ON MODEL-BASED ONLINE INVERSE

REINFORCEMENT LEARNING

By

RYAN VOYD SELF

Bachelor of Science in Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma
2014

Master of Science in Mechanical and Aerospace
Engineering
Oklahoma State University
Stillwater, Oklahoma
2016

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2020

ON MODEL-BASED ONLINE INVERSE

REINFORCEMENT LEARNING

Dissertation Approved:

Dr. Rushikesh Kamalapurkar

Dissertation Adviser

Dr. He Bai

Dr. Jamey Jacob

Dr. Gary Yen

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude towards my advisor, Dr. Rushikesh Kamalapurkar, for his guidance, patience, support and encouragement, as I would not have completed my degree without him. I would also like to thank my committee members, Dr. He Bai, Dr. Jamey Jacob and Dr. Gary Yen for their valuable insights and advice on my dissertation. Finally, I would like to thank all members of the SCC and CoRAL research groups for fostering a positive and engaging learning environment.

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.

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

Name: RYAN VOYD SELF

Date of Degree: DECEMBER, 2020

Title of Study: ON MODEL-BASED ONLINE INVERSE REINFORCEMENT
LEARNING

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: Based on the premise that the most succinct representation of the behavior of an entity is its reward structure, inverse reinforcement learning aims to recover the reward (or cost) function by observing an agent perform a task and monitoring state and control trajectories of the observed agent [118]. In general, it has been shown that it is easier to show how to perform a task rather than to describe how to perform the task [101]. Autonomous agents can use this same ideology to develop a mathematical representation, called a reward function, which inherently describes the overall task objective. Inverse reinforcement learning (IRL) is a process in which machines learn to perform complex tasks through analyzing state and control trajectories. Most research that has been done on IRL has been offline, which only allows for repetitive tasks and unchanging environments. The development of real-time IRL techniques, by allowing the autonomous agent to update its reward function in time, would help autonomous entities adapt to changes in the environment by correcting previously inaccurate information, and allow for a more dynamic response to unforeseen alterations in task objectives. In this dissertation, data-driven model-based inverse reinforcement learning techniques are developed that facilitate reward function estimation in real-time. The dissertation then builds off that foundation to explore techniques to resolve sub-optimal trajectories, data sparsity, and partial/imperfect measurements, which are inherent challenges to IRL. An application section is then discussed, including a novel pilot behavior modeling approach.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter I

INTRODUCTION

1.1 Motivation

The use of robots in everyday life has long been a sought-after goal for humans, and as a result, over the past few decades, autonomous systems have been utilized to perform increasingly complex tasks. Autonomous agents have significant advantages over humans due to their repeatability, precision, and resistance to fatigue. The use of autonomous systems is also advantageous in a variety of situations that are potentially harmful to humans [148], such as war-zones and toxic areas. However, increased use of autonomy results in increasingly complex theoretical and practical design challenges. In particular, as autonomous systems become more sophisticated, control objectives tend to become increasingly complex and as a result, require advanced control synthesis strategies.

A popular method for synthesis of complex control strategies is Learning from Demonstration (LfD) [135]. LfD is an ideology in which traditional programming techniques are replaced with "programming by demonstration" [21]. Within LfD, a control strategy is discovered by monitoring a demonstration (i.e., a set of state-action sequences) provided by an expert teacher. The learner's goal is to act in a similar manner using the found policy. LfD is a supervised form of learning using examples, as opposed to unsupervised learning techniques that facilitate learning from experience (such as Reinforcement Learning (RL) [141]). LfD has gained significant interest since it has allowed for synthesis of policies for systems with complicated or potentially unknown dynamics, and for situations in which utilization of traditional control techniques is challenging [45]. However, the potential of autonomous systems will not be fully realized unless the systems are able to adapt to changes and systematically update their control objectives in real-time.

One way to increase adaptability is to learn the mathematical representation to describe the agent's intent (i.e., reward/cost function) while executing a task, rather than simply how the agent completed the task (i.e., the policy). If the reward function can be uncovered for a specific task, the policy can then be synthesized to maximize a cumulative reward using established methods such as reinforcement learning. If an agent solely focused on learning how to complete a specific task (i.e., learning a feedback controller), then any alterations in the environment or task objectives render the task previously learned sub-optimal or infeasible. Instead, if the intent driving the execution of the task is learned (i.e., the reward function), then the agent will be better equipped to adjust to unforeseen changes in real-time, including changes to the agent's dynamics, the environment, or the overall task itself. Learning the reward function, rather than the policy, also facilitates task transfer between two heterogeneous autonomous systems as long as they share the same state space and are capable of learning policies to meet given tasks. For a given reward function, optimal behaviors to achieve tasks can be significantly different for different autonomous systems, especially when the system dynamics are nonlinear. For instance, quadrotors and fixed wing aircraft with similar objectives will perform tasks differently due to the holonomic and nonholonomic nature of their kinematics. Therefore, simply analyzing the trajectories of an agent, though the task trajectories could look very different, the underlying reward function may be the same. Based on the hypothesis that the reward function can be used to succinctly encode a task [118], an optimal control framework, where the behavior describing a task is identified with the reward function of an optimal control problem, is utilized in this research.

Inverse reinforcement learning (IRL) [4] solves the problem of estimating a reward function that describes a task by observing an agent acting in an environment. IRL is a way in which machines can learn how to achieve complex tasks without the explicit programming of the task. IRL is motivated by the difficulty in explicitly defining a reward function to encode a given task beforehand. For many problems, weighing the many, possibly infinite, features that define a reward function such that maximization of cumulative reward that would result in the desired behavior is a nearly impossible task. The reward function difficulties only magnifies when considering teams of autonomous systems.

Many applications, such as search and rescue, reconnaissance, and warfare, require cooperative teams of autonomous systems working together to achieve an overall goal. Team cooperation has distinct advantages over single agents. The use of multi-agent teams to achieve the same goal allows for division of labor and the use of simpler, cheaper robots as opposed to one large complex robot [30, 49]. Though using a team of autonomous systems does have its advantages, the larger the team gets, the more complicated the control structure and communication networks become. If autonomous agents can estimate the reward functions of other agents, they may be able to properly adapt their actions to support the perceived motivation behind the observed behavior, even without direct communication.

The overall goal of this research is to develop novel inverse reinforcement learning (IRL) techniques which facilitate reward function estimation in real-time. To achieve this goal, a cognitive entity will be modeled as an optimal decision maker and the reward/cost function that drives the optimization will be interpreted as the perceived intent of the entity. Estimation of this reward function in real time will allow for seamless execution of an agent in real time by facilitating adaptation of the autonomous agents to updated task objectives and robustness to uncertainties in the environment.

1.2 Review of Literature

1.2.1 Learning from Demonstration

Learning from Demonstration [135] (LfD) has become a popular topic of research with a variety of techniques existing in the literature [122]. However, many of the methods can be categorized into two distinct approaches. The first, known as apprenticeship learning [3], imitation learning [10,12,130,136] or behavioral cloning [133], is primarily focused on learning a policy, or a mapping of states and actions, from the demonstrations or mimicking the demonstrations of the expert demonstrator. While the second one, more commonly referred to as Inverse Reinforcement Learning, aims to recover the true reward function that describes observed demonstrations.

Apprenticeship learning [1,3] uses trajectories with the goal of achieving a policy to behave in a similar manner as an expert demonstrator. These methods are good for situations in which the robot will be performing in a similar environment or if the robot is working in repetitive situations. Many of the techniques in the field have greatly helped in complicated situations in which classical control techniques have been challenging, and there have been numerous methods developed that allow machines to perform complicated tasks [2,5,83]. However, methods in this field are generally not transferable between robots with different dynamics, and are not aimed at uncovering true reward functions. The goal of methods that fall in this category is to uncover any reward function that can be utilized to achieve the desired task.

