and (2.2.5), is resolved, without loss of generality,

by assigning an arbitrary value to one element of $\hat{W}'$. Selecting r_1 arbitrarily and removing it from (2.2.8) yields scale-aware definitions of the control residual error and the inverse Bellman error given by

$$\begin{bmatrix} \delta\left(x, u, \hat{W}\right) \\ \Delta_u\left(x, u, \hat{W}\right) \end{bmatrix} = \begin{bmatrix} \sigma_\delta(x, u) \\ \sigma_{\Delta_u}(x, u) \end{bmatrix} \begin{bmatrix} \hat{W}_S \\ \hat{W}_Q \\ \hat{W}_R^- \end{bmatrix} + \begin{bmatrix} u_1^2 r_1 \\ 2u_1 r_1 \\ 0_{m-1\times 1} \end{bmatrix}, \tag{2.2.11}$$

where $\hat{W}_R^-$ denotes $\hat{W}_R$ with the first element removed. In this chapter, the error system in (2.2.11) is used as an *equivalence metric* to develop an observer-based IRL method.

2.2.2 The Original History Stack Observer

Pairing the innovation $y - C\hat{x}$ with the inverse bellman error and control residual error from (2.2.11) yields the observation error [1]

$$\omega = \left(\begin{bmatrix} Cx \\ \Sigma_u \end{bmatrix} - \begin{bmatrix} C\hat{x} \\ \hat{\Sigma}\hat{W} \end{bmatrix} \right). \tag{2.2.12}$$

Using the observation error, the history stack observer is designed to be of the form

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{W}} \end{bmatrix} = \begin{bmatrix} A\hat{x} + Bu \\ 0_{P_S+P_Q+M-1} \end{bmatrix} + K \left(\begin{bmatrix} Cx \\ \Sigma_u \end{bmatrix} - \begin{bmatrix} C\hat{x} \\ \hat{\Sigma}\hat{W} \end{bmatrix} \right) \tag{2.2.13}$$

where $\hat{W} = [\hat{W}_S, \hat{W}_Q, \hat{W}_R^-]$,

$$\hat{\Sigma} \coloneqq \begin{bmatrix} \sigma_\delta\left(\hat{x}(t_1), u(t_1)\right) \\ \sigma_{\Delta_u}\left(\hat{x}(t_1), u(t_1)\right) \\ \vdots \\ \sigma_\delta\left(\hat{x}(t_N), u(t_N)\right) \\ \sigma_{\Delta_u}\left(\hat{x}(t_N), u(t_N)\right) \end{bmatrix}, \Sigma_u \coloneqq \begin{bmatrix} -u_1^2(t_1) r_1 \\ -2u_1(t_1) r_1 \\ 0_{m-1\times 1} \\ \vdots \\ -u_1^2(t_N) r_1 \\ -2u_1(t_N) r_1 \\ 0_{m-1\times 1} \end{bmatrix},$$

[1] See [33] for the detailed process

and the gain K is selected to be

$$K := \begin{bmatrix} K_3 & 0_{n\times N+Nm} \\ 0_{P_S+P_Q+M-1\times L} & K_4(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma})^{-1}\hat{\Sigma}^{\mathrm{T}} \end{bmatrix}. \tag{2.2.14}$$

Remark 2.2.1 *In the case of noisy measurements, the feedback gain K in* (2.2.14) *can be replaced by the Kalman gain. While the use of the Kalman gain results in improved performance (see Section 2.3.3), extending the stability guarantees of the HNSO to the case where the measurements are noisy and K is the Kalman gain is out of the scope of this thesis.*

The matrices $\hat{\Sigma} \in \mathbb{R}^{N(m+1)\times P_S+P_Q+M-1}$ and $\Sigma_u \in \mathbb{R}^{N(m+1)}$ are constructed using the dataset $\{(\hat{x}(t_i), u(t_i))\}_{i=1}^{N}$, recorded at time instances $\{t_1, \dots t_N\}$, with $N \geq P_S + P_Q + M - 1$. The dataset is referred to hereafter as a *history stack*. To ensure convergence of the weights, updated using (2.2.13), to an equivalent solution (see Theorem 2.2.1 below), the history stack is recorded using a minimum singular value maximization algorithm. At any time, two separate history stacks, H_1 and H_2 are maintained. The history stack H_1 is used to compute the matrices $\hat{\Sigma}$ and Σ_u in (2.2.13) and H_2 is populated with current state estimates and control inputs. Both history stacks are initialized as zero matrices of the appropriate size. As state estimates become available, they are selectively added, along with the corresponding control input, to H_2. A new state estimate is selected to replace an existing state estimate in H_2 if the replacement decreases the condition number of $(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma})$. Once the data in H_2 are such that the condition number of $(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma})$ is lower than a user-selected threshold, and a predetermined amount of time has passed since the last update of H_1, we set $H_1 = H_2$ and purge H_2 by setting it back to a zero matrix. The purging process ensures that old and possibly erroneous state estimates are removed from H_1.

The IRL method developed in this chapter requires that the expert's behavior is optimal, which implies that $u(t) = K_{EP}x(t)$ for all t. Since the true values of the state are not accessible, in general, for the data points stored in the history stack H_1, $K_{EP}\hat{x}(t_i) - u(t_i) \neq 0$, which results in inaccuracy in the estimation of an equivalent solution. Since the state

estimates converge to the true state exponentially, the purging process described above ensures that the discrepancy $\max_{i=1,\cdots,N} \|K_{EP}\hat{x}(t_i) - u(t_i)\|$ is monotonically decreasing in time, and so is the resulting inaccuracy in the estimation of an equivalent solution.

Generally, given a system model with output (or state) and control trajectories, there are multiple sets of Q, R, and S matrices that all solve the IRL problem [14, 15]. As such, the IRL problem, as posed in [33], is not well-defined. In fact, the stability theorem in [33] relies on the assumption that $\hat{\Sigma}$ is full rank. Due to purging and improved state estimates, Σ being full rank implies $\hat{\Sigma}$ is eventually full rank, and as a result, $\Sigma W = \Sigma_u$ has a unique solution. Since uniqueness does not generally hold [15], the HSO must be modified to address the non-unique case. In this chapter, the full rank condition, and subsequently, the uniqueness assumption is relaxed using an update rule motivated by ridge [10] and lasso [35] regression. Furthermore, unlike [33], since convergence to a specific set of parameters can no longer be guaranteed, direct analysis of the parameter estimation error is no longer viable. As such, an analysis framework is developed where convergence of the *equivalence metric* is analyzed using Lyapunov methods. Data richness conditions are then derived to ensure that convergence of the equivalence metric implies convergence to equivalent solutions.

2.2.3 Regularized History Stack Observer for Non-Unique Solutions (NHSO)

To avoid the uniqueness assumption, and subsequently, to allow for a rank-deficient $\hat{\Sigma}$, the gain matrix of the HSO is modified in this chapter to include a regularization term to yield

$$K \coloneqq \begin{bmatrix} K_3 & 0_{n\times N+Nm} \\ 0_{P_S+P_Q+M-1\times L} & K_4(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I)^{-1}\hat{\Sigma}^{\mathrm{T}} \end{bmatrix}, \tag{2.2.15}$$

where $\epsilon \geq 0$ is a small constant selected by the user to ensure invertibility of $\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I$. Instead of using the condition number of $(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma})$ to select data points for storage in the history stack, the condition number of $(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I)$ is used. In addition, since $\hat{\Sigma}$ cannot be full rank, we need a different way to detect whether the recorded data are sufficient for estimation of an equivalent solution.

The following theorems establish that under a novel informativity condition on the recorded data, the modification above leads to an equivalent solution when the IRL problem admits multiple solutions, and the correct solution when the IRL problem admits a unique solution up to a scaling factor. While the modification itself is relatively minor, the above somewhat surprising results are the key contributions of this work.

To facilitate the analysis, let $\Delta(t) \coloneqq \Sigma_u - \hat{\Sigma}\hat{W}(t)$ where Σ_u and $\hat{\Sigma}$ are piecewise constant through the purging process of the history stacks. Using the update law in (2.2.13), the time-derivative of Δ can be expressed as

$$\dot{\Delta} = -\hat{\Sigma}K_4(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I)^{-1}\hat{\Sigma}^{\mathrm{T}}\Delta \tag{2.2.16}$$

The analysis requires a data informativity condition summarized in Definition 2.2.2 below.

Definition 2.2.2 *The signal $(\hat{x}, u)$ is finitely informative (FI) if there exists a time instance $T > 0$ such that for some $\{t_1, t_2, \ldots, t_N\} \subset [0, T]$,*

$$\begin{aligned} \operatorname{span}\{\hat{x}(t_i)\}_{i=1}^{N} &= \mathbb{R}^n \\ \operatorname{span}\left\{\hat{x}(t_i)\hat{x}(t_i)^{\mathrm{T}}\right\}_{i=1}^{N} &= \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\mathrm{T}}\} \quad \textit{and} \\ \Sigma_u &\in (\operatorname{Null}(\hat{\Sigma}^{\mathrm{T}}))^{\perp}. \end{aligned} \tag{2.2.17}$$

Remark 2.2.2 *The three FI conditions in Definition 2.2.2 are utilized in the subsequent analysis to show that as the equivalence metric converges to zero, the corresponding weight estimates converge to an equivalent solution. These conditions are not restrictive as long as a sufficient excitation signal is used to fulfill the conditions. In practice, at least as many unique signals as weights are needed to learn an equivalent solution.*

1. *The condition $\Sigma_u \in (\operatorname{Null}(\hat{\Sigma}^{\mathrm{T}}))^{\perp}$ is equivalent to $\Sigma_u \in \operatorname{Range}(\hat{\Sigma})$. It is met provided at least one set of weights $\hat{W}$ satisfies $\Sigma_u = \hat{\Sigma}\hat{W}$. Since the expert is assumed to be optimal, we know that $\Sigma_u \in \operatorname{Range}(\Sigma)$. Due to improving state estimates and the purging algorithm, $\hat{\Sigma}$ converges to Σ, and as a result, there exists $T > 0$ such that $\Sigma_u \in \operatorname{Range}(\hat{\Sigma})$ for all $t \geq T$.*

2. *The condition* $\operatorname{span}\{\hat{x}(t_i)\}_{i=1}^{N} = \mathbb{R}^n$ *is an excitation-like condition that requires the state estimates to be linearly independent. This condition is not overly restrictive, but can fail if the system trajectories evolve on a subspace of dimension less than* n*.*

3. *The condition* $\operatorname{span}\left\{\hat{x}(t_i)\hat{x}(t_i)^{\mathrm{T}}\right\}_{i=1}^{N} = \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\mathrm{T}}\}$ *a sufficient condition for* $\hat{x}_i^{\mathrm{T}}\hat{M}\hat{x}_i = 0, \forall i = 1, \cdots, N$ *to imply* $\hat{M} = 0$*. This condition is needed for satisfaction of the HJB equation at a finite number of data points to also imply satisfaction of the ARE.*

Lemma 2.2.1 *If* $\hat{\Sigma}$ *and* Σ_u *satisfy* (2.2.17)*, then*

$$\Omega_\Delta \cap \operatorname{Null}(\hat{\Sigma}^{\mathrm{T}}) = \{0\}, \tag{2.2.18}$$

where

$$\Omega_\Delta \coloneqq \left\{\Delta \in \mathbb{R}^{N(m+1)} \mid \Delta = \Sigma_u - \hat{\Sigma}y, \text{ for some } y \in \mathbb{R}^{P_S+P_Q+M-1}\right\}.$$

Proof. If $\Delta \in \operatorname{Null}(\hat{\Sigma}^{\mathrm{T}})$, then Δ is given by some linear combination of the basis for the null space of $\hat{\Sigma}^{\mathrm{T}}$. Let Σ_{Null} be a matrix whose columns are the basis vectors of the null space of $\hat{\Sigma}^{\mathrm{T}}$. Then, $\Delta \in \operatorname{Null}(\hat{\Sigma}^{\mathrm{T}})$ implies that $\Delta = \Sigma_{\mathrm{Null}} W_{\mathrm{Null}}$ for some vector W_{Null} whose elements are the coefficients in the linear combination of the basis of the null space of $\hat{\Sigma}^{\mathrm{T}}$ that makes up Δ. This Δ has to also be equal to $\Sigma_u - \hat{\Sigma}\hat{W}$ for some $\hat{W}$. So, there exist weights W_{Null} and $\hat{W}$ such that $\Sigma_{\mathrm{Null}} W_{\mathrm{Null}} = \Sigma_u - \hat{\Sigma}\hat{W}$. Rearranging the terms, there exist weights W_{Null} and $\hat{W}$ such that $\begin{bmatrix}\Sigma_{\mathrm{Null}} & \hat{\Sigma}\end{bmatrix}\begin{bmatrix} W_{\mathrm{Null}} \\ \hat{W}\end{bmatrix} = \Sigma_u$. That is, Σ_u can be written as a linear combination of the columns of $\hat{\Sigma}$ and the columns of Σ_{Null}. However, since $\operatorname{Rank}(\hat{\Sigma}) = \operatorname{Null}(\hat{\Sigma}^{\mathrm{T}})^{\perp}$, every linear combination of columns of $\hat{\Sigma}$ is orthogonal to every linear combination of the columns of Σ_{Null}, we know that Σ_u has two orthogonal components, one that lives in the range space of $\hat{\Sigma}$ and another that is contained in the null space of $\hat{\Sigma}^{\mathrm{T}}$. If our data are such that $\Sigma_u \in \operatorname{Null}(\hat{\Sigma}^{T})^{\perp}$, then the component that is contained in the null space of $\hat{\Sigma}^{\mathrm{T}}$ is zero. That is, $W_{\mathrm{Null}} = 0$, which implies that $\Delta = 0$. ■

Remark 2.2.3 *Note that if the IRL problem has a unique solution, then the condition,* $\Sigma_u \in (\text{Null}(\hat{\Sigma}^{\text{T}}))^{\perp}$ *in Definition 2.2.2, is trivially met whenever* $N \geq P_S + P_Q + M - 1$ *and* $\hat{\Sigma}$ *is full rank.*

Theorem 2.2.1 below shows that provided the weights $\hat{W}$ are updated using the update law in (2.2.13), and the trajectories are finitely informative as per Definition 2.2.2, then the equivalence metric Δ converges to the origin for a given *fixed* $\hat{\Sigma}$ and Σ_u.