Inverse Reinforcement Learning [132], sometimes called inverse optimal control [65], is based on the premise that the most succinct representation of the behavior of an entity is its reward structure [118]. IRL aims to recover the reward (or cost) function by observing an agent performing a task and monitoring state and control trajectories of the observed agent. One of the first, and potentially most obvious, advantages of IRL is that these methods can facilitate implementation of reinforcement learning without reward shaping [88, 89, 117]. Reward shaping is the process of designing a reward function so that control policies generated via reinforcement learning methods aimed at maximizing cumulative reward will result in the desired behavior. For some simple problems, such as most video games [144], the reward function is the overall score. However, for more complex problems, defining a reward function may be very difficult. For example, formulating a reward function for driving a car would be challenging given the nearly infinite feature space. It would be easy to assign large negative rewards for features such as driving off the road or running into another car, but assigning rewards for less obvious features becomes increasingly difficult, such as how often to change lanes, how fast to accelerate, how to interpret the actions of another driver at a crossroad, what is the safe distance given between your car and the car in front of you. Taking into consideration all the characteristics that would define the proper reward function for optimally driving a car is a nearly impossible task to achieve. One possible solution to address the aforementioned challenge is to allow the machine to observe the task through set of demonstrations and formulate the reward function using IRL, instead of having to initially express the reward function mathematically to achieve some goal beforehand.

Another advantage of IRL is that the reward function is model agnostic and transferable from one agent to another. The reward function encodes the task, and in general, is independent of the dynamic model of the learner or the demonstrator. As long as the learner and the demonstrator share the same state space, the demonstrator's policy can be readily transferred to the learner using (forward) reinforcement learning to maximize the cumulative reward.

IRL certainly has some challenges. The first is the reward function ambiguity. In the field of inverse reinforcement learning, there exist three types of reward function ambiguities: 1) inherent, where the optimal control problem itself can exhibit similar behaviors under different linearly independent reward functions, 2) scaling, where a reward function and a constant multiple of that reward function result in the exact same policy, and 3) data scarcity, where the availability of a relatively small number of trajectories results in multiple reward functions that can explain the observed behavior. Therefore, when the goal is to uncover an unknown reward function by observing a task, an infinite number of reward functions can typically be found. To address the aforementioned ambiguities, a maximum margin planning technique was developed in [11, 126] where solutions which resulted in a large margin from the demonstration are eliminated through a loss function.

Another challenge that has faced the field of IRL is the requirement for optimal demonstrations. In [159,160], Ziebert et al. presented a Maximum Entropy (MaxEnt) IRL method to help relax the requirement that the demonstrations provided by the expert had to be optimal. The benefit of the MaxEnt IRL approach is that reward function estimation can be achieved in the presence of suboptimal trajectories. This was a useful development since multiple optimal demonstrations are difficult to reproduce. In 2011, Boularias et al. [26] further developed on the idea of Maximum Entropy by using sub-optimal demonstrations for situations in which accurate model knowledge is unavailable. The authors discuss how inaccurate models can lead to poor reward function estimation, and their model-free IRL approach alleviates the difficulty in finding the unknown reward function for such situations. In [157], the authors developed a Maximum Casual Entropy IRL technique for infinite time horizon problems where a stationary soft Bellman policy which helps enable the learning of the transition models is utilized.

Considering multi-agent situations and inter-agent coordinations only magnifies the aforementioned challenges. Multi-agent inverse reinforcement learning was introduced in [114] where they aimed to recover the individual reward function for each agent and used an average-reward based approach to calculate the overall centralized reward function. Bogert and Doshi [22, 23] further developed multi-agent IRL methods by incorporating ideas from the MaxEnt IRL approach. In [140], Šošić et al. extended the concept of multi-agent IRL to systems of swarms. Swarm systems have an inherently unique and interesting challenge, in that the number of agents is so significant that the system cannot be analyzed globally, only locally. Šošić et al. exploited the homogeneity of the swarm problem and were able to formulate local IRL problems for each agent which allow for the use of previously developed single agent IRL approaches.

Generally, inverse reinforcement learning methods involve estimating the reward function as a linear combination of features. However, [93] and [50] extended the IRL ideas to nonlinear formulations, such as Gaussian Processes and multilayer neural networks. In [28], Brown and Niekum took a different approach to the overall IRL problem. While most work on IRL has been focused on uncovering the single reward function from the demonstration, Brown and Niekum investigate machine teaching methods [158] and focused on determining the minimal number of demonstrations required for single reward function estimation.

In [113], Muelling et al. used a model-free IRL approach to learn reward functions for playing table tennis which facilitated reward function estimation without having to develop an accurate model of the human body. Beyond this, many extensions of IRL have been developed, including formulation of feature construction [92], solving IRL using gradient based methods [116], Bayesian Inverse Reinforcement Learning [35–37, 107, 124] designed to define a probability distribution over of reward functions, and game theoretic approaches, as in [143], which suggest the possibility of finding solutions that outperformed the expert.

Techniques such as inverse reinforcement learning, inverse optimal control [65, 111, 160] and apprenticeship learning [3, 116, 142], have been used to teach autonomous agents to perform specific tasks in an *offline* setting. However, *offline* approaches to IRL cannot handle unforeseen changes in task objectives and are ill-suited for adaptation in real-time. The development of real-time IRL techniques is motivated by the need for robustness to uncertainties in the system model and responsiveness to adapt to changing reward structures.

Inspired by real-time reinforcement learning techniques [19, 74, 108, 149, 151], inverse reinforcement learning techniques which facilitate reward function estimation in real time have recently started gaining attention [8, 9, 64, 68, 94, 106, 110, 127–129, 137, 138]. In [106, 127–129], the authors formulate the online IRL problem as a lifelong learning problem, [8,9] extends the Maximum Entropy IRL method to situations in real-time, [64] shows convergence guarantees for real-time IRL, and [68] utilizes a batch IRL method method for linear infinite horizon optimal control problems, while a recursive technique for linear systems is proposed in [110]. While IRL techniques for real-time reward function estimation have started gaining some attention recently, further research is needed to facilitate reward function convergence utilizing single demonstrations under non-ideal situations.

In summary, wide-spread use of IRL for real-time behavior monitoring and control synthesis is hampered by five key challenges: (a) sparsity of available data, (b) nonuniqueness of solutions, (c) partial measurements, (d) imperfect/noisy measurements, and (e) inconsistent observations. This dissertation aims to partially address the above challenges by developing novel real-time IRL methods to be utilized for real-time behavior monitoring and control synthesis. Since the techniques developed in this dissertation are model-based, Chapter III develops a state and parameter estimation technique to help estimate unknowns of the system in real-time. Chapter IV develops a data-driven model-based inverse reinforcement learning technique that is less data intensive than its model-free counterparts, which help facilitate reward function estimation in real-time. Chapter V addresses IRL for scenarios where the observed trajectories of an agent under observation are inconsistent with its internal reward function. Chapter VI attempts to further address the issue of sparsity of available data by formulating a method to artificially create additional data to help drive reward function estimation if trajectories are not sufficiently information rich. Chapter VII formulates the IRL problem in an observer framework to solve the IRL problem in the presence of noisy or imperfect measurements. Chapter VIII discusses applications relevant to the methods developed in this dissertation and presents preliminary results. Chapter IX details the future work section, and concludes the dissertation.

Chapter II

NOTATION

The notation $\mathbb{R}^n$ represents the $n-$dimensional Euclidean space, and the elements of $\mathbb{R}^n$ are interpreted as column vectors, where $(\cdot)^T$ denotes the vector transpose operator. The set of positive integers excluding 0 is denoted by $\mathbb{N}$. For $a \in \mathbb{R}$, $\mathbb{R}_{\geq a}$ denotes the interval $[a, \infty)$, and $\mathbb{R}_{>a}$ denotes the interval (a, ∞). Unless otherwise specified, an interval is assumed to be right-open. If $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$, then $[a; b]$ denotes the concatenated vector $\begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^{m+n}$. The notations I_n and 0_n denote the $n \times n$ identity matrix and the zero element of $\mathbb{R}^n$, respectively. Whenever it is clear from the context, the subscript n is suppressed. $\chi|_{\mathbb{R}^n}$ denotes the projection of $\chi \subseteq \mathbb{R}^n \times \mathbb{R}^m$ onto $\mathbb{R}^n$. The notation $f \in C^N(X, Y)$ denotes that the function $f : X \to Y$ is N-times continuously differentiable. The notation $f \in O(g)$ denotes that there exists $c, M \in \mathbb{R}_{>0}$ such that $|f(x)| \leq c|g(x)| \ \forall \ x > M$.

Chapter III

STATE AND PARAMETER ESTIMATION

For certain classes of systems (made precise in the following), the data-sparsity challenge in IRL can be effectively addressed by using additional insights gained from the dynamic model of the system. Since system models are uncertain and changeable, an adaptive system identification technique is developed in this chapter to support the development and implementation of model-based IRL methods.

3.1 Introduction

Traditional adaptive control methods handle uncertainty in the system dynamics by maintaining a parametric estimate of the model and utilizing it to generate a feedforward control signal (see, e.g., [60, 85, 134]). While the feedforward-feedback architecture guarantees stability of the closed-loop, the control law is not robust to disturbances, and seldom provides information regarding the quality of the estimated model (cf. [60] and [134]). While accurate parameter estimation can improve robustness and transient performance of adaptive controllers, (see, e.g., [40, 48, 86]), parameter convergence typically requires restrictive assumptions such as persistence of excitation (PE) [7, 51, 52, 60]. An excitation signal is often added to the controller to ensure persistence of excitation; however, the added signal can cause mechanical fatigue and compromise the tracking performance. Therefore, the development of techniques that facilitate parameter convergence without the requirement of PE is motivated.

Parameter convergence can be achieved under a finite excitation condition using data-driven methods, such as concurrent learning (CL) (see, e.g., [40, 44, 79]), where the parameters are estimated by storing data during time-intervals when the system is excited, and then utilizing the stored data to drive adaptation when excitation is unavailable. In addition to parameter estimation, CL adaptive control methods also possess similar robustness to bounded disturbances as $\sigma-$modification, $e-$modification, etc., without the associated drawbacks such as drawing the parameter estimates to arbitrary set-points [40, 43, 44, 79].

Adaptation techniques similar to CL are utilized to implement reinforcement learning under finite excitation conditions in results such as [20, 71, 73, 74, 100, 109]. CL methods have also been extended to classes of switched systems [147, 153], systems driven by stochastic processes [42], and systems with time-varying parameters [90]. A major drawback of CL methods is that they require numerical differentiation of the state measurements. CL methods that do not require numerical differentiation of the state measurements are developed in results such as [72] and [120], however, they require full state feedback. Since full state feedback is often not available, the development of an output-feedback CL framework is well-motivated.

Due to advantageous properties such as the separation principle, there is a large body of literature on simultaneous state and parameter estimation for linear systems [13, 84, 115]. Estimation methods for linear systems typically use popular techniques, such as Kalman filters, because of their well-documented effectiveness. More recently, researchers have also explored the state and parameter estimation problem for nonlinear systems [15–18, 33, 39, 46, 87, 99, 103, 146]. While tools such as particle filters [33], extended Kalman filters [146], multi-observers [39], and adaptive observers [15–18, 99, 103] have been examined for nonlinear simultaneous state and parameter estimation, they either do not provide theoretical performance guarantees [33, 146] or require stringent assumptions [15–18, 39, 99, 103], such as PE, which are generally difficult, if not impossible, to check online. While relaxed PE results are presented in [97, 98, 119], these results still require a persistent excitation condition. Therefore, this chapter aims to provide both theoretical guarantees and finite (as opposed to persistent) excitation assumptions that are verifiable online.

In this chapter, the preliminary results from [66] and [67] are consolidated and generalized to yield an output feedback concurrent learning method for simultaneous state and parameter estimation in uncertain linear and nonlinear systems. In particular, this chapter yields a formal method for simultaneous state and parameter estimation for a broad class of dynamical systems that includes the Brunovsky canonical form studied in [66] and [67] as a special case. An adaptive state-observer is utilized to generate estimates of the state from input-output data. The estimated state trajectories along with the known inputs are then utilized in a novel data-driven parameter estimation scheme to achieve simultaneous state and parameter estimation. Convergence of the state estimates and the parameter estimates to a small neighborhood of the origin is established under a *finite* (as opposed to *persistent*) excitation condition.

This chapter is organized as follows. In Section 3.2, the class of nonlinear systems that the developed method applies to is described. An integral error system that facilitates parameter estimation is developed in Section 3.2.2. Section 3.2.3 is dedicated to the design of a robust state observer. Section 3.2.4 details the developed parameter estimator. Section 3.2.5 details the algorithm for selection and storage of the data that is used to implement concurrent learning. Section 3.2.6 is dedicated to a Lyapunov-based analysis of the developed technique. In Section 3.3, linear systems are considered. A linear error system is developed in Section 3.3.2 to facilitate CL-based adaptation. A CL-based parameter estimator is designed in Section 3.3.3. A Lyapunov-based stability analysis of the parameter estimator is presented in Section 3.3.4. Section 3.4 demonstrates the efficacy of the developed method via a numerical simulation and Section 3.5 concludes the chapter.

3.2 Nonlinear Systems

3.2.1 Problem Formulation

Consider a nonlinear system of the form

$$\begin{aligned}
\dot{x}_1 &= f_1(x^-, u), \\
\dot{x}_2 &= f_2(x^-, u) + x_3, \\
\dot{x}_3 &= f_3\left(x, u\right), \\
y &= x^-,
\end{aligned} \tag{1}$$

where $x_1 \in \mathbb{R}^{n_1}$ and $x_2, x_3 \in \mathbb{R}^{n_2}$ denote the state variables, $x := \begin{bmatrix} x_1^T & x_2^T & x_3^T \end{bmatrix}^T$ is the system state, $f_1 : \mathbb{R}^{n_1+n_2} \times \mathbb{R}^m \to \mathbb{R}^{n_1}$ and $f_2 : \mathbb{R}^{n_1+n_2} \times \mathbb{R}^m \to \mathbb{R}^{n_2}$ are known and locally Lipschitz continuous, $f_3 : \mathbb{R}^{n_1+2n_2} \times \mathbb{R}^m \to \mathbb{R}^{n_2}$ is locally Lipschitz continuous, $u \in \mathbb{R}^m$ is the controller, $y \in \mathbb{R}^{n_1+n_2}$ denotes the output, and $x^- := \begin{bmatrix} x_1^T & x_2^T \end{bmatrix}^T$ denotes the measurable part of the system state. The model, f_3, is comprised of a known nominal part and an unknown part, i.e., $f_3 = f^o + g$, where $f^o : \mathbb{R}^{n_1+2n_2} \times \mathbb{R}^m \to \mathbb{R}^{n_2}$ is known and locally Lipschitz and $g : \mathbb{R}^{n_1+2n_2} \times \mathbb{R}^m \to \mathbb{R}^{n_2}$ is unknown and locally Lipschitz. The objective is to design an adaptive estimator to identify the state, x, and the unknown function, g, online, using input-output measurements.

Systems of the form (1) encompass N^{th}-order linear systems and Euler-Lagrange models with invertible inertia matrices, and hence, represent a wide class of physical plants, including but not limited to robotic manipulators and autonomous ground, aerial, and underwater vehicles.

Assumption 1 *A compact set $\chi \subseteq \mathbb{R}^{n_1+2n_2} \times \mathbb{R}^m$ such that $(x(t), u(t)) \in \chi, \forall\ t \in \mathbb{R}_{\geq T_0}$*[1] *and $\forall\ T_0 \geq 0$ is known, where $T_0 \in \mathbb{R}_{\geq 0}$ denotes the initial time.*

[1] For $a \in \mathbb{R}$, the notation $\mathbb{R}_{\geq a}$ denotes the interval $[a, \infty)$ and the notation $\mathbb{R}_{>a}$ denotes the interval (a, ∞).