Theorem 2.2.1 *If* $\Sigma_u \in \text{Null}(\hat{\Sigma}^{\text{T}})^{\perp}$, $\epsilon \geq 0$ *is selected to ensure invertibility of* $\hat{\Sigma}^{\text{T}}\hat{\Sigma} + \epsilon I$, *and* $\hat{R}$ *is invertible then the solutions of* (2.2.16) *with the gain* K *in* (2.2.15) *satisfy* $\lim_{t\to\infty} \Delta(t) = 0$.

In addition if full state information is available (i.e., $\hat{x} = x$ *and as a result,* $\hat{\Sigma} = \Sigma$*),* $\Delta = \Sigma_u - \Sigma\hat{W} = 0$, $\text{span}\{x_i\}_{i=1}^{N} = \mathbb{R}^n$, *and if* $\text{span}\{x_i x_i^{\text{T}}\}_{i=1}^{N} = \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\text{T}}\}$, *then the matrices* $\hat{Q}$, $\hat{S}$, *and* $\hat{R}$, *extracted from* $\hat{W}$, *constitute an equivalent solution of the IRL problem per Definition 2.2.1.*

Remark 2.2.4 *The invertibility of* $\hat{R}$ *is needed for* K_P *to be well-defined. While this is difficult to ensure apriori in general, it can be guaranteed in the specific case where* R *is diagonal by using a projection operator to ensure that all diagonal elements of* $\hat{R}$ *remain positive.*

Proof. Let $D = \mathbb{R}^{N(m+1)}$ and consider the positive definite and radially unbounded candidate Lyapunov function

$$V(\Delta) = \frac{1}{2}\Delta^{\text{T}}\Delta. \tag{2.2.19}$$

The orbital derivative of V along the solutions of (2.2.16) is given by

$$\dot{V}(\Delta) = -\Delta^{\text{T}}\hat{\Sigma}K_4(\hat{\Sigma}^{\text{T}}\hat{\Sigma} + \epsilon I)^{-1}\hat{\Sigma}^{\text{T}}\Delta. \tag{2.2.20}$$

For any $c > 0$, the sublevel set $\Omega_c := \{\Delta \in D | V(\Delta) \leq c\}$ is compact and positively invariant and the set Ω_Δ in (2.2.1) can be shown to be closed and positively invariant. As such, the

intersection $\Omega = \Omega_c \cap \Omega_\Delta$ is compact and positively invariant. By the invariance principle [17, Th 4.4], all trajectories of Δ in (2.2.16) starting in Ω converge to the largest invariant subset of $\{\Delta \in \Omega \mid \dot{V}(\Delta) = 0\}$. The set $\{\Delta \in \Omega | \dot{V}(\Delta) = 0\}$, is equal to $\text{Null}(\hat{\Sigma}^{\text{T}}) \cap \Omega$ as $\hat{\Sigma}^{\text{T}}\Delta = 0$ only when $\Delta \in \text{Null}(\hat{\Sigma}^{\text{T}})$. Furthermore, from Lemma 2.2.1, provided $\Sigma_u \in (\text{Null}(\hat{\Sigma}^{\text{T}}))^{\perp}$, the only Δ that can be in $\hat{\Sigma}^{\text{T}} \cap \Omega_\Delta$ is $\Delta = 0$. Since the singleton $\{0\}$ is positively invariant with respect to the dynamics in (2.2.16), it is also the largest invariant subset of $\{\Delta \in \Omega | \dot{V}(\Delta) = 0\}$. As a result, by the invariance principle, all trajectories starting in Ω converge to the origin. Since V is radially unbounded, Ω_c can be selected to be large enough to include any initial condition in Ω_Δ. Thus, all trajectories starting in Ω_Δ converge to the origin.

To prove equivalence, the equality $\hat{R}^{-1}B^{\text{T}}\hat{S} = K_{EP}$ must be established. Indeed, if $\{x_i\}_{i=1}^N$ spans $\mathbb{R}^n$ there is a unique matrix K that satisfies $u_i = Kx_i$ for all $i = 1, \ldots, N$. Letting $\mathbb{U} = [u_1, \ldots, u_N]$ and $\mathbb{X} = [x_1, \ldots, x_N]$, this unique matrix is given by $K = \mathbb{U}\mathbb{X}^{\text{T}}(\mathbb{X}\mathbb{X}^{\text{T}})^{-1}$. It is also known that because the expert's behavior is optimal, the observed data satisfy $u_i = -K_{EP}x_i$ for all $i = 1, \ldots, N$. Since $\Delta = 0$, the observed data points satisfy $u_i = -\hat{R}^{-1}B^{\text{T}}\hat{S}x_i$ for all $i = 1, \ldots, N$. Since there is only one matrix K that satisfies $u_i = -Kx_i$ for all $i = 1, \ldots, N$, all three of the matrices above must be equal, i.e., $K = K_{EP} = \hat{R}^{-1}B^{\text{T}}\hat{S}$.

The fact that if $\Delta = 0$ then $x_i^{\text{T}}\left(A^{\text{T}}\hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^{\text{T}}\hat{S} + \hat{Q}\right)x_i = 0$ holds for all points in H_1 is immediate from the construction of Δ. Furthermore, with a slight modification of the proof from [18], $(\hat{Q},\ \hat{S},\ \hat{R})$ can be proven to satisfy the ARE if $\Delta = 0$ and $\{x_i x_i^{\text{T}}\}_{i=1}^N$ spans all symmetric matrices. To that end, let e_i be the basis vector of zeros with a one in the i^{th} position such that $e_j e_k^{\text{T}} + e_k e_j^{\text{T}} = \sum_{i=1}^N \alpha_i x_i x_i^{\text{T}}$ for some $\alpha_1 \cdots \alpha_N \in \mathbb{R}$. Rewriting (2.2.1) with $\hat{M} = \left(A^{\text{T}}\hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^{\text{T}}\hat{S} + \hat{Q}\right)$, $\sum_{i=1}^N \alpha_i x_i^{\text{T}}\hat{M}x_i = \sum_{i=1}^N \sum_{p=1}^n \sum_{q=1}^n \alpha_i x_{i,p}\hat{M}_{p,q}x_{i,q} = \sum_{i=1}^N \sum_{p=1}^n \hat{M}_{p,q} \sum_{q=1}^n \alpha_i x_{i,p}x_{i,q}$. Now,

for any fixed j, k, select $\{\alpha_i\}_{i=1}^N$ such that $\sum_{i=1}^N \alpha_i x_i x_i^{\mathrm{T}} = e_j e_k^{\mathrm{T}} + e_k e_j^{\mathrm{T}}$, where

$$\sum_{i=1}^{N} \alpha_i x_i x_i^{\mathrm{T}} = \begin{cases} 1 & \text{if } p = j, q = k \\ 1 & \text{if } p = k, q = j \\ 0 & \text{otherwise} \end{cases} \tag{2.2.21}$$

As a result, $\sum_{i=1}^N \sum_{p=1}^n \hat{M}_{p,q} \sum_{q=1}^n \alpha_i x_{i,p} x_{i,q} = e_k^{\mathrm{T}} \hat{M} e_j + e_j^{\mathrm{T}} \hat{M} e_k = \hat{M}_{j,k} + \hat{M}_{k,j} = 2\hat{M}_{j,k} = 0$. Since j and k were arbitrary, $\hat{M} = 0$. That is, the tuple $(\hat{Q},\ \hat{S},\ \hat{R})$ satisfies the ARE and constitutes an equivalent solution of the IRL problem. ■

Theorem 2.2.1 can be used to obtain the final result summarized in the definition and the corollary below.

Definition 2.2.3 *Given $\varpi \geq 0$ A solution $(\hat{Q},\ \hat{S},\ \hat{R})$ to the IRL problem is called an $\varpi-$equivalent solution of the IRL problem if*

$$\left\| \hat{M} \right\| \leq \varpi,$$

where $\hat{M} = A^{\mathrm{T}}\hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^{\mathrm{T}}\hat{S} + \hat{Q}$, and optimization of the performance index J, with $Q = \hat{Q}$ and $R = \hat{R}$, results in a feedback matrix, $\hat{K}_p$, that satisfies

$$\left\| \hat{R}^{-1}B^{\mathrm{T}}\hat{S} - K_{EP} \right\| \leq \varpi.$$

Corollary 2.2.1 *Given $\varpi \geq 0$ if t_1 is large enough, $\Sigma_u \in \mathrm{Null}(\hat{\Sigma}^{\mathrm{T}})^{\perp}$, $\epsilon \geq 0$ is selected to ensure invertibility of $\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I$, K_3 is selected so that $A - K_3 C$ is Hurwitz, $\mathrm{span}\{\hat{x}(t_i)\}_{i=1}^N = \mathbb{R}^n$, and $\mathrm{span}\{\hat{x}(t_i)(\hat{x}(t_i))^{\mathrm{T}}\}_{i=1}^N = \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\mathrm{T}}\}$, and if the matrix $\hat{R}$, extracted from $\hat{W}$ is invertible such that $\|\hat{R}^{-1}(t)\| \leq \underline{R}$ for some $0 \leq \underline{R} < \infty$, then the matrices $\hat{Q}$, $\hat{S}$, and $\hat{R}$, extracted from $\hat{W}$, converge to a $\varpi-$equivalent solution of the IRL problem.*

Proof. The control residual error established in (2.2.6) can be manipulated into

$$\sigma_{\Delta_u'}\left(\hat{x}(t_i), u_i\right) \hat{W}'(t) = \hat{R}(t)\left(\tilde{K}_P(t)\hat{x}(t_i) + K_{EP}\tilde{x}(t_i)\right)$$

where $\tilde{K}_P(t) \coloneqq \hat{R}^{-1}(t)B^{\mathrm{T}}\hat{S}(t) - K_{EP}$ and $\tilde{x}(t_i) \coloneqq x(t_i) - \hat{x}(t_i)$. Using the triangle inequality $\left\|\tilde{K}_P\hat{x}(t_i)\right\| \leq \left\|\hat{R}^{-1}(t)\sigma_{\Delta'_u}\left(\hat{x}(t_i), u_i\right)\hat{W}'(t)\right\| + \|K_{EP}\tilde{x}(t_i)\|$.

Note that if $\mathrm{span}\{\hat{x}(t_i)_{i=1}^N\} = \mathbb{R}^n$ then $\exists\ c$ such that $\left\|\tilde{K}_P\hat{x}(t_i)\right\| \leq \frac{\varpi}{c}, \forall i$ implies $\left\|\tilde{K}_P\right\| \leq \varpi$. For this $c > 0$, if t and t_i are large enough such that the state estimation error $\tilde{x}(t_i)$ and equivalence metric $\Delta(t)$ satisfy $\|\tilde{x}(t_i)\| \leq \frac{\varpi}{2c\|K_{EP}\|}$ and $\left\|\sigma_{\Delta'_u}\left(\hat{x}(t_i), u_i\right)\hat{W}'(t)\right\| \leq \frac{\varpi}{2c\underline{R}}$, respectively, then $\left\|\tilde{K}_P\hat{x}(t_i)\right\| \leq \frac{\varpi}{c}$, which implies $\left\|\tilde{K}_P\right\| \leq \varpi$.