Remark 1 *The problem formulation in* (1) *incorporates commonly occurring dynamical systems described using the Brunovsky canonical form [29]*

$$\{\dot{x}_i = x_{i+1}\}_{i=1}^{N-1},\ \dot{x}_N = f(x,u),\ y = x^-, \tag{2}$$

and the extended Brunovsky form

$$\left\{\dot{x}_i = f_i(x^-,u) + x_{i+1}\right\}_{i=1}^{N-1},\ \dot{x}_N = f(x,u),\ y = x^-, \tag{3}$$

where $x_1, x_2, \ldots, x_N \in \mathbb{R}^n$ *denote the state variables,* $x := \begin{bmatrix} x_1^T & x_2^T & \ldots & x_N^T \end{bmatrix}^T$ *is the system state,* $f_1, f_2, \ldots, f_{N-1} : \mathbb{R}^{(N-1)n} \times \mathbb{R}^m \to \mathbb{R}^n$ *and* $f : \mathbb{R}^{Nn} \times \mathbb{R}^m \to \mathbb{R}^n$ *are locally Lipschitz continuous,* $u \in \mathbb{R}^m$ *is the controller,* $y \in \mathbb{R}^{(N-1)n}$ *denotes the output, and* $x^- := \begin{bmatrix} x_1^T & x_2^T & \ldots & x_{N-1}^T \end{bmatrix}^T$ *denotes the measureable part of the system state.*

3.2.2 Error System for Estimation

Given a constant $\overline{\epsilon} \in \mathbb{R}_{>0}$, there exist $p \in \mathbb{N}$ and $\overline{\sigma}, \overline{\theta} \in \mathbb{R}_{>0}$, such that the unknown function g can be approximated, over the compact set χ, using basis functions $\sigma : \mathbb{R}^{n_1+2n_2} \times \mathbb{R}^m \to \mathbb{R}^p$ as $g\left(x,u\right) = \theta^T \sigma\left(x,u\right) + \epsilon\left(x,u\right)$, where $\epsilon : \mathbb{R}^{n_1+2n_2} \times \mathbb{R}^m \to \mathbb{R}^{n_2}$ denotes the approximation error, $\theta \in \mathbb{R}^{p \times n_2}$ is a constant matrix of unknown parameters, and $\max_{(x,u)\in\chi} \left\|\sigma\left(x,u\right)\right\| < \overline{\sigma}$, $\max_{(x,u)\in\chi} \left\|\nabla\sigma\left(x,u\right)\right\| < \overline{\sigma}$, $\max_{(x,u)\in\chi} \left\|\epsilon\left(x,u\right)\right\| < \overline{\epsilon}$, $\max_{(x,u)\in\chi} \left\|\nabla\epsilon\left(x,u\right)\right\| < \overline{\epsilon}$, and $\|\theta\| < \overline{\theta}$ [58,59]. To obtain an error signal for parameter identification, the system in (1) is expressed in the form

$$\dot{x}_3 = f^o\left(x,u\right) + \theta^T \sigma\left(x,u\right) + \epsilon\left(x,u\right). \tag{4}$$

Integrating (4) over the interval $[t - \tau_1, t]$ for some constant $\tau_1 \in \mathbb{R}_{>0}$ and then over the interval $[t - \tau_2, t]$ for some constant $\tau_2 \in \mathbb{R}_{>0}$,

$$\begin{aligned} \int_{t-\tau_2}^{t} \big(x_3\left(\zeta_2\right) - x_3\left(\zeta_2 - \tau_1\right)\big)\, \mathrm{d}\zeta_2 &= \left[\mathcal{I}_2 f^o\right](t) + \theta^T \left[\mathcal{I}_2 \sigma\right](t) \\ &\quad + \left[\mathcal{I}_2 \epsilon\right](t), \forall t \in \mathbb{R}_{\geq \mathscr{T}}, \text{where } \mathscr{T} = T_0 + \tau_1 + \tau_2, \end{aligned} \tag{5}$$

and $\mathcal{I}_2$ denotes the double integral operator

$$f \mapsto \int_{t-\tau_2}^{t} \int_{\zeta_2-\tau_1}^{\zeta_2} f\left(x\left(\zeta_1\right), u\left(\zeta_1\right)\right) \mathrm{d}\zeta_1 \, \mathrm{d}\zeta_2.$$

Using the Fundamental Theorem of Calculus and the fact that $x_3(t) = \dot{x}_2(t) - f_2(x^-(t), u(t))$, for almost all $t \in \mathbb{R}_{\geq T_0}$,

$$x_2(t-\tau_2-\tau_1) - x_2(t-\tau_1) - x_2(t-\tau_2) + x_2(t) = \left[\mathcal{I}_1 f_2\right](t) - \left[\mathcal{I}_1 f_2\right](t-\tau_1) + \left[\mathcal{I}_2 f^o\right](t) + \theta^T \left[\mathcal{I}_2 \sigma\right](t) + \left[\mathcal{I}_2 \epsilon\right](t), \forall t \in \mathbb{R}_{\geq \mathscr{T}}, \quad (6)$$

and $\mathcal{I}_1$ denotes the single integral operator

$$f \mapsto \int_{t-\tau_2}^{t} f\left(x_1\left(\zeta_2\right), u\left(\zeta_2\right)\right) \mathrm{d}\zeta_2.$$

The expression in (7) can be rearranged to form the affine system

$$X(t) = F(t) + \theta^T G(t) + E(t), \ \forall t \in \mathbb{R}_{\geq T_0}, \quad (7)$$

where[2]

$$X(t) := \begin{cases} x_2(t-\tau_2-\tau_1) - x_2(t-\tau_1) - x_2(t-\tau_2) + x_2(t), & t \in [\mathscr{T}, \infty), \\ 0, & t < \mathscr{T}, \end{cases} \quad (8)$$

$$F(t) := \begin{cases} \left[\mathcal{I}_1 f_2\right](t) - \left[\mathcal{I}_1 f_2\right](t-\tau_1) + \left[\mathcal{I}_2 f^o\right](t), & t \in [\mathscr{T}, \infty), \\ 0, & t < \mathscr{T}, \end{cases} \quad (9)$$

$$G(t) := \begin{cases} \left[\mathcal{I}_2 \sigma\right](t), & t \in [\mathscr{T}, \infty), \\ 0, & t < \mathscr{T}, \end{cases} \quad (10)$$

[2] The matrices X, F, G, and E are evaluated along the trajectories of (1), and as such, are functions of T_0, $x(\cdot)$ and $u(\cdot)$. Since the bound on $x(\cdot)$ and $u(\cdot)$ imposed by Assumption 1 is uniform in T_0, the dependence of X, F, G, and E on T_0 is not relevant to the subsequent analysis, and as such, is not made explicit in the notation.

$$E\left(t\right):=\begin{cases}\left[\mathcal{I}_{2}\epsilon\right]\left(t\right), & t\in\left[\mathscr{T},\infty\right),\\ 0, & t<\mathscr{T}.\end{cases}\tag{11}$$

The affine relationship in (7) is valid for all $t\in\mathbb{R}_{\geq T_0}$; however, it provides useful information about the vector θ only after $t\geq\mathscr{T}$. In the following, (7) will be used to solve the simultaneous state and parameter estimation problem.

While (7) can be used to learn the unknown parameters, θ, knowledge of the state variable x_3 is required to compute the matrices F and G. A robust adaptive state estimator is developed in the following to generate estimates of x_3.

3.2.3 State Estimator Design

To generate estimates of x_3, a state estimator inspired by [47] is developed. The estimator is given by

$$\begin{aligned}\dot{\hat{x}}_1&=f_1(\hat{x}^-,u)\\ \dot{\hat{x}}_2&=f_2(\hat{x}^-,u)+\hat{x}_3\\ \dot{\hat{x}}_3&=f^o\left(\hat{x},u\right)+\hat{\theta}^T\sigma\left(\hat{x},u\right)+\nu,\end{aligned}\tag{12}$$

where $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$, $\hat{x}$, $\hat{x}^-$, and $\hat{\theta}$ are estimates of x_1, x_2, x_3, x, x^-, and θ, respectively, and ν is a feedback term designed in the following.

To facilitate the design of ν, let the state and parameter estimation errors be defined as

$$\tilde{x}=x-\hat{x},\quad\tilde{\theta}=\theta-\hat{\theta},\tag{13}$$

and define the model error as

$$\dot{\tilde{x}}_1=\tilde{f}_1(x^-,u,\hat{x}^-)-\alpha\tilde{x}_1,\tag{14}$$

where α is a positive constant, and $\tilde{f}_1\left(x^-,u,\hat{x}^-\right):=f_1\left(x^-,u\right)-f_1\left(\hat{x}^-,u\right)$. The feedback component ν is designed as

$$\nu=\alpha^2\tilde{x}_2-\left(k+\alpha+\beta\right)\eta,\tag{15}$$

where the signal η is added to compensate for the fact that the state variable x_3 is not measurable. Based on the subsequent stability analysis, the signal η is designed as the output of the dynamic filter

$$\dot{\zeta} = -(\beta + k)\,\eta - k\alpha\tilde{x}_2 + (k + \alpha)\,\tilde{f}_2(x^-, u, \hat{x}),$$
$$\eta = \zeta - (k + \alpha)\left(\tilde{x}_2 - \tilde{x}_2(T_0)\right), \zeta(T_0) = 0, \tag{16}$$

where k and β are positive constants and the error signal r is defined as

$$r := \dot{\tilde{x}}_2 + \alpha\tilde{x}_2 - \tilde{f}_2(x^-, u, \hat{x}^-) + \eta, \tag{17}$$

and $\tilde{f}_2(x^-, u, \hat{x}^-) := f_2(x^-, u) - f_2(\hat{x}^-, u)$.

Using integration by parts to eliminate the auxiliary variable ζ, the dynamic filter can be expressed in the equivalent form

$$\dot{\eta} = -\beta\eta - kr - \alpha\tilde{x}_3, \quad \eta\,(T_0) = 0. \tag{18}$$

In the following, the filter in (16) is used for implementation and the filter in (18), which is not implementable due to its dependence on $\tilde{x}_3$, is used for analysis.

Since $\left(x^-, u\right) \mapsto f_1\left(x^-, u\right)$ and $\left(x^-, u\right) \mapsto f_2\left(x^-, u\right)$ are locally Lipschitz, given a compact set $\tilde{\chi} \subset \chi \times \mathbb{R}^{n_1+2n_2}$, Assumption 1 can be used to conclude that there exists an $L > 0$, independent of T_0, such that

$$\max_{\left(x^-, u, \tilde{x}^-\right) \in \tilde{\chi}} \left\|\tilde{f}_1\left(x^-, u, \hat{x}^-\right)\right\| \leq L \left\|\tilde{x}^-\right\|,$$

and

$$\max_{\left(x^-, u, \tilde{x}^-\right) \in \tilde{\chi}} \left\|\tilde{f}_2\left(x^-, u, \hat{x}^-\right)\right\| \leq L \left\|\tilde{x}^-\right\|. \tag{19}$$

To generate the estimates $\hat{\theta}$, a *concurrent learning* [41] technique that utilizes only the output measurements is developed, motivated by the affine error system in (7).

3.2.4 Parameter Estimator Design

To obtain an output-feedback concurrent learning update law for the parameter estimates, a history stack, denoted by $\mathcal{H}$, is utilized. A history stack is defined as a set of ordered pairs $\left\{\left(X_i, \hat{F}_i, \hat{G}_i\right)\right\}_{i=1}^{M}$ such that

$$X_i = \hat{F}_i + \theta^T \hat{G}_i + \mathcal{E}_i, \; \forall i \in \{1, \cdots, M\}, \tag{20}$$

where $\mathcal{E}_i$ is a matrix with an induced 2-norm that is small enough in a sense that is made precise in the subsequent analysis. Typically, a history stack that satisfies (20) is not available a priori. The history stack is recorded online using the relationship in (7), by selecting an increasing set of time-instances $\{t_i\}_{i=1}^{M}$ (see Fig. 1) and letting

$$X_i = X\left(t_i\right), \quad \hat{F}_i = \hat{F}\left(t_i\right), \quad \hat{G}_i = \hat{G}\left(t_i\right), \tag{21}$$

where[3]

$$\hat{F}\left(t\right) := \begin{cases} \left[\hat{\mathcal{I}}_1 f_2\right]\left(t\right) - \left[\hat{\mathcal{I}}_1 f_2\right]\left(t - \tau_1\right) + \left[\hat{\mathcal{I}}_2 f^o\right]\left(t\right), & t \in \left[\mathscr{T}, \infty\right), \\ 0, & t < \mathscr{T}, \end{cases} \tag{22}$$

$$\hat{G}\left(t\right) := \begin{cases} \left[\hat{\mathcal{I}}_2 \sigma\right]\left(t\right), & t \in \left[\mathscr{T}, \infty\right), \\ 0, & t < \mathscr{T}, \end{cases} \tag{23}$$

where $\hat{\mathcal{I}}_2$ denotes the double integral operator

$$f \mapsto \int_{t-\tau_2}^{t} \int_{\zeta_2 - \tau_1}^{\zeta_2} f\left(\hat{x}\left(\zeta_1\right), u\left(\zeta_1\right)\right) \mathrm{d}\zeta_1 \, \mathrm{d}\zeta_2,$$

and $\hat{\mathcal{I}}_1$ denotes the single integral operator

$$f \mapsto \int_{t-\tau_2}^{t} f\left(\hat{x}^{-}\left(\zeta_2\right), u\left(\zeta_2\right)\right) \mathrm{d}\zeta_2.$$

[3] The matrices $\hat{F}$ and $\hat{G}$ are evaluated along the trajectories of (12), (18), (25), and (26), and as such, depend on T_0, $u(\cdot)$, $x(\cdot)$, $\hat{x}(T_0)$, and $\hat{\theta}(T_0)$. For brevity of notation, the matrices are denoted as functions of time.

In this case, the error term $\mathcal{E}_i$ is given by $\mathcal{E}_i = E(t_i)+F(t_i)-\hat{F}(t_i)+\theta^T\Big(G\left(t_i\right)-\hat{G}\left(t_i\right)\Big)$. Let $[t_1, t_2) \subseteq \mathbb{R}_{\geq \mathscr{T}}$ be an interval over which the history stack was recorded. Provided the states and the state estimation errors remain within the compact sets $\chi|_x$ and $\tilde{\chi}|_{\tilde{x}}$, respectively,[4] over $I := [t_1 - \tau_1 - \tau_2, t_2)$, the error terms can be bounded as

$$\|\mathcal{E}_i\| \leq L_1\overline{\epsilon} + L_2\overline{\tilde{x}}_I, \forall i \in \{1, \cdots, M\}, \tag{24}$$

where $\overline{\tilde{x}}_I := \max_{i\in\{1,\cdots,M\}} \sup_{t\in I} \left\|\tilde{x}\left(t\right)\right\|$ and $L_1, L_2 > 0$ are constants.

The concurrent learning update law to estimate the unknown parameters is designed as

$$\dot{\hat{\theta}} = k_\theta \Gamma \sum_{i=1}^{M} \frac{\hat{G}_i}{1+\kappa\|\hat{G}_i\|^2}\left(X_i - \hat{F}_i - \hat{\theta}^T\hat{G}_i\right)^T, \tag{25}$$

where $k_\theta \in \mathbb{R}_{>0}$ is a constant adaptation gain and $\Gamma \in \mathbb{R}^{p\times p}$ is the least-squares gain updated using the update law

$$\dot{\Gamma} = \beta_1\Gamma - k_\theta\Gamma\mathscr{G}\Gamma, \tag{26}$$

where the matrix $\mathscr{G} \in \mathbb{R}^{p\times p}$ is defined as $\mathscr{G} := \sum_{i=1}^{M}\left(\frac{\hat{G}_i}{\sqrt{1+\kappa\|\hat{G}_i\|^2}}\right)\left(\frac{\hat{G}_i}{\sqrt{1+\kappa\|\hat{G}_i\|^2}}\right)^T$ and $\kappa, \beta_1 \in \mathbb{R}_{>0}$.

3.2.5 Purging

The update law in (25) is motivated by the fact that if the full state were available for feedback and if the approximation error, ϵ, were zero, then using

$$\begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}^T = \begin{bmatrix} F_1 & \cdots & F_n \end{bmatrix}^T + \begin{bmatrix} G_1 & \cdots & G_n \end{bmatrix}^T \theta, \tag{27}$$

the parameters could be estimated via the least squares estimate

$$\begin{aligned}\hat{\theta}_{\mathrm{LS}} = \mathscr{G}^{-1}\begin{bmatrix} G_1 & \cdots & G_n \end{bmatrix}\begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}^T \\ - \mathscr{G}^{-1}\begin{bmatrix} G_1 & \cdots & G_n \end{bmatrix}\begin{bmatrix} F_1 & \cdots & F_n \end{bmatrix}^T.\end{aligned} \tag{28}$$

[4] $\chi|_x := \{x \in \mathbb{R}^{n_1+2n_2} | (x,u) \in \chi\}$ and $\tilde{\chi}|_{\tilde{x}} := \{\tilde{x} \in \mathbb{R}^{n_1+2n_2} | (x,u,\tilde{x}) \in \tilde{\chi}\}$.