The inverse Bellman error established in (2.2.7) can be manipulated into

$$\sigma_{\delta'}\left(\hat{x}(t_i), u_i\right)\hat{W}'(t) = \hat{x}^{\mathrm{T}}(t_i)\hat{M}(t)\hat{x}(t_i) + g\left(\hat{K}_P(t), \hat{x}_i, K_{EP}, x_i\right),$$

where g satisfies $g = O\left(\left\|\tilde{K}_P\right\| + \|\tilde{x}_i\|\right)$.[2]. Using the triangle inequality $\left\|\hat{x}^{\mathrm{T}}(t_i)\hat{M}(t)\hat{x}(t_i)\right\| \leq \left\|\sigma_{\delta'}\left(\hat{x}(t_i), u_i\right)\hat{W}'(t)\right\| + \left\|g\left(\tilde{K}_P(t), \tilde{x}(t_i)\right)\right\|$.

Equivalence of matrix norms implies that there exists $c > 0$ such that if $\left|\hat{M}_{j,k}\right| \leq \varpi/c$ for all $j, k = 1, \cdots, n$ then $\left\|\hat{M}\right\| \leq \varpi$. If t and t_i are large enough, then an argument similar to the proof of Theorem 2.2.1 can be used to show that $\left|\hat{M}_{j,k}\right| \leq \varpi/c, \forall j, k = 1, \cdots, n$. Indeed let e_i be the basis vector of zeros with a one in the i^{th} position. For a fixed j ane k, selecting constants $\alpha_{1,j,k} \cdots \alpha_{N,j,k} \in \mathbb{R}$ and rewriting (2.2.1) with $\hat{M} = \left(A^{\mathrm{T}}\hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^{\mathrm{T}}\hat{S} + \hat{Q}\right)$, we have $\sum_{i=1}^N \alpha_{i,j,k}\hat{x}^{\mathrm{T}}(t_i)\hat{M}x(t_i) = \sum_{i=1}^N \sum_{p=1}^n \sum_{q=1}^n \alpha_{i,j,k}\hat{x}_p(t_i)\hat{M}_{p,q}\hat{x}_q(t_i) = \sum_{i=1}^N \sum_{p=1}^n \hat{M}_{p,q} \sum_{q=1}^n \alpha_{i,j,k}\hat{x}_p(t_i)\hat{x}_q(t_i)$.

If $\mathrm{span}\{\hat{x}(t_i)\hat{x}^{\mathrm{T}}(t_i)\}_{i=1}^N = \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\mathrm{T}}\}$ holds then for any fixed j, k, we can select $\{\alpha_{i,j,k}\}_{i=1}^N$ such that $\sum_{i=1}^N \alpha_{i,j,k}\hat{x}(t_i)\hat{x}^{\mathrm{T}}(t_i) = e_j e_k^{\mathrm{T}} + e_k e_j^{\mathrm{T}}$, that is,

$$\sum_{i=1}^N \alpha_{i,j,k}\hat{x}(t_i)\hat{x}^{\mathrm{T}}(t_i) = \begin{cases} 1 & \text{if } p = j, q = k \\ 1 & \text{if } p = k, q = j \\ 0 & \text{otherwise} \end{cases} \tag{2.2.22}$$

As a result, $\sum_{i=1}^N \sum_{p=1}^n \hat{M}_{p,q} \sum_{q=1}^n \alpha_{i,j,k}\hat{x}_p(t_i)\hat{x}_q(t_i) = e_k^{\mathrm{T}}\hat{M}e_j + e_j^{\mathrm{T}}\hat{M}e_k = \hat{M}_{j,k} + \hat{M}_{k,j} = 2\hat{M}_{j,k}$. If t is large enough and as a result $\Delta(t)$ is small enough so that $\left\|\hat{x}^{\mathrm{T}}(t_i)\hat{M}(t)\hat{x}(t_i)\right\| \leq$

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

$\|\Delta(t)\| \leq \frac{2\varpi}{c \max_{i,j,k}(\{|\alpha_{i,j,k}|\}_{i=1}^{N})N}$ for all $i = 1, \cdots, N$, then $\left\|\sum_{i=1}^{N} \alpha_{i,j,k}\hat{x}^{\mathrm{T}}(t_i)\hat{M}\hat{x}(t_i)\right\| \leq \max_{i,j,k}(\{|\alpha_{i,j,k}|\}_{i=1}^{N})N \max_i \left(\left\{\left\|\hat{x}^{\mathrm{T}}(t_i)\hat{M}(t)\hat{x}(t_i)\right\|\right\}_{i=1}^{N}\right) \leq \frac{2\varpi}{c}$. As a result, $\left|\hat{M}_{j,k}\right| \leq \frac{\varpi}{c}$, and therefore $\left\|\hat{M}\right\| \leq \varpi$, and the corollary is established.

■

2.3 Simulations

In this section, the efficacy of the developed method is demonstrated using a linearly transformed separable state space model that admits non-unique solutions. That model is then modified to admit a unique solution per [15]. A noisy simulation for the non-unique model is then evaluated. Finally, the transferability of equivalent solutions of the IRL problem to different dynamical systems is examined.

2.3.1 A linear IRL poblem with nonunique solutions

In this section, we construct an academic example that ensures non-uniqueness of IRL solutions using the procedure developed in [15].

The state space model is given by

$$A = \begin{bmatrix} -0.2 & 0.4 & 1.6 \\ 3.7 & 1.6 & -3.1 \\ -3.2 & 0.4 & 4.6 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & -1 \\ -1 & 3 & 4 \\ 1 & 2 & -3 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.7 & -0.4 & -1.1 \\ -0.1 & 0.2 & 0.3 \\ 0.5 & 0 & -0.5 \end{bmatrix}.$$

The expert implements a feedback policy that minimizes the cost function in (2.1.3) with[3]

[3]The notation diag(v) represents a diagonal matrix with the elements of the vector v along the diagonal.

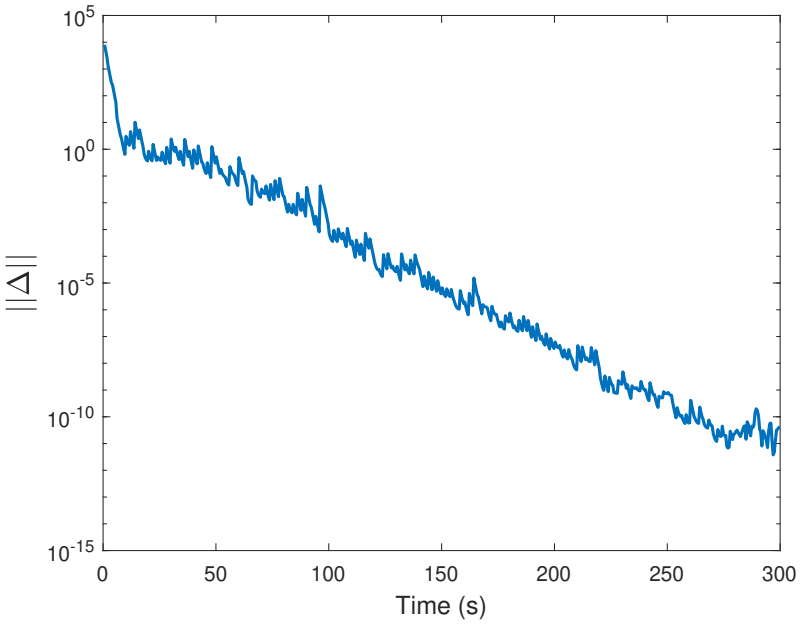


Figure 1: A logscale plot of the norm of Δ as a function of time for the example that admits non-unique solutions.

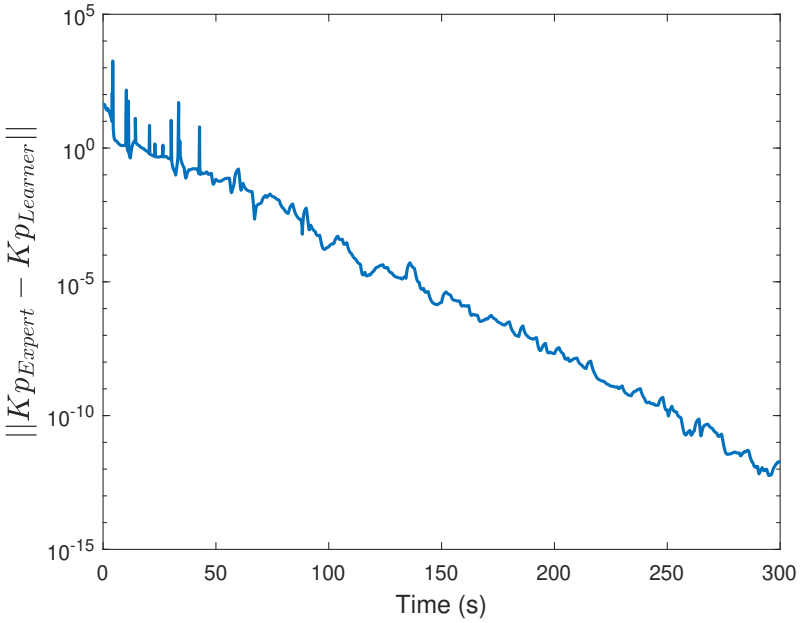


Figure 2: A logscale plot of the induced 2-norm of the error between the estimated feedback gain and the expert's feedback gain as a function of time for the example that admits non-unique solutions.

$$Q = \begin{bmatrix} 12.32 & -2.74 & -8.26 \\ -2.74 & 0.68 & 1.82 \\ -8.26 & 1.82 & 5.68 \end{bmatrix} \text{ and } R = \text{diag}([1, 4, 7]). \tag{2.3.1}$$

To ensure that the history stack satisfies the sufficient condition in (2.2.17), we construct an excitation signal comprised of a sum of 20 sinusoids with 0.5 magnitude and randomly selected frequencies and phases ranging from $0.001Hz$ to $1Hz$ and 0rad to πrad, respectively. This excitation signal is added into the learner system's input (2.2.13) and into the expert system's input (2.1.1). Data are added to the history stack every 0.05 seconds and is purged when full if the condition number of $\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I < 1 \times 10^5$, or 2 seconds have elapsed since the last purge, see [16] for a similar condition number minimization process. A Luenberger observer is utilized for state estimation by selecting the gain K_3 to place the poles of $(A - K_3C)$ at $p_1 = -0.1$, $p_2 = -1.5$ and $p_3 = -2$ using the MATLAB "place" command. The parameters of the NHSO are held constant for all simulations in this chapter unless otherwise stated.

As predicted by Theorem 2.2.1, Fig. 1 demonstrates Δ converges to zero and thus, the feedback matrix corresponding to the estimated weights, $\hat{W}$, converges to a neighborhood of the feedback matrix of the expert, as demonstrated in Fig. 2. Finally, Fig. 3 indicates that the cost functional converges to a functional that is different from that of the expert, confirming the existence of multiple equivalent solutions.

To demonstrate the sufficient condition detailed in Definition 2.2.2, a stem plot is generated that equals 1 when $\Sigma_u \in \text{Range}(\hat{\Sigma})$ and equals zero otherwise. This condition on Σ_u, as shown in the stem plot, is obtained through the application of the rank nullity theorem to the FI condition. The fact that convergence is obtained without the FI condition is indicative of the FI condition being sufficient and not necessary.