However, since the history stack contains the estimated terms $\hat{F}$ and $\hat{G}$, during the transient period where the state estimation error is large, the history stack does not accurately (within the error bound introduced by ϵ) represent the system dynamics. Hence, the history stack needs to be purged whenever *better* estimates of the state are available.

Since the state estimator *exponentially* drives the estimation error to a small neighborhood of the origin, a newer estimate of the state can be assumed to be at least as good as an older estimate, apart from the small error introduced by practical stability of the estimator. This fact motivates the dwell time based greedy purging algorithm developed in the following to utilize newer data for estimation while preserving stability of the estimator.

The algorithm maintains two history stacks, a main history stack and a transient history stack, labeled $\mathcal{H}$ and $\mathcal{G}$, respectively. As soon as the transient history stack is full and sufficient dwell time has passed, the main history stack is emptied and the transient history stack is copied into the main history stack. A lower bound on the required dwell time, denoted by $\mathcal{T}$, is determined in Section 3.2.6 using a Lyapunov-based stability analysis.

Parameter identification in the developed framework requires a full rank history stack $\mathcal{H}$, which is achieved provided the trajectories contain sufficient information, as quantified by the following assumption.

Assumption 2 *There exist $\underline{c}, T > 0$ such that for all $T_0 \in \mathbb{R}_{\geq 0}$, $\hat{x}(T_0) \in \tilde{\chi}|_{\hat{x}}$, $\hat{\theta}(T_0) \in \mathbb{R}^p$, and system trajectories $x : \mathbb{R}_{\geq T_0} \to \chi|_x$ in response to the controllers $u : \mathbb{R}_{\geq T_0} \to \chi|_u$, there exist $M \in \mathbb{N}$ and time instances $T_0 \leq t_1 < t_2 < \ldots < t_M \leq T$, such that a history stack recorded using Fig. 1 satisfies*

$$\underline{c} < \lambda_{\min}\left\{\mathscr{G}(t)\right\}, \forall t \in \mathbb{R}_{\geq T_0}, \tag{29}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum singular value of a matrix.

Remark 2 *Uniformity of excitation, with respect to initial conditions and the true state and control trajectories, is required for uniform stability of the estimator (cf. [119]). If uniformity of excitation cannot be guaranteed, then, as long as* (29) *holds for a specific set of initial conditions and state and control trajectories, the estimation error of the developed state and parameter estimator, starting from the given initial conditions and evaluated along the given true state and control trajectories, can be shown to be ultimately bounded using analysis techniques similar to Section 3.2.6.*

Motivated by the observation that the rate of decay of the parameter estimation errors is proportional to the minimum singular value of $\mathscr{G}$, a singular value maximization algorithm is used to select the time instances $\{t_i\}_{i=1}^{M}$. That is, a data-point $\left(X_j, \hat{F}_j, \hat{G}_j\right)$ in the history stack is replaced with a new data-point $\left(X^*, \hat{F}^*, \hat{G}^*\right)$, where $\hat{F}^* = \hat{F}(t)$, $X^* = X(t)$, and $\hat{G}^* = \hat{G}(t)$, for some t, only if

$$\lambda_{\min}\left(\sum_{i\neq j} \mu_i \hat{G}_i \hat{G}_i^T + \mu_j \hat{G}_j \hat{G}_j^T\right) < \frac{\lambda_{\min}\left(\sum_{i\neq j} \mu_i \hat{G}_i \hat{G}_i^T + \mu^* \hat{G}^* \hat{G}^{*T}\right)}{(1+\psi)}, \tag{30}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum singular value of a matrix, ψ is a tunable constant, $\mu_i = \frac{1}{1+\kappa\|G_i\|^2}$, $\mu_j = \frac{1}{1+\kappa\|G_j\|^2}$, and $\mu^* = \frac{1}{1+\kappa\|G^*\|^2}$. To simplify the analysis, it is assumed that new data points are only collected $\tau_1 + \tau_2$ seconds after a purging event. Since the history stack is updated using a singular value maximization algorithm, the matrix $\mathscr{G}$ is a piece-wise constant function of time with the property that once it satisfies (29), at some $t = T$, and for some $\underline{c}$, the condition $\underline{c} < \lambda_{\min}\left(\mathscr{G}(t)\right)$ holds for all $t \geq T$. The developed purging algorithm is summarized in Fig. 1.

A Lyapunov-based analysis showing uniform ultimate boundedness of the parameter and the state estimation errors is presented in the following section.

3.2.6 Analysis

Each purging event represents a discontinuous change in the system dynamics; hence, the resulting closed-loop system is a switched system. To facilitate the analysis of the switched system, let $\rho : \mathbb{R}_{\geq T_0} \to \mathbb{N}$ denote a switching signal such that $\rho(T_0) = 1$, and $\rho(t) = j + 1$, where j denotes the number of times the update $\mathcal{H} \leftarrow \mathcal{G}$ was carried out over the time interval (T_0, t). For a given $s \in \mathbb{N}$, let $\mathcal{H}_s$ denote the history stack active during the time interval $\{t \mid \rho(t) = s\}$, containing the elements $\left\{\left(X_{si}, \hat{F}_{si}, \hat{G}_{si}\right)\right\}_{i=1,\cdots,M}$, and let $\mathcal{E}_{si}^T$ be the corresponding error term. To simplify the notation, let $\mathscr{G}_s := \sum_{i=1}^{M} \frac{\hat{G}_{si}\hat{G}_{si}^T}{1+\kappa\|G_{si}\|^2}$, and $Q_s := \sum_{i=1}^{M} \frac{\hat{G}_{si}\mathcal{E}_{si}^T}{1+\kappa\|G_{si}\|^2}$.

Using (20) and (25), the dynamics of the parameter estimation error can be expressed as

$$\dot{\tilde{\theta}} = -k_\theta \Gamma \mathscr{G}_s \tilde{\theta} - k_\theta \Gamma Q_s. \tag{31}$$

Since the functions $\mathscr{G}_s : \mathbb{R}_{\geq T_0} \to \mathbb{R}^{p\times p}$ and $Q_s : \mathbb{R}_{\geq T_0} \to \mathbb{R}^{p\times n}$ are piece-wise continuous, the trajectories of (31), and of all the subsequent error systems involving $\mathscr{G}_s$ and Q_s, are defined in the sense of Carathéodory [54]. The algorithm in Fig. 1 ensures that there exists a constant $\underline{c} > 0$ such that $\lambda_{\min}\{\mathscr{G}_s\} \geq \underline{c}$, $\forall s \in \mathbb{N}$.

Using the dynamics in (1), (12) - (18), and the design of the feedback component in (15), the evolution of the error signal r is described by

$$\begin{aligned}\dot{r} = -kr + \tilde{f}^o(x,u,\hat{x}) + \theta^T \tilde{\sigma}(x,u,\hat{x}) - \tilde{\theta}^T \tilde{\sigma}(x,u,\hat{x})\\ + \tilde{\theta}^T \sigma(x,u) + \epsilon(x,u) - \alpha^2 \tilde{x}_2 + \alpha \tilde{f}_2(x^-,u,\hat{x}^-) + (k+\alpha)\eta,\end{aligned} \tag{32}$$

where $\tilde{\sigma}(x,u,\hat{x}) = \sigma(x,u) - \sigma(\hat{x},u)$ and $\tilde{f}^o(x,u,\hat{x}) = f(x,u) - f(\hat{x},u)$. Since $(x,u) \mapsto f(x,u)$ and $(x,u) \mapsto \sigma(x,u)$ are locally Lipschitz, given a compact set $\tilde{\chi} \subset \chi \times \mathbb{R}^{n_1+2n_2}$, Assumption 1 can be used to conclude that there exist $L_f, L_\sigma > 0$, independent of T_0, such that

$$\sup_{(x,u,\tilde{x})\in\tilde{\chi}} \left\|\tilde{f}^o(x,u,\hat{x})\right\| \leq L_f \|\tilde{x}\|,$$

and

$$\sup_{(x,u,\tilde{x})\in\tilde{\chi}} \left\|\tilde{\sigma}(x,u,\hat{x})\right\| \leq L_\sigma \|\tilde{x}\|. \tag{33}$$

```
$\delta\left(T_0\right) \leftarrow 0$, $\Omega\left(T_0\right) \leftarrow 0$
if $t > \delta\left(t\right) + \tau_1 + \tau_2$ and a data point is available then
    if $\mathcal{G}$ is not full then
        add the data point to $\mathcal{G}$
    else
        add the data point to $\mathcal{G}$ if (30) holds
    end if
    if $\lambda_{\min}\left(\mathscr{G}\right) \geq \xi\Omega\left(t\right)$ then
        if $t - \delta\left(t\right) \geq \mathcal{T}\left(t\right)$ then
            $\mathcal{H} \leftarrow \mathcal{G}$ and $\mathcal{G} \leftarrow 0$          ▷ purge and replace $\mathcal{H}$
            $\delta\left(t\right) \leftarrow t$
            if $\Omega\left(t\right) < \lambda_{\min}\left(\mathscr{G}\right)$ then
                $\Omega\left(t\right) \leftarrow \lambda_{\min}\left(\mathscr{G}\right)$
            end if
        end if
    end if
end if
```

Figure 1: Algorithm for history stack purging with dwell time. At each time instance t, $\delta\left(t\right)$ stores the last time instance $\mathcal{H}$ was purged, $\Omega\left(t\right)$ stores the highest minimum singular value of $\mathscr{G}$ encountered so far, $\mathcal{T}\left(t\right)$ denotes the dwell time, and $\xi \in (0, 1]$ denotes a threshold fraction.

To facilitate the analysis, let $\left\{T_s \in \mathbb{R}_{\geq 0} \mid s \in \mathbb{N}\right\}$ be a set of switching time instances defined as $T_s = \left\{t \,|\, \rho\left(\tau\right) < s+1, \forall \tau \in [T_0, t) \wedge \rho\left(\tau\right) \geq s+1, \forall \tau \in [t, \infty)\right\}$. That is, for a given switching index s, T_s denotes the time instance when the $(s+1)^{\text{th}}$ subsystem is switched on. The analysis is carried out separately over the time intervals $[T_{s-1}, T_s)$, $s \in \mathbb{N}$, where $T_1 \geq T_0 + \tau_1 + \tau_2 + t_M$. Since the history stack $\mathcal{H}$ is not updated over the intervals $[T_{s-1}, T_s)$, $s \in \mathbb{N}$, the matrices $\mathscr{G}_s$ and Q_s are constant over each interval. The history stack that is active over the interval $[T_s, T_{s+1})$ is denoted by $\mathcal{H}_{s+1}$. To ensure boundedness of the trajectories in the interval $t \in [T_0, T_1)$, the history stack $\mathcal{H}_1$ is computed using arbitrarily selected trajectories $x(\cdot), \hat{x}(\cdot), u(\cdot)$ that are confined within $\tilde{\chi}$ and make $\mathcal{H}_1$ full rank[5]. The analysis is carried out over the aforementioned intervals using the state vectors $Z := \begin{bmatrix} \tilde{x}_1^T & \tilde{x}_2^T & r^T & \eta^T & \operatorname{vec}\left(\tilde{\theta}\right)^T \end{bmatrix}^T \in \mathbb{R}^{n_1+3n_2+p}$ and $Y := \begin{bmatrix} \tilde{x}_1^T & \tilde{x}_2^T & r^T & \eta^T \end{bmatrix}^T \in \mathbb{R}^{n_1+3n_2}$.

A summary of the stability analysis is provided in the following, along with a graphical representation in Fig. 2.

Interval 1: First, it is established that Z is bounded over $[T_0, T_1)$, where the bound is $O\left(\left\|Z\left(T_0\right)\right\| + \left\|\sum_{i=1}^{M} \mathcal{E}_{1i}\right\| + \overline{\epsilon}\right)$[6]. Then, for a given $\varepsilon \in \mathbb{R}_{>0}$, the bound on Z is utilized to select state estimator gains such that $\left\|Y\left(T_1\right)\right\| < \varepsilon$.

Interval 2: The history stack $\mathcal{H}_2$, which is active over $[T_1, T_2)$, is recorded over $[T_0, T_1)$. Without loss of generality, it can be ensured that $\mathcal{H}_2$ represents the system better than $\mathcal{H}_1$ (which is arbitrarily selected), that is, $\left\|\sum_{i=1}^{M} \mathcal{E}_{1i}\right\| \geq \left\|\sum_{i=1}^{M} \mathcal{E}_{2i}\right\|$. The bound on Z over $[T_1, T_2)$ is then shown to be smaller than that over $[T_0, T_1)$, which is utilized to show that $\left\|Y\left(t\right)\right\| \leq \varepsilon$, for all $t \in [T_1, T_2)$.

Interval 3: Using (24), the errors $\mathcal{E}_{3i}$ are shown to be $O\left(\left\|Y_{3i}\right\| + \overline{\epsilon}\right)$ where Y_{3i}

[5] Arbitrary selection of $\mathcal{H}_1$ results in potentially large initial error $\mathcal{E}_1$ in (20). While large $\mathcal{E}_1$ could potentially result in large parameter estimation errors, $\tilde{\theta}$, during $[T_0, T_1)$, as long as $\mathcal{H}_1$ is full rank, the first term in (31) ensures that $\tilde{\theta}$ remains bounded over $[T_0, T_1)$.

[6] $f \in O\left(g\right)$ denotes that there exists $c, M \in \mathbb{R}_{>0}$ such that $|f(x)| \leq c|g(x)| \;\forall\; x > M$.

denotes the value of Y at the time when the point $\left(X_{3i}, \hat{F}_{3i}, \hat{G}_{3i}\right)$ was recorded. Using the facts that the history stack $\mathcal{H}_3$, which is active over $[T_2, T_3)$, is recorded over $[T_1, T_2)$ and $\left\|Y(t)\right\| \leq \varepsilon$, for all $t \in [T_1, T_2)$, the error $\left\|\sum_{i=1}^{M} \mathcal{E}_{3i}\right\|$ is shown to be $O(\varepsilon + \overline{\epsilon})$. If $T_3 = \infty$ then it is established that $\limsup_{t\to\infty} \left\|Z(t)\right\| = O(\varepsilon + \overline{\epsilon})$. If $T_3 < \infty$ then the fact that the bound on Z over $[T_2, T_3)$ is smaller than that over $[T_1, T_2)$ is utilized to show that $\left\|Y(t)\right\| \leq \varepsilon$, for all $t \in [T_2, T_3)$. The analysis is then continued in an inductive argument to show that $\limsup_{t\to\infty} \left\|Z(t)\right\| = O(\varepsilon + \overline{\epsilon})$ and $\left\|Y(t)\right\| \leq \varepsilon$, for all $t \in [T_2, \infty)$.

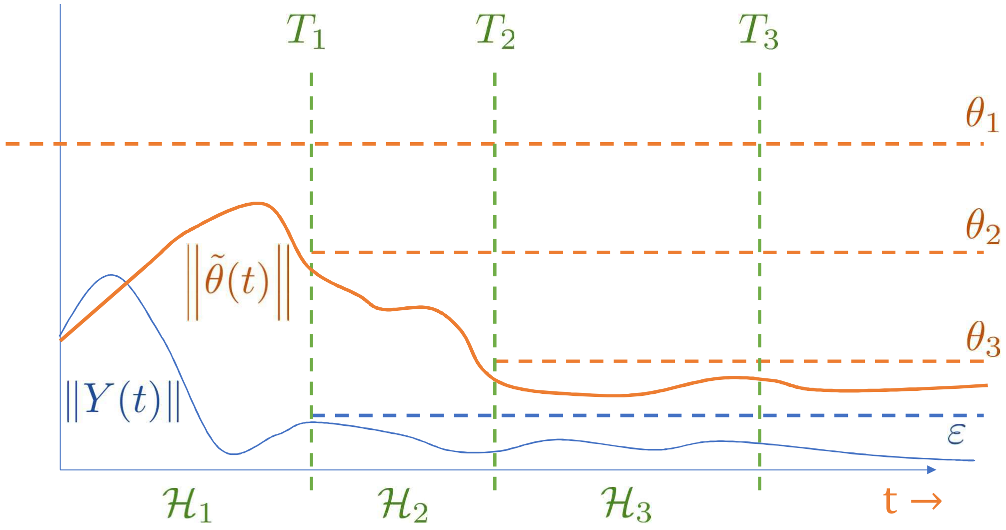


Figure 2: Error signals utilized in the stability analysis.