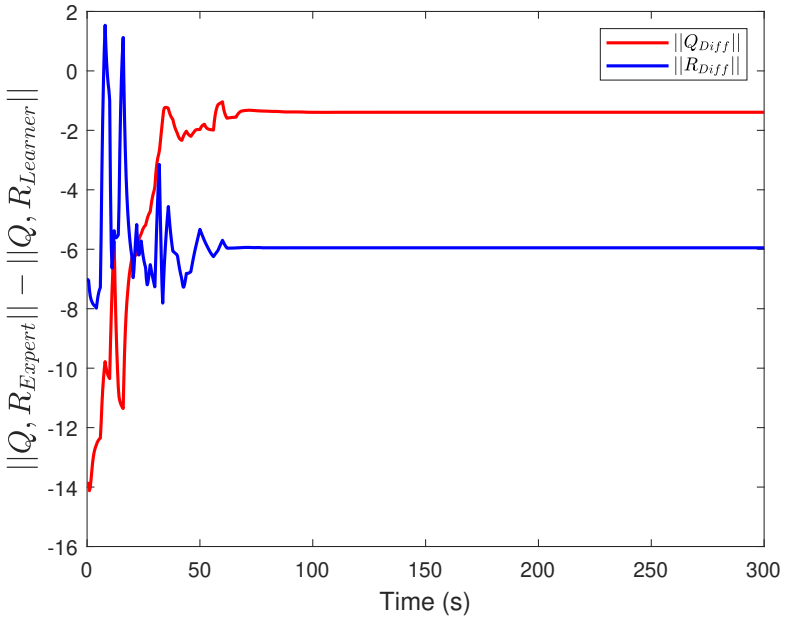


Figure 3: A plot of the induced 2-norm of the error between the estimated Q (red) and R (blue) matrices and the expert's Q and R matrices as a function of time for the example that admits non-unique solutions

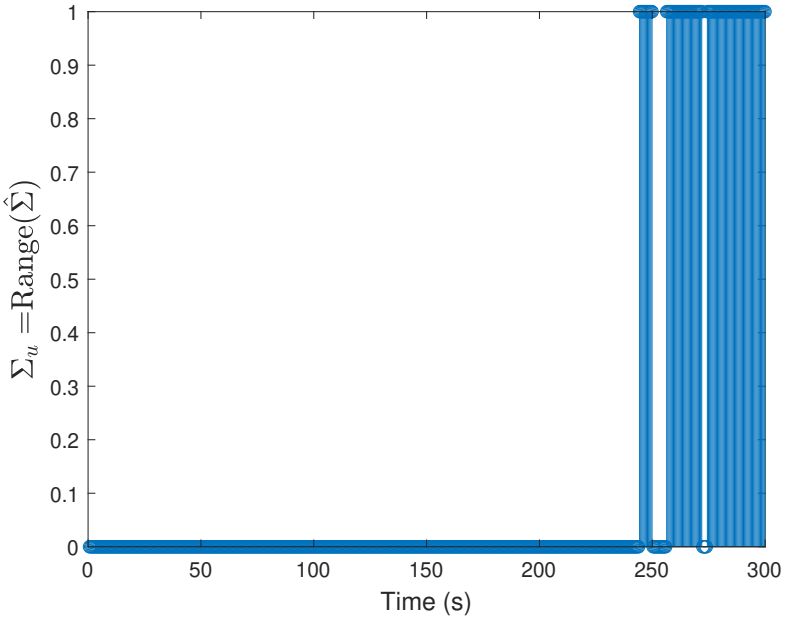


Figure 4: This stem plot to tracks the FI condition by plotting 1 whenever $\Sigma_u = \text{Range}(\hat{\Sigma})$ and 0 otherwise.

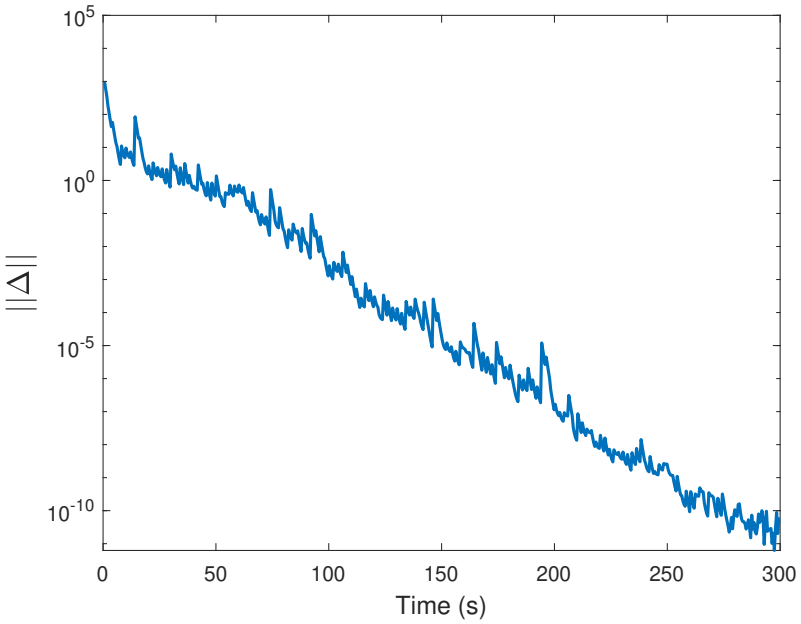


Figure 5: A logscale plot of the norm of Δ as a function of time for the example that admits a unique solution.

2.3.2 A linear IRL problem with a unique solution

If the system matrix for the system in Section 2.3.1 is changed to

$$A = \begin{bmatrix} 1 & 0.4 & 1.6 \\ 3.7 & 1.6 & -3.1 \\ -3.2 & 0.4 & 4.6 \end{bmatrix}$$

the state space model is no longer separable and thus the corresponding IRL problem admits a unique solution. Similar to the non-unique example, convergence of Δ to zero per Theorem 2.2.1 is observed in Fig. 5. Exact convergence of the learner's feedback matrix to the expert's is observed in Fig. 6. Fig. 7 indicates that when the IRL problem has a unique solution, the HSO developed in this chapter also recovers the true cost functional. As such, the HSO developed here is an extension of the HSO in [33] that applies to IRL problems with unique and non-unique solutions.

	State	Weight
Process Noise Cov Matrix	$0.001I$	$0.001I$
Measurement Noise Cov Matrix	R_1, R_2, R_3	$50I$

Table 1: The measurement and process noise matrices used in the continuous time Kalman filter implementation.

2.3.3 Kalman gain and the effects of measurement noise

This simulation provides insight into the noise robustness of the NHSO and its Kalman filter implementation (NHSO-KF). This investigation is purely heuristic in nature as the analysis requires for K_4 in (2.2.15) to be some matrix that is a scalar times an identity matrix. For the simulation with noise, $K_4(\hat{\Sigma}^{\mathrm{T}}\hat{\Sigma} + \epsilon I)^{-1}\hat{\Sigma}^{\mathrm{T}}$ is replaced with the Kalman gain, see Table 1, (NHSO-KF) and is then compared to K_4 selected as the identity matrix (NHSO). Zero-mean Gausian noise is added to y' with three separate noise variances, $R_1 = \mathrm{diag}([0.01^2, 0.01^2, 0.01^2])$, $R_2 = \mathrm{diag}([0.1^2, 0.1^2, 0.1^2])$, and $R_3 = \mathrm{diag}([0.5^2, 0.5^2, 0.5^2])$. Fifty Monte-Carlo simulations for each noise level are conducted and compared.

The same model and simulation setup as Section 2.3.1 is used in this section with the magnitude of the excitation signal modified from 0.5 to 1, when the R_3 noise metric is implemented, for improved convergence.

The recovered optimal trajectory under the learned cost function is compared against the expert's optimal trajectory in Fig. 9. The difference in optimal trajectories for each noise standard deviation is illustrated. This figure shows advantage of the Kalman gain implementation with noise rejection compared to the Luenberger observer implementation. Further supporting the noise advantage of the Kalman gain, Fig. 8 shows the difference between the expert's feedback gain matrix and the learner's feedback gain matrix for the last 30 seconds of the learning process. The calculations for Fig. 9 is $||X_{Expert} - X_{Learner}||$ and is similar for Fig. 8. The results are promising and indicate that a Kalman gain can be

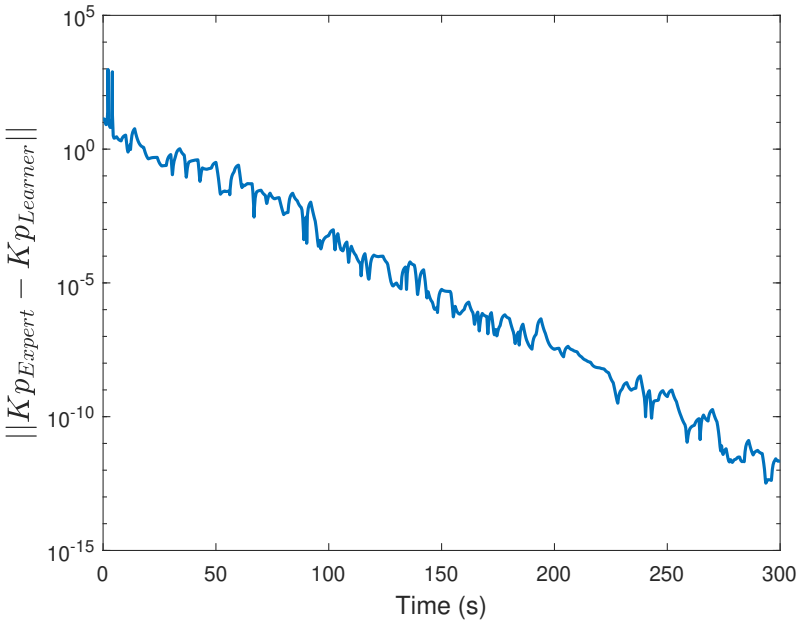


Figure 6: A logscale plot of the induced 2-norm of the error between the estimated feedback gain and the expert's feedback gain as a function of time for the example that admits a unique solution.

applied to reduce the error between the learner's feedback matrix and the expert's feedback matrix in the presence of noise.

2.3.4 Transferable Equivalent Solutions

As it is widely recognized that the most succinct representation of the behavior of an expert is encoded in its cost function [25]. This sections aims to identify the transferability of non-unique solutions of the IRL problem.

To further clarify, let there be two systems (A_1, B_1) and (A_2, B_2) that have their respective feedback gain matrices, K_{EP1} and K_{EP2} that optimize the expert's cost function weights of Q^*, S^*, and R^*. Their respective equivalent solutions are characterized as $(\hat{Q}_1, \hat{S}_1, \hat{R}_1)$ and $(\hat{Q}_2, \hat{S}_2, \hat{R}_2)$. Now, a transferred feedback matrix $\hat{K}_{P1,2}$ can be generated using the system (A_1, B_1) with the equivalent solution $(\hat{Q}_2, \hat{S}_2, \hat{R}_2)$. Also, a transferred feedback matrix $\hat{K}_{P2,1}$ can be generated using (A_2, B_2) paired with the equivalent solution $(\hat{Q}_1, \hat{S}_1, \hat{R}_1)$. The

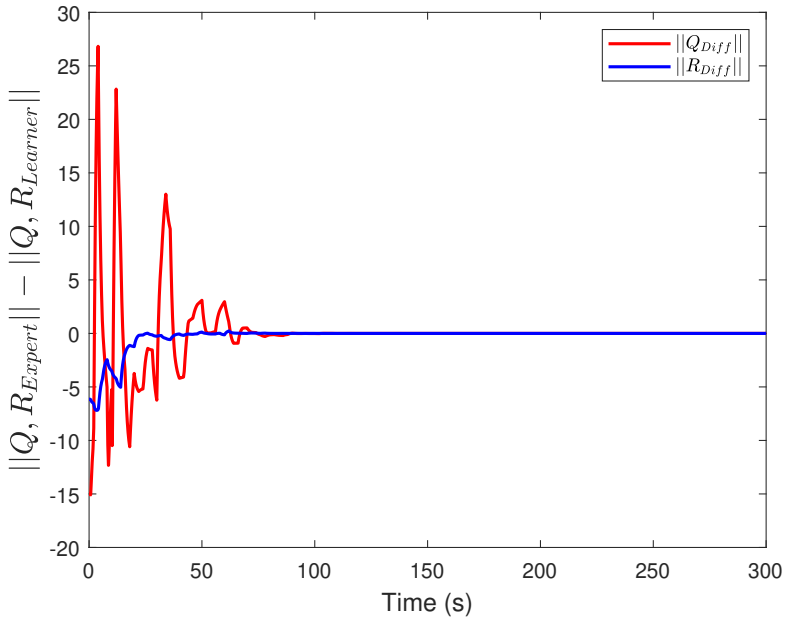


Figure 7: A plot of the induced 2-norm of the error between the estimated Q (red) and R (blue) matrices and the expert's Q and R matrices as a function of time for the example that admits a unique solution

transferred matrices $\hat{K}_{P1,2}$ and $\hat{K}_{P2,1}$ are then compared against the expert matrices K_{EP1} and K_{EP2}, respectively.

To gauge transferrability, we run five simulations using the same setup as Section 2.3.1 with randomly generated A and B matrices that non-unique solutions (per [15]) for the resulting IRL problem. Each system has the same expert using the optimal reward weights Q^*, S^* and R^*. After finding an equivalent solution of $(\hat{Q}, \hat{S}, \hat{R})$, each equivalent solution is paired with every randomly generated A and B matrix to generate 20 transferred feedback gain matrices. The feedback gain matrices are then compared to the expert's feedback gain for the corresponding system. The average over all 20 combinations of the difference between the transferred feedback gain matrices and the expert's feedback gain matrices is 2.0866×10^{-8}.

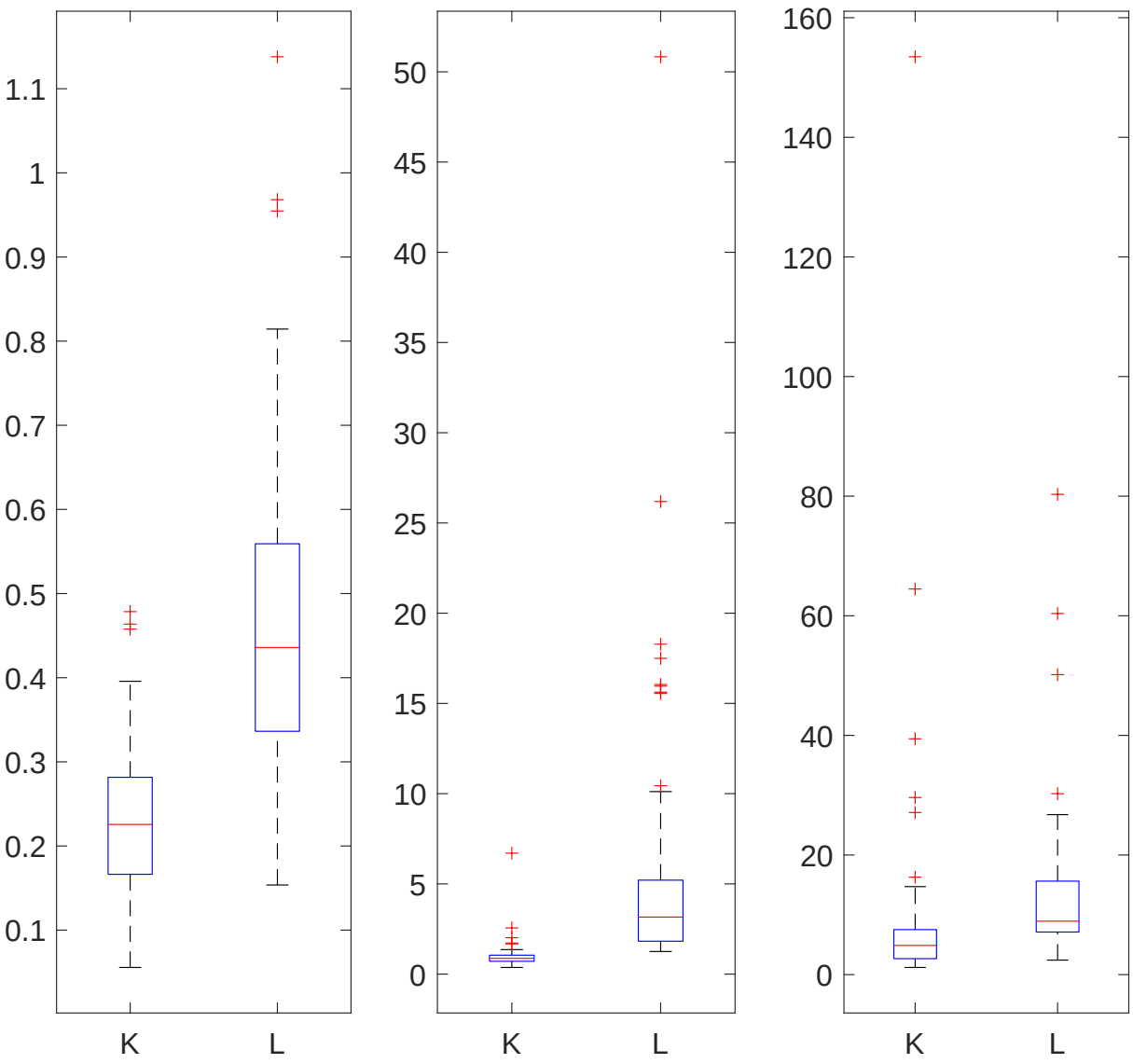
2.3.5 Discussion

1. Each simulation shows the convergence of Δ to zero and the convergence of the learner's feedback matrix, $\hat{K}_P$, to the expert's feedback matrix, K_{EP}. The metric to judge an equivalent solution is then able to be appropriately encoded in Δ to gauge the closeness of the learned equivalent solution to the expert's solution, for the corresponding IRL problem.

2. In all simulations, the NHSO converges to either an equivalent solution or a unique solution. The solution type is dependent on the structure of the system matrices. Therefore, the NHSO is a complete extension to the HSO [33] as it works IRL problems with both unique and non-unique solutions.

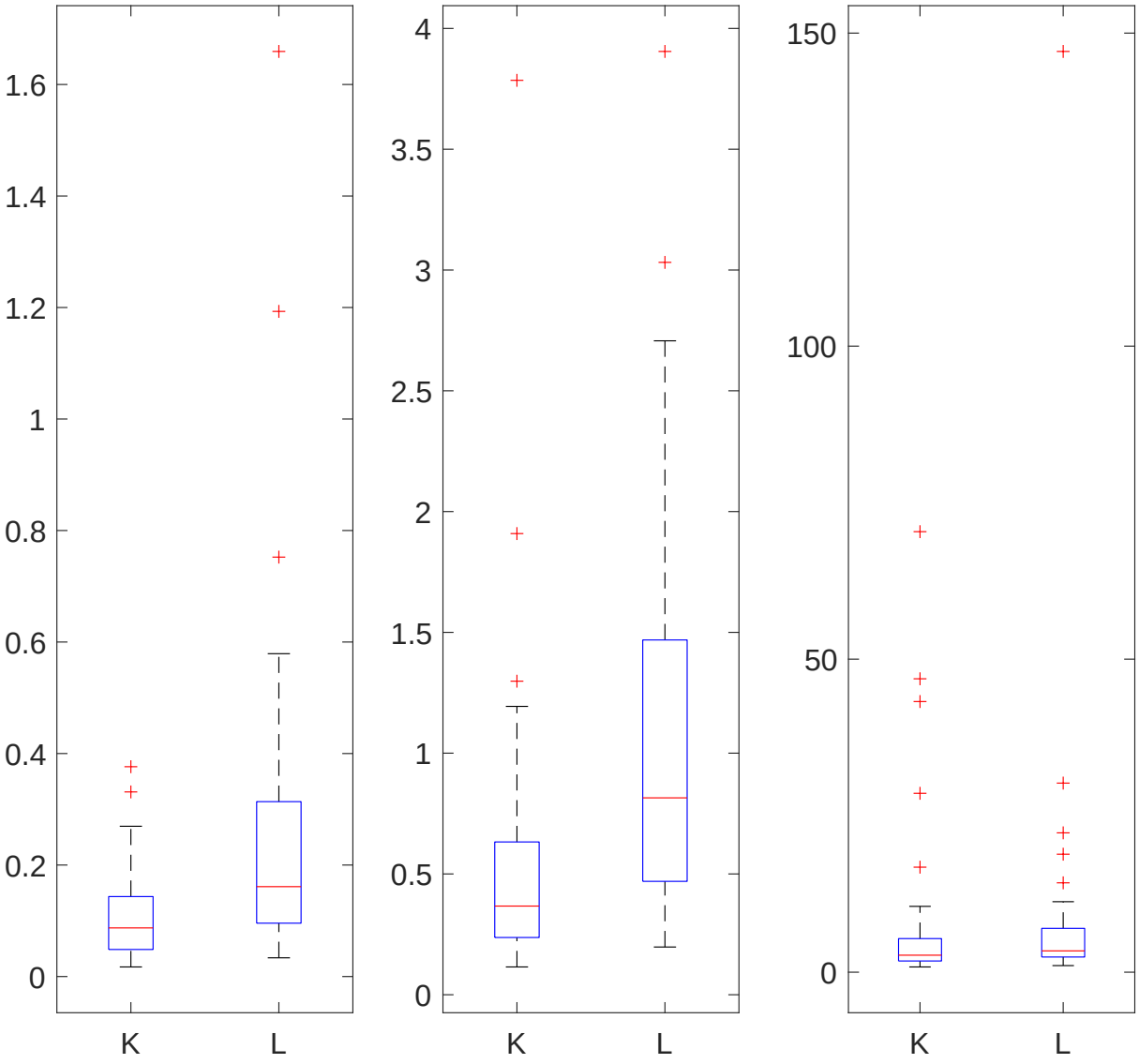
3. The transferability of equivalent solutions is briefly tested. This result is is interesting in that an equivalent solution to the IRL problem for one system, is also equivalent to the solutions for different systems as long as the same original cost function, or equivalent cost function, was utilized for each system. The implication is that a set of equivalent solution may be identifiable. We hypothesize that this transferability of equivalent solutions can be attributed to the product structure of all five systems being identical.

4. If ϵ is selected according to Theorem 2.2.1, then regardless of the size of ϵ, Δ converges to zero and a unique or non-unique solution is obtained. This result is surprising as the NHSO utilizes a regression technique. It is beneficial to understand that using even the best Σ_u and $\hat{\Sigma}$, an offline ridge regression technique [35] cannot find a $\hat{W}$ that constitutes an equivalent solution to the IRL problem.



(a) $SD = 0.01$ (b) $SD = 0.1$ (c) $SD = 0.75$

Figure 8: Boxplot of error between expert's feedback gain and learner's feedback gain for the last 30 seconds for three separate standard deviations (SD) of noise added to the measurement for a Luenberger observer (L) and Kalman gain (K).



(a) $SD = 0.01$ (b) $SD = 0.1$ (c) $SD = 0.75$

Figure 9: Boxplot of the error between the expert's trajectory and the learner's trajectory for the entire simulation time for three seperate standard deviations (SD) of noise added to the measurement with both a Luenberger observer (L) and Kalman gain (K).

CHAPTER III

PILOT PERFORMANCE MODELING VIA OBSERVER-BASED INVERSE REINFORCEMENT LEARNING

The focus of this chapter is behavior modeling for pilots of unmanned aerial vehicles. The pilot is assumed to make decisions that optimize an unknown cost functional, which is estimated from observed trajectories using a novel inverse reinforcement learning (IRL) framework. The resulting IRL problem often admits multiple solutions. In this chapter, a recently developed novel IRL observer is adapted to the pilot modeling problem. The observer is shown to converge to one of the equivalent solutions of the IRL problem. The developed technique is implemented on a quadcopter where the pilot is modeled as a linear quadratic regulator. Experimental results demonstrate the robustness of the method and its ability to learn an equivalent cost functional.

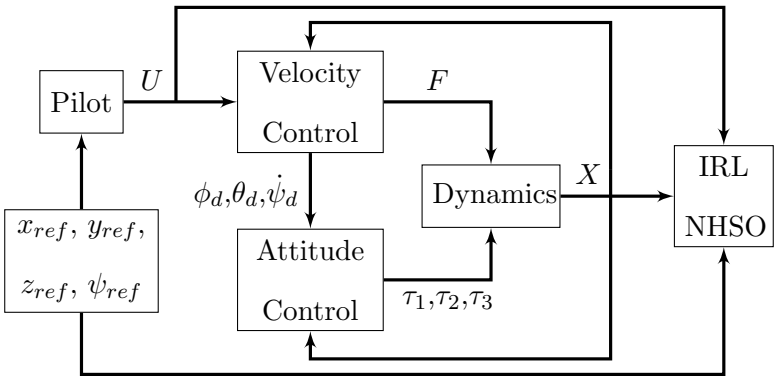


Figure 10: Pilot and Quadcopter Combined Model

3.1 Modeling

3.1.1 Problem formulation

This study concerns a quadcopter UAV 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 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 UAV 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 quadrotor are used to learn the surrogate pilot's cost functionals using an observer-based inverse reinforcement learning (IRL) algorithm. Since the IRL problem, as formulated in Section 3.1 admits multiple solutions, we aim to recover an equivalent cost functional per Definition 3.1.1.

3.1.2 Pilot Model

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

$$\dot{X}(t) = AX + BU, \tag{3.1.1}$$

where the state is $X \in \mathbb{R}^{12}$ and the control input is $U \in \mathbb{R}^4$. The system matrices are given as $A \in \mathbb{R}^{12\times 12}$ and $B \in \mathbb{R}^{12\times 4}$.

The pilot is assumed to be an optimal controller that optimizes the cost functional

$$J(X_0, U(\cdot)) = \int_0^\infty \left(X(t)^{\mathrm{T}} Q X(t) + U(t)^{\mathrm{T}} R U(t)\right) dt, \tag{3.1.2}$$

where $X(\cdot)$ is the system trajectory under the control signal $U(\cdot)$ and starting from the initial condition x_0, and $Q \in \mathbb{R}^{12\times12}$ and $R \in \mathbb{R}^{4\times4}$ are unknown positive semi-definite matrices.

Assumption 3.1.1 *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.*

The algebraic Riccati equation (ARE),

$$A^{\mathrm{T}}S + SA - SBR^{-1}B^{\mathrm{T}}S + Q = 0, \tag{3.1.3}$$

with respect to the optimal control problem described by (3.1.1) and (3.1.2) can then be solved, which yields the policy of the pilot given by $u = K_{EP}x$.

The pilot's policy is recovered by estimating, online, and in real-time, the unknown matrices using the known system matrices, A, B, and C, given X and U.

3.1.3 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 inputs, and aim to model the cost functional of a surrogate LQR pilot that generates velocity commands that are then implemented by the autopilot.

The model used in this study closely follows the development in [5, 6, 13] The state variables of the model are

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

where x, y, and z, are the translational positions, $\dot{x}$, $\dot{y}$, and $\dot{z}$, are the transnational velocities. Also, ϕ, θ, and ψ, are the roll pitch and yaw angular positions and $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ are their

respective angular velocities. The control input is given by

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

where $\dot{x}_d$, $\dot{y}_d$, and $\dot{z}_d$, are the desired translational velocities with $\dot{\psi}_d$ as the desired heading angular velocity. The translational dynamics of a quadcopter are described in the North, East, Down (NED) coordinate frame by [13]

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix} - k_t \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{3.1.4}$$

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

$$R = \begin{bmatrix} 1 & \phi\theta - \psi & \theta + \phi\psi \\ \psi & \phi\theta\psi + 1 & \theta\psi - \phi \\ -\theta & \phi & 1 \end{bmatrix}. \tag{3.1.5}$$

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

$$F = mg + mk_{p_{13}}(\dot{z} - \dot{z}_d). \tag{3.1.6}$$

The rotational motion of the quadcopter is described by [5, 6]

$$\begin{aligned} \ddot{\phi} I_{xx} &= \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + l\tau_1 \\ \ddot{\theta} I_{yy} &= \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + l\tau_2 \\ \ddot{\psi} I_{zz} &= \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + \tau_3 \end{aligned} \tag{3.1.7}$$

with I_{xx}, I_{yy}, and I_{zz} being moment of inertia and τ_1, τ_2, and τ_3 being torques designed as

$$\begin{aligned} \tau_1 &= k_{p_{21}}(\phi_d - \phi) - k_{d_1}\dot{\phi}, \\ \tau_2 &= k_{p_{22}}(\theta_d - \theta) - k_{d_2}\dot{\theta}, \\ \tau_3 &= k_{d_3}(\dot{\psi}_d - \dot{\psi}). \end{aligned} \tag{3.1.8}$$

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

$$\begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{k_{p_{12}}(\dot{y}_d-\dot{y})\sin\psi+k_{p_{11}}(\dot{x}_d-\dot{x})\cos\psi}{g+k_{p_{13}}(\dot{z}_d-\dot{z})}\right) \\ \arctan\left(cos\theta_d\frac{k_{p_{11}}(\dot{x}_d-\dot{x})\sin\psi-k_{p_{12}}(\dot{y}_d-\dot{y})\cos\psi}{g+k_{p_{13}}(\dot{z}_d-\dot{z})}\right) \end{bmatrix} \tag{3.1.9}$$

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 with small angle approximations and a linear approximation for the inverse tangent function [27] to yield

$$\begin{aligned} \theta_d &= \frac{\pi}{4}\left(\frac{k_{p_{12}}(\dot{y}_d-\dot{y})\psi + k_{p_{11}}(\dot{x}_d-\dot{x})}{g+k_{p_{13}}(\dot{z}_d-\dot{z})}\right), \\ \phi_d &= \frac{\pi}{4}\left(\frac{k_{p_{11}}(\dot{x}_d-\dot{x})\psi - k_{p_{12}}(\dot{y}_d-\dot{y})}{g+k_{p_{13}}(\dot{z}_d-\dot{z})}\right). \end{aligned} \tag{3.1.10}$$

Linearizing (3.1.4) and (3.1.7) about the origin, while using (3.1.6), (3.1.8), and (3.1.10), yields the linear system

$$\begin{aligned} \ddot{x} &= -g\theta - \frac{k_t}{m}\dot{x} \\ \ddot{y} &= g\phi - \frac{k_t}{m}\dot{y} \\ \ddot{z} &= k_{p_{13}}(\dot{z}_d-\dot{z}) - \frac{k_t}{m}\dot{z} \\ \ddot{\phi} &= \frac{b_1\pi k_{p_{21}}k_{p_{12}}(\dot{y}-\dot{y}_d)}{4g} - b_1k_{d_1}\dot{\phi} - b_1k_{p_{21}}\phi \\ \ddot{\theta} &= \frac{b_2\pi k_{p_{22}}k_{p_{11}}(\dot{x}_d-\dot{x})}{4g} - b_2k_{d_2}\dot{\theta} - b_2k_{p_{22}}\theta \\ \ddot{\psi} &= b_3k_{d_3}(\dot{\psi}_d-\dot{\psi}) \end{aligned} \tag{3.1.11}$$

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 pictured in Figure 10, given measurements of the state variables, i.e., translational position $[x, y, z]$, translational velocities $[\dot{x}, \dot{y}, \dot{z}]$, angular position $[\phi, \theta, \psi]$, and 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 LQR surrogate pilot, we aim to find an equivalent solution ($\hat{Q}$, $\hat{S}$, $\hat{R}$) according to the following definition based on [36].

Definition 3.1.1 *A solution ($\hat{Q}$, $\hat{S}$, $\hat{R}$) is called an equivalent solution of the IRL problem if it satisfies the ARE*

$$A^{\mathrm{T}}\hat{S} + \hat{S}A - \hat{S}B\hat{R}^{-1}B^{\mathrm{T}}\hat{S} + \hat{Q} = 0 \tag{3.1.12}$$

and optimization of the performance index J, with $Q = \hat{Q}$ and $R = \hat{R}$, results in the same feedback matrix as the one utilized by the pilot, that is,

$$\hat{K}_P \coloneqq \hat{R}^{-1} B^{\mathrm{T}} \hat{S} = K_{EP}.$$

3.2 Inverse Reinforcement Learning

This section contains the IRL algorithm used in identifying the pilot's cost function, see [36] for further details.

3.2.1 Regularized History Stack Observer for Non-Unique Solutions (NHSO)

The following development is a special case of the NHSO developed in [36], where the system state is assumed to be measurable. The state estimates generated by the onboard Kalman filter are used in the experiment to obtain an equivalent solution per Definition 3.1.1, the NHSO is constructed as follows.

If the pilot model developed in Section 3.1 follows Assumption 3.1.1, and if the pilot's states, X, and inputs, U, are optimal with respect to the cost functional in (3.1.2), then there exists a matrix S such that Q, R, A, B, and S satisfy the Hamilton-Jacobi-Bellman (HJB) equation

$$X^{\mathrm{T}} \left(A^{\mathrm{T}} S + SA - SBR^{-1} B^{\mathrm{T}} S + Q \right) X = 0 \tag{3.2.1}$$

for all $x \in \mathbb{R}^{12}$ and the optimal control equation

$$U(t) = -R^{-1} B^{\mathrm{T}} S X(t) \tag{3.2.2}$$

$\forall t \in \mathbb{R}_{\geq 0}$. However, the linear system in (3.1.11) has a product structure and therefore admits multiple solutions to the IRL problem [15]. From Definition 3.1.1, the pilot's feedback matrix given by $K_{Ep} = R^{-1} B^{\mathrm{T}} S$ is also be given by $K_{EP} = \hat{R}^{-}1 B^{\mathrm{T}} \hat{S} = \hat{K}_P$, where $\hat{R}$ and $\hat{S}$ are part of an equivalent solution to the IRL problem and $\hat{K}_P$ is the learned feedback matrix of the pilot.

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, (3.2.1) and (3.2.2) can be evaluated to develop an observation error that evaluates to zero if the estimates of the matrices Q, R, and S are correct. The final form of the NHSO without state estimation is

$$\dot{\hat{W}} = (\Sigma^{\mathrm{T}}\Sigma + \epsilon I)^{-1}\Sigma^{\mathrm{T}}\left(\Sigma_u - \Sigma\hat{W}\right) \tag{3.2.3}$$

where $\hat{W} = [\hat{W}_S, \hat{W}_Q, \hat{W}_R^-]$ are the weights of $\hat{Q}$, $\hat{R}$, and $\hat{S}$ with the first value of R, r_1, removed for scaling ambiguity. Theorem 3.2.1, which guarantees convergence of (3.2.3) to an equivalent solution, relies on the formulation of an error metric $\Delta(t) \coloneqq \Sigma_u - \Sigma\hat{W}(t)$ and its subsequent time derivative,

$$\dot{\Delta} = -\Sigma(\Sigma^T\Sigma + \epsilon I)^{-1}\Sigma^T\Delta, \tag{3.2.4}$$

along with the following data informativity condition.

Assumption 3.2.1 *The signal (X, U) is finitely informative (FI) if there exists a time instance $T > 0$ such that for some $\{t_1, t_2, \ldots, t_N\} \subset [0, T]$,*

$$\operatorname{span}\{X(t_i)\}_{i=1}^N = \mathbb{R}^n$$

$$\operatorname{span}\left\{X(t_i)X(t_i)^{\mathrm{T}}\right\}_{i=1}^N = \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\mathrm{T}}\} \quad \textit{and}$$

$$\Sigma_u \in (\operatorname{Null}(\hat{\Sigma}^{\mathrm{T}}))^{\perp}. \tag{3.2.5}$$

Theorem 3.2.1 *If $\Sigma_u \in \operatorname{Null}(\Sigma^{\mathrm{T}})^{\perp}$ and $\epsilon \geq 0$ is selected to ensure invertibility of $\Sigma^T\Sigma + \epsilon I$, then the solutions of (3.2.4) satisfy $\lim_{t\to\infty}\Delta(t) = \{0\}$.*

In addition $\Delta = \Sigma_u - \Sigma\hat{W} = 0$, $\operatorname{span}\{X_i\}_{i=1}^N = \mathbb{R}^n$, and if $\operatorname{span}\{X_iX_i^{\mathrm{T}}\}_{i=1}^N = \{\mathbb{Z} \in \mathbb{R}^{n\times n} | \mathbb{Z} = \mathbb{Z}^{\mathrm{T}}\}$, then the matrices $\hat{Q}$, $\hat{S}$, and $\hat{R}$, extracted from $\hat{W}$, constitute an equivalent solution of the IRL problem per Definition 3.1.1.

The history stack is constructed as

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

During the learning process, two separate sets of the history stacks are maintained, H_1 and H_2, that each contain a pair of Σ_u and Σ. Both history stacks are initialized as zero matrices where $\Sigma_u \in \mathbb{R}^{425}$ and $\Sigma \in \mathbb{R}^{425\times 85}$. Data are then added into H_2 at a set interval until it is filled, then a condition number minimization algorithm, similar to [16], that replaces old data with new data if the condition number will be lower is utilized until the condition number of $\Sigma^{\mathrm{T}}\Sigma + \epsilon\, \mathrm{I}$ is less than a set value or until a specified period has passed between purges. The history stack, H_2 is then transferred to H_1, overwriting the originally stored values. The matrices in H_1 are subsequently used in (3.2.3) with H_2 being reset to zero and the process for filling it begins again. See [36] for additional details regarding the manipulation of the history stacks.

Within the history stacks, Σ_u is known from the removed r_1 and given associated input U_1. Σ is constructed using data from the state space model such that

$$\sigma_\delta\left(X, U\right)\hat{W} = (AX + BU)^{\mathrm{T}}(\nabla_X \sigma_S(X))^{\mathrm{T}}\hat{W}_S + \sigma_Q(X)^{\mathrm{T}}\hat{W}_Q + \sigma_{R1}^-(U)^{\mathrm{T}}\hat{W}_R^-, \tag{3.2.6}$$

and

$$\sigma_{\Delta_u}\left(X, U\right)\hat{W} = B^{\mathrm{T}}(\nabla_X \sigma_S(X))^{\mathrm{T}}\hat{W}_S + 0_{m\times P_S+P_Q}\hat{W}_Q + 2\sigma_{R2}^-(U)\hat{W}_R^-, \tag{3.2.7}$$

where $(\hat{W}_S)^{\mathrm{T}}\sigma_S(X) = x^{\mathrm{T}}\hat{S}X$, $(\hat{W}_Q)^{\mathrm{T}}\sigma_Q(X) = X^{\mathrm{T}}\hat{Q}X$, $(\hat{W}_R)^{\mathrm{T}}\sigma_{R1}(u) = (U^-)^{\mathrm{T}}\hat{R}^-(U^-)$, and $\sigma_{R2}(U)\hat{W}_R^- = R^-U^-$, and $\hat{W}_S \in \mathbb{R}^{P_S}$, $W_Q^* \in \mathbb{R}^{P_Q}$, $\hat{W}_R^- \in \mathbb{R}^{M-1}$ are the ideal weights with $P_S = 78$, $P_Q = 4$, and $M = 4$ being the number of basis functions in the respective linear

parameterizations, see [33] for an exact characterization of the basis functions. The vector U^- represents the vector U with the first element removed and R^- represents the matrix R with the first column removed. These removed values are the data that are stored in Σ_u.

3.3 Experiments

Experimental results obtained using the developed NHSO on a quadcopter are presented in this section. The ability of the developed IRL method to learn the non-unique weights of the quadcopter pilot represented as an LQR controller is demonstrated.

3.3.1 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 orientation is captured through a motion capture system (OptiTrack) whereas angular velocity and acceleration are measured from an onboard inertial measurement unit (IMU). Both systems have their data fused in a Kalman filter for accurate state estimation. 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, $S.k_t = 0.01$, $g = 9.81$ 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 $k_{d_3} = 0.1$.

Remark 3.3.1 *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 controller presented in Section 3.1. The Px4 autopilot, while able to track a velocity input, cannot maneuver the real-life quadcopter as adeptly as the modeled controller can with the modeled quadcopter. To ensure that the closed-loop model presented in Section 3.1 fits the real quadcopters, the proportional and derivative gains in the model are adjusted so that the response of the model and the real quadcopter to velocity commands is as close as possible.*

3.3.2 Controller Implementation

The quadcopter is controlled through an offboard ground control station that implements the surrogate LQR pilot with the control policy that optimizes the cost function in (3.1.2) with[1]

$$Q = \text{diag}([9.57, 6.91, 2.84, 0, 0, 0, 0, 0, 11.68, 0, 0, 0]) \text{ and}$$
$$R = \text{diag}([9.57, 3.48, 14.40, 0.17]). \tag{3.3.1}$$

The cost function is designed under the assumption that the pilot only penalizes the translational position and heading with commanded translational velocities and heading angluar velocities, where the pilot is constructed using the linearized system in (3.1.11). To reduce the number of weights, the sparsity structure of Q and R is assumed to be known and only the nonzero elements of Q and R are estimated. The pairs (A, B) and $(A, \sqrt{Q})$ are confirmed to satisfy stabilizability and detectability through their respective PBH test [9, Theorem 14.3, 16.6].

To satisfy the FI condition in Assumption 3.2.1, the ground control station adds an excitation signal onto the commanded velocities so the final commanded velocity is

$$U_{cmd} = U_{exc} + U. \tag{3.3.2}$$

Where $U = -K_{EP}x$ is the command generated by the surrogate pilot, without the excitation, which is recorded in the history stacks and U_{exc} is the excitation signal.

3.3.3 Methods

The quadcopter for each of the 13 experiments started at a randomly generated hover point that is contained in the operating area. The surrogate LQR pilot then commands the quadcopter to fly to the origin with a z-offset where the pilot attempts to maintain the quadcopter's position irrespective of the excitation signals for 200 seconds. The input signal

[1]The notation diag(v) represents a diagonal matrix with the elements of the vector v along the diagonal.

in (3.1.1) is subjected to the excitation, U_{exc}, which is composed of 4 sets of 75 sinusoids. Each set spans a frequency range from $0.001Hz$ to $10Hz$, with a varying frequency and a magnitude of 0.03. The NHSO is implemented with regularization parameter $\epsilon = 0.002$, and data are stored in the history stacks at a rate of 0.08 seconds. The main history stack is purged if the auxiliary history stack is full and if 9 seconds have passed or the condition number minimization algorithm makes the condition number of $\Sigma^{\mathrm{T}}\Sigma + \epsilon I < 1 \times 10^9$. The initial guesses for the unknown weights are randomly generated with a normal distribution between $[-5, 5]$.

3.3.4 Results and Discussion

The experimental results in Figs. 11-15 are obtained from the same flight. The position of the quadcopter as a function of time is shown in Fig. 11, and the linear velocity of the quadcopter as a function of time is shown in Fig. 12. The quadcopter holds position at the origin with a z-offset of $1.5m$ and the velocity appears noisy due to the excitation signal. The convergence of Δ to the zero in Fig. 13[2], combined with the convergence of Q and R in Fig. 15 to some value, indicates that an equivalent solution per Definition 3.1.1 is discovered from Theorem 3.2.1. As such, the difference between the feedback matrices in Fig. 14 converges to zero.

Figs. 13 and 14 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 16, the cost functional recovered from data in each of the 13 experiments converges to one of the equivalent solutions. The particular equivalent solution recovered in each run depends on the initial guess of the unknown weights used in that run.

[2] $\|\cdot\|$ defines the euclidean norm when applied to a vector and the Frobenius norm when applied to a matrix.

	NHSO	HSO
Mean($\|K_{Ep} - K_p\|$)	2.7553e-08	NaN
Cov($\|K_{Ep} - K_p\|$)	2.1605e-15	NaN

Table 2: The NHSO and HSO [33] are evaluated by computing the mean and covariance of the Frobenius norm of the difference between the final values of the feedback matrices for the 13 tests.

From the 13 experiments, it is evident that NHSO finds equivalent solutions for the pilot modeling problem. A sufficiently excited system state is needed to meet the data sufficiency conditions in Assumption 3.2.1. In this effort, to achieve excitation, an excitation signal is added to the surrogate LQR pilot's command. The excitation signal is designed using trial and error. It is observed in Table 2 that the convergence is much greater for the quadcopter pilot modeling application than the simulation results in [36], as there is more information in the signals. Furthermore, as evidenced by Table 2, while the history stack observer (HSO) [33] diverges in this experiment due to nonuniqueness of solutions of the underlying IRL problem, the NHSO converges to an equivalent solution. Furthermore, the linearized quadcopter model does not capture the nonlinear dynamics or disturbances inherent in a real-world implementation of a quadcopter, however, as mentioned previously, these modeling differences were mitigated through tuning.

The tuning of the NHSO starts with choosing a large enough ϵ for invertibility while maintaining a fast convergence rate. 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 $\Sigma^{\mathrm{T}}\Sigma + \epsilon I$ whereas shorter intervals allow for faster population and purging of the history stacks, which results in the convergence not stagnating.

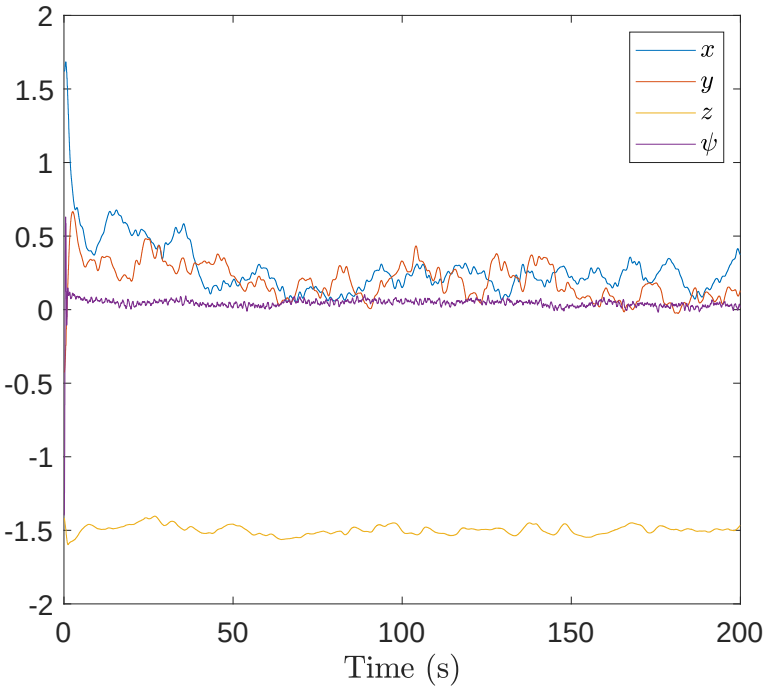


Figure 11: Position of the quadcopter for one experiment.

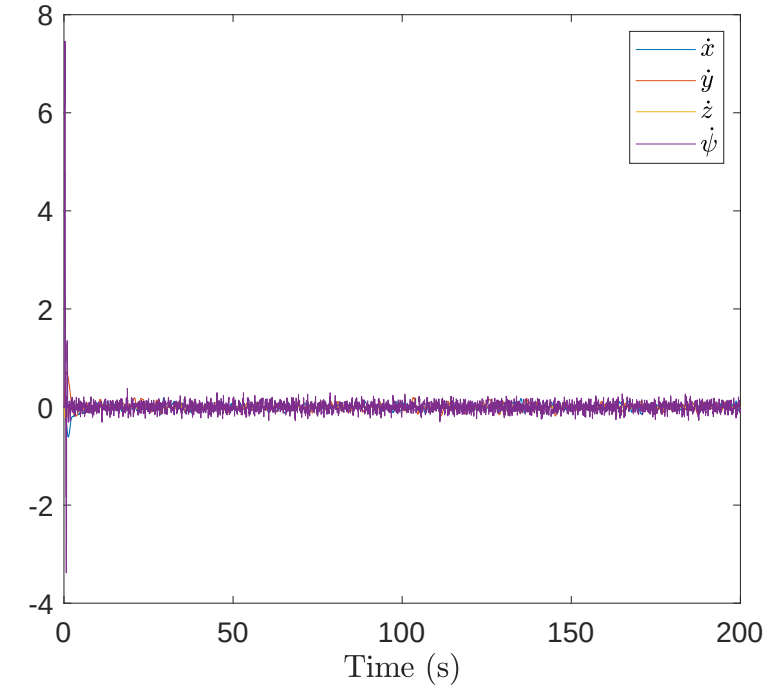


Figure 12: Velocity of the quadcopter for one experiment.

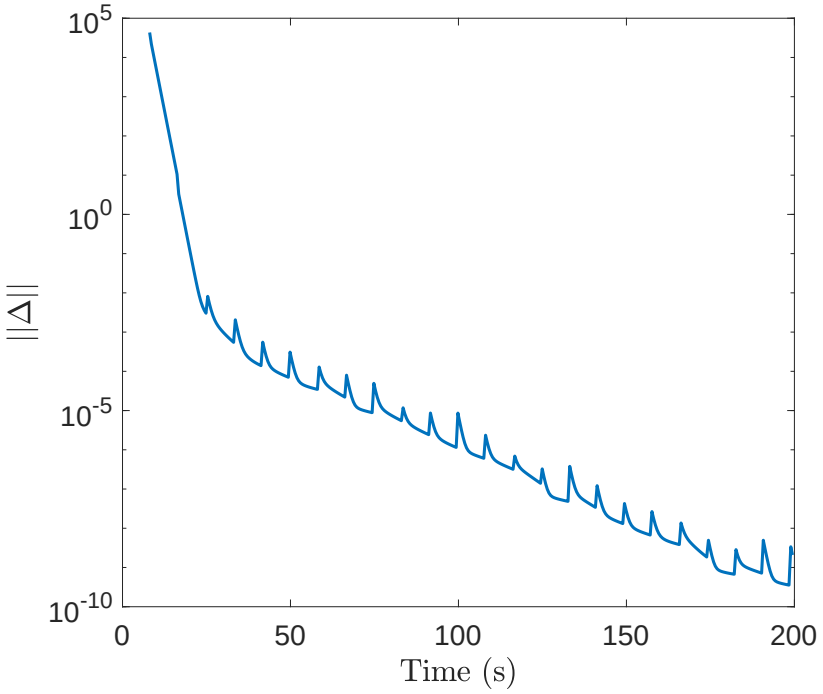


Figure 13: A logscale plot of the norm of Δ as a function of time throughout one experiment.

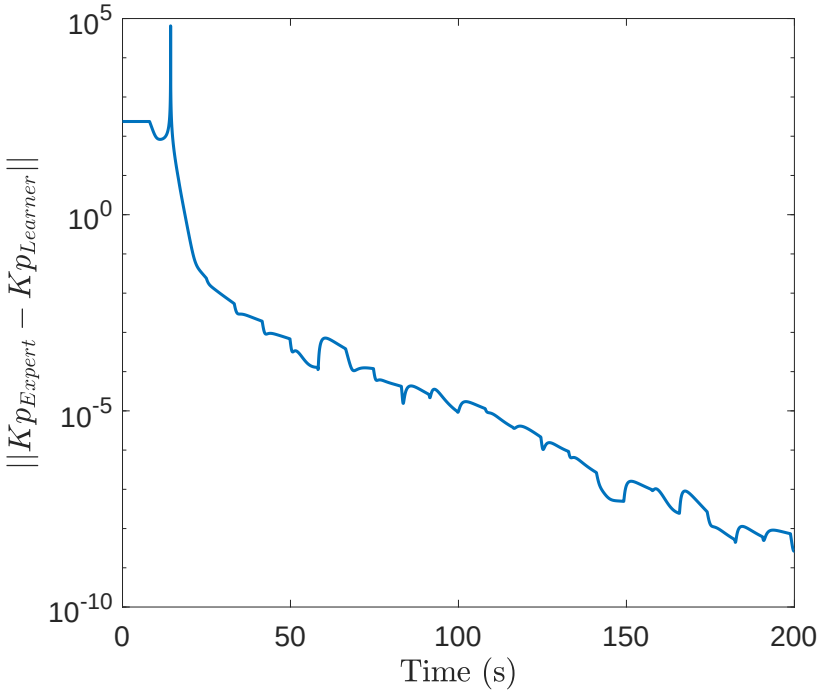


Figure 14: A logscale plot of the induced 2-norm of the error between the estimated feedback gain and the pilot's feedback gain as a function of time throughout one experiment.

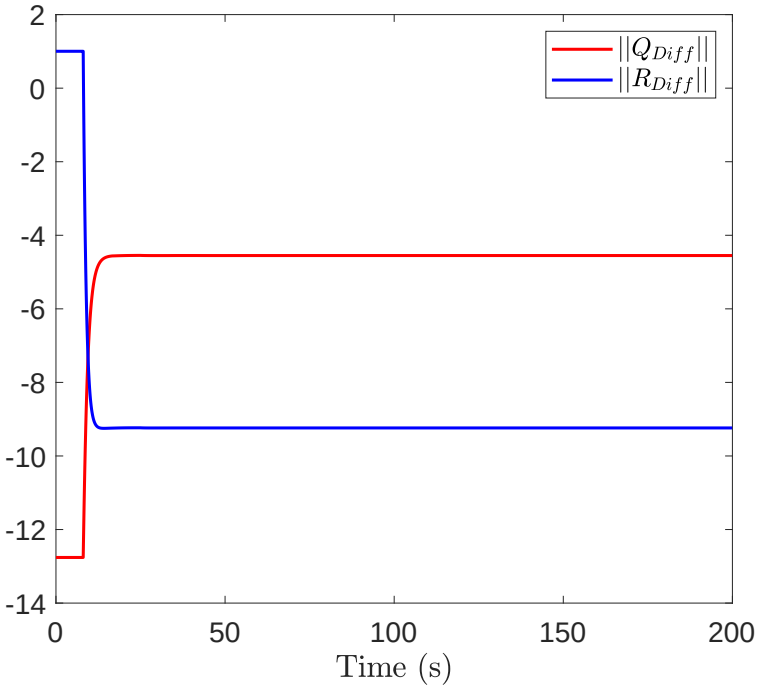


Figure 15: A plot of the induced 2-norm of the error between the estimated Q (red) and R (blue) matrices and the pilot's Q and R matrices as a function of time throughout one experiment.

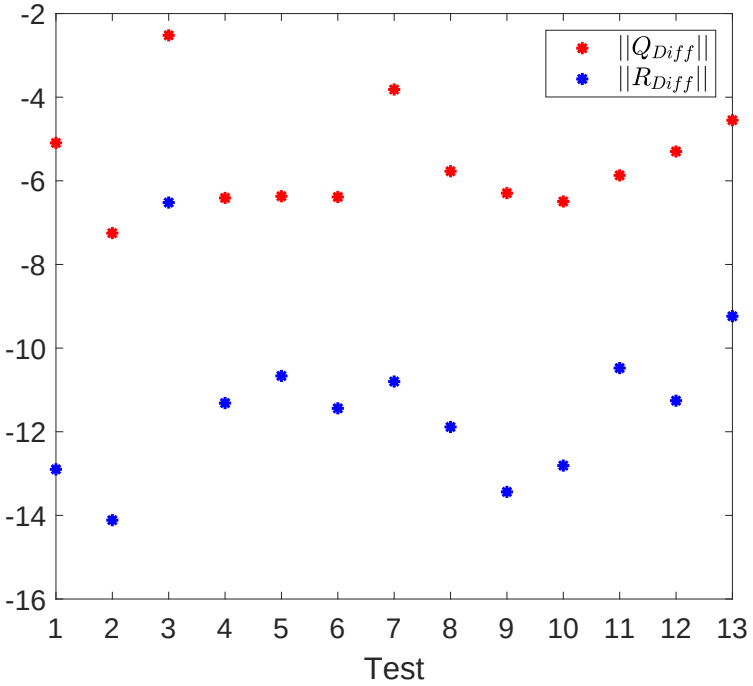


Figure 16: Recovered final error of the Q and R matrices between the learner and pilot for all 13 tests.

CHAPTER IV

CONCLUSION

We develop a novel IRL framework for the estimation of a cost functional, in IRL problems with multiple solutions, through a modification to the HSO [33]. This modification, while simple, requires an exhaustive and rigorous proof to demonstrate convergence to an equivalent solution when multiple solutions are present. We further show that the NHSO is a complete extension of the HSO through convergence to a unique solution when the system is not separable. The Monte-Carlo simulations demonstrate the utility of the NHSO when using a Kalman gain for better noise robustness.

The experimental results demonstrate the ability of the NHSO to consistently learn an equivalent solution of a surrogate LQR pilot's cost function. The estimated cost function reproduces the surrogate pilot's feedback matrix. The robustness of the algorithm 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 as an optimal controller that has full state information and that the pilot transmits velocity commands to the quadcopter. The results of this thesis 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.

Additional assumptions that will need addressed are as follows. The assumption that excitation signals can be designed that do not interrupt a human pilot from performing their mission. Also the assumption that a human behaves according to some deterministic model in addition to assuming said human operates with respect to some LQR basis functions.

Future work will involve experimentation involving human pilots and attempting to replicate their performance through learning equivalent cost functionals. Future work will also involve possible extensions of the developed framework for probabilistic models of pilot behavior.

REFERENCES

[1] Pieter Abbeel, Adam Coates, and Andrew Ng, *Autonomous helicopter aerobatics through apprenticeship learning*, Int. J. Robot. Res. **29** (2010), no. 13, 1608–1639.

[2] Pieter Abbeel and Andrew Y. Ng, *Apprenticeship learning via inverse reinforcement learning*, Proc. Int. Conf. Mach. Learn., 2004.

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

[4] Saurabh Arora, Prashant Doshi, and Bikramjit Banerjee, *Online inverse reinforcement learning under occlusion*, Proc. Conf. Auton. Agents MultiAgent Syst., International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1170–1178.

[5] Samir Bouabdallah, Andre Noth, and Roland Siegwart, *PID vs LQ control techniques applied to an indoor micro quadrotor*, Proc. Intell. Robot. Syst., vol. 3, IEEE, 2004, pp. 2451–2456.

[6] Samir Bouabdallah and Roland Siegwart, *Full control of a quadrotor*, Proc. Intell. Robot. Syst., 2007, pp. 153–158.

[7] Vrushabh S. Donge, Bosen Lian, Frank L. Lewis, and Ali Davoudi, *Multi-agent graphical games with inverse reinforcement learning*, IEEE Trans. Control Netw. Syst. (2022), 1–12.

[8] Michael Herman, Volker Fischer, Tobias Gindele, and Wolfram Burgard, *Inverse reinforcement learning of behavioral models for online-adapting navigation strategies*, Proc. IEEE Int. Conf. Robot. Autom., 2015, pp. 3215–3222.

[9] João P. Hespanha, *Linear systems theory*, Princeton University Press, 2009.

[10] Arthur E. Hoerl and Robert W. Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics **12** (1970), no. 1, 55–67.

[11] Jairo Inga, Esther Bischoff, Timothy Molloy, Michael Flad, and Soren Hohmann, *Solution sets for inverse non-cooperative linear-quadratic differential games*, IEEE Control Syst. Lett. **3** (2019), no. 4, 871–876.

[12] Jairo Inga, Andreas Creutz, and Sören Hohmann, *Online inverse linear-quadratic differential games applied to human behavior identification in shared control*, Proc. Eur. Control Conf., 2021, pp. 323–360.

[13] Maidul Islam, Mohamed Okasha, and Moumen Mohammad Idres, *Trajectory tracking in quadrotor platform by using PD controller and LQR control approach*, IOP Conf. Mater. Sci. Eng., vol. 260, 2017, pp. 2451–2456.

[14] Antony Jameson and Eliezer Kreindler, *Inverse problem of linear optimal control*, SIAM J. Control **11** (1973), no. 1, 1–19.

[15] Frédéric Jean and Sofya Maslovskaya, *Inverse optimal control problem: the linear-quadratic case*, Proc. IEEE Conf. Decis. Control, 2018, pp. 888–893.

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

[17] Hassan K. Khalil, *Nonlinear systems*, third ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[18] lonza leggiera, *Quadratic form vanishing at certain points*, `https://math.stackexchange.com/questions/3230018/quadratic-form-vanishing-at-certain-points?rq=1`, Accessed: 2022-10-18.

[19] Sergey Levine, Zoran Popovic, and Vladlen Koltun, *Feature construction for inverse reinforcement learning*, 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.

[20] ———, *Nonlinear inverse reinforcement learning with Gaussian processes*, 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.

[21] Bosen Lian, Vrushabh S Donge, Frank L Lewis, Tianyou Chai, and Ali Davoudi, *Data-driven inverse reinforcement learning control for linear multiplayer games*, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[22] Bosen Lian, Wenqian Xue, Frank L. Lewis, and Tianyou Chai, *Online inverse reinforcement learning for nonlinear systems with adversarial attacks*, Int. J. Robust Nonlinear Control **31** (2021), no. 14, 6646–6667.

[23] Katja Mombaur, Anh Truong, and Jean-Paul Laumond, *From human to humanoid locomotion—an inverse optimal control approach*, Auton. Robot. **28** (2010), no. 3, 369–383.

[24] Gergely Neu and Csaba Szepesvari, *Apprenticeship learning using inverse reinforcement learning and gradient methods*, Proc. Anu. Conf. Uncertain. Artif. Intell. (Corvallis, Oregon), AUAI Press, 2007, pp. 295–302.

[25] Andrew Y. Ng and Stuart Russell, *Algorithms for inverse reinforcement learning*, Proc. Int. Conf. Mach. Learn., Morgan Kaufmann, 2000, pp. 663–670.

[26] Anil Phatak, Howard Weinert, Ilana Segall, and Carroll N. Day, *Identification of a modified optimal control model for the human operator*, Automatica **12** (1976), no. 1, 31–41.

[27] Sreeraman Rajan, Sichun Wang, Robert Inkol, and Alain Joyal, *Efficient approximations for the arctangent function*, IEEE Signal Process. Mag. **23** (2006), no. 3, 108–111.

[28] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich, *Maximum margin planning*, Proc. Int. Conf. Mach. Learn., 2006.

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

[30] Stuart Russell, *Learning agents for uncertain environments (extended abstract)*, Proc. Conf. Comput. Learn. Theory, 1998.

[31] Ryan V. Self, *On model-based online inverse reinforcement learning*, Ph.D. thesis, Oklahoma State University, 2020.

[32] Ryan V. Self, Moad Abudia, S M Nahid Mahmud, and Rushikesh Kamalapurkar, *Model-based inverse reinforcement learning for deterministic systems*, Automatica **140** (2022), no. 110242, 1–13.

[33] Ryan V. Self, Kevin Coleman, He Bai, and Rushikesh Kamalapurkar, *Online observer-based inverse reinforcement learning*, IEEE Control Syst. Lett. **5** (2021), no. 6, 1922–1927.

[34] Umar Syed and Robert E. Schapire, *A game-theoretic approach to apprenticeship learning*, Adv. Neural Inf. Process. Syst. (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), Curran Associates, Inc., 2008, pp. 1449–1456.

[35] Robert Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B Methodol. **58** (1996), 267–288.

[36] Jared Town, Zachary Morrison, and Rushikesh Kamalapurkar, *Nonuniqueness and convergence to equivalent solutions in observer-based inverse reinforcement learning*, 2022.

[37] Shuting Xu, Wenqian Tan, Alexander V. Efremov, Liguo Sun, and Xiangju Qu, *Review of control models for human pilot behavior*, Annual Reviews in Control **44** (2017), 274–291.

[38] Wenqian Xue, Patrik Kolaric, Jialu Fan, Bosen Lian, Tianyou Chai, and Frank L Lewis, *Inverse reinforcement learning in tracking control based on inverse optimal control*, IEEE Trans. Cybern. (2021).

[39] Zhengyuan Zhou, Michael Bloem, and Nicholas Bambos, *Infinite time horizon maximum causal entropy inverse reinforcement learning*, IEEE Trans. Autom. Control **63** (2018), no. 9, 2787–2802.

[40] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey, *Maximum entropy inverse reinforcement learning*, Proc. AAAI Conf. Artif. Intel., 2008, pp. 1433–1438.

VITA

Jared Curtis Town

Candidate for the Degree of

Master of Science

Thesis: NONUNIQUENESS AND EQUIVALENCE IN ONLINE INVERSE REINFORCEMENT LEARNING WITH APPLICATIONS TO PILOT PERFORMANCE MODELING

Major Field: Mechanical and Aerospace Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2023.

Completed the requirements for the Bachelor of Science in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in 2021.