The stability result is summarized in the following theorem.

Theorem 1 *Let $\varepsilon > 0$ be given. If Assumptions 1 and 2 hold, the history stacks $\mathcal{H}$ and $\mathcal{G}$ are populated using the algorithm detailed in Fig. 1, the learning gains selected to satisfy the sufficient gain conditions in (38), (39), (44), and (48), there exists a time instance $T \in \mathbb{R}_{>0}$ such that the system states are informative over $[T_0, T]$, that is, the history stack can be replenished if purged at any time $t \in [T_0, T]$, over each switching interval $\left\{t \mid \rho(t) = s\right\}$, let the dwell-time, $\mathcal{T}$, is selected such that $\mathcal{T}(t) = \mathcal{T}_s$, where $\mathcal{T}_s$ is selected to be large enough to satisfy (47), and if the excitation* *interval is large enough so that* $T_2 < T$,[7] *then* $\limsup_{t\to\infty}\left\Vert Z(t)\right\Vert = O(\varepsilon + \overline{\epsilon})$.

Proof. Consider the candidate Lyapunov function

$$V(Z,t) := \frac{\alpha^2}{2}\sum_{j=1}^{2}\tilde{x}_j^T\tilde{x}_j + \frac{1}{2}r^Tr + \frac{1}{2}\eta^T\eta + \frac{1}{2}\mathrm{tr}\left(\tilde{\theta}^T\Gamma^{-1}(t)\,\tilde{\theta}\right). \tag{34}$$

Using arguments similar to [60, Corollary 4.3.2], it can be shown that provided $\lambda_{\min}\left\{\Gamma^{-1}(T_0)\right\} > 0$ and Assumption 2 holds, the least squares gain matrix satisfies

$$\underline{\Gamma}\mathrm{I}_p \le \Gamma(t) \le \overline{\Gamma}\mathrm{I}_p, \forall t \in \mathbb{R}_{\ge T_0}, \forall\, T_0 \ge 0, \tag{35}$$

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

The bound in (35) implies that the candidate Lyapunov function satisfies

$$\underline{v}\left\Vert Z\right\Vert^2 \le V(Z,t) \le \overline{v}\left\Vert Z\right\Vert^2, \tag{36}$$

where $\overline{v} := \frac{1}{2}\max\left\{1, \alpha^2, {}^1\!/\underline{\Gamma}\right\}$ and $\underline{v} := \frac{1}{2}\min\left\{1, \alpha^2, {}^1\!/\overline{\Gamma}\right\}$.

Over the time interval $[T_{s-1}, T_s)$, the orbital derivative of V is given by[8,9]

$$\dot{V}_s = -\alpha^3\sum_{j=1}^{2}\tilde{x}_j^T\tilde{x}_j + \alpha^2\sum_{j=1}^{2}\tilde{x}_j^T\tilde{f}_j - kr^Tr - (\beta-\alpha)\eta^T\eta + r^T\tilde{f}^o + r^T\theta^T\tilde{\sigma} + r^T\tilde{\theta}^T\sigma$$
$$-r^T\tilde{\theta}^T\tilde{\sigma} + r^T\epsilon + \alpha r^T\tilde{f}_2 - k_\theta\mathrm{tr}\left(\tilde{\theta}^TQ_s\right) - \frac{1}{2}\mathrm{tr}\left(\tilde{\theta}^T\left(k_\theta\mathscr{G}_s + \beta_1\Gamma^{-1}\right)\tilde{\theta}\right).$$

Assuming that $\mathcal{H}_s$ was computed using values of $\hat{x}$ that correspond to trajectories that stay inside $\tilde{\chi}$, the orbital derivative can be bounded by

$$\dot{V}_s \le -\alpha^3\sum_{j=1}^{2}\left\Vert\tilde{x}_j\right\Vert^2 + \alpha^2L\sum_{j=1}^{2}\left\Vert\tilde{x}_j\right\Vert^2 + 2\alpha^2L\Vert\tilde{x}_1\Vert\Vert\tilde{x}_2\Vert - k\left\Vert r\right\Vert^2 - (\beta-\alpha)\left\Vert\eta\right\Vert^2$$
$$+L_f\left\Vert r\right\Vert\left\Vert\tilde{x}\right\Vert + \left\Vert r\right\Vert\overline{\theta}L_\sigma\left\Vert\tilde{x}\right\Vert + L_\sigma\left\Vert r\right\Vert\left\Vert\tilde{\theta}\right\Vert\left\Vert\tilde{x}\right\Vert + \overline{\sigma}\left\Vert r\right\Vert\left\Vert\tilde{\theta}\right\Vert + \left\Vert r\right\Vert\overline{\epsilon} - \frac{1}{2}\underline{a}\left\Vert\tilde{\theta}\right\Vert^2$$
$$+\alpha L\Vert r\Vert\Vert\tilde{x}_1\Vert + \alpha L\Vert r\Vert\Vert\tilde{x}_2\Vert + k_\theta\left\Vert\tilde{\theta}\right\Vert\overline{Q}_s, \tag{37}$$

[7] A minimum of two purges are required to remove the randomly initialized data, and the data recorded during transient phase of the derivative estimator from the history stack.

[8] $\dot{V}(Z,t) := \frac{\partial V}{\partial Z}(Z,t)\,h_Z(Z,t) + \frac{\partial V}{\partial t}(Z,t)$ where $h_Z : \mathbb{R}^{n_1+3n_2+p} \times \mathbb{R}_{\ge T_0} \to \mathbb{R}^{n_1+3n_2+p}$ is constructed using (18), (17), (26), (31), and (32) so that $\dot{Z} = h_Z(Z,t)$.

[9] For brevity, function dependencies will be omitted over the rest of the analysis.

where $\underline{a} = k_{\theta}\underline{c} + \frac{\beta_1}{\overline{\Gamma}}$, $\overline{Q}_s$ is a positive constant such that $\overline{Q}_s \geq \|Q_s\|$, and the bounds $L, L_f, L_{\sigma}, \overline{\epsilon}$, and $\overline{\sigma}$ depend on the compact set $\tilde{\chi}$. Provided

$$\begin{aligned} & k > 2\left(L_f + \overline{\theta}L_{\sigma}\right) + \frac{2}{\underline{a}}\overline{\sigma}^2 + \frac{2}{\alpha^3}\left(L_f + \overline{\theta}L_{\sigma} + \alpha L\right)^2, \\ & \alpha^2 > 4\alpha L + 2L^2, \\ & \beta > \alpha, \end{aligned} \tag{38}$$

then (37) simplifies to

$$\begin{aligned} \dot{V}_s \leq & -\frac{\alpha^3}{4}\left\|\tilde{x}_1\right\|^2 - \frac{\alpha^3}{4}\left\|\tilde{x}_2\right\|^2 - \frac{k}{4}\left\|r\right\|^2 - \frac{\underline{a}}{16}\left\|\tilde{\theta}\right\|^2 - (\beta - \alpha)\left\|\eta\right\|^2 \\ & - \left(\frac{k}{4} - \frac{2L_{\sigma}^2}{\underline{a}}\left\|\tilde{x}\right\|^2\right)\left\|r\right\|^2 + \frac{\overline{\epsilon}^2}{2k} + \frac{4k_{\theta}^2}{\underline{a}}\overline{Q}_s^2. \end{aligned}$$

Since $\left\|\tilde{x}\right\|^2 \leq \left\|Z\right\|^2$, $\dot{V}_s \leq -v\left(\left\|Z\right\|^2 - \frac{\iota_s}{v}\right)$ in the domain

$$\mathcal{D} := \left\{ Z \in \mathbb{R}^{n_1+3n_2+p} \mid \|Z\| < \sqrt{\frac{k\underline{a}}{8L_{\sigma}^2}} \right\}.$$

That is, $\dot{V}_s$ is negative definite on $\mathcal{D}$ provided $\mathcal{H}_s$ was computed using values of $\hat{x}$ that correspond to trajectories that stay inside $\tilde{\chi}$, and provided $\|Z\| > \sqrt{\frac{\iota_s}{v}} > 0$, where

$$v := \frac{1}{4}\min\left\{\alpha^3, k, 4(\beta - \alpha), \underline{a}\frac{1}{4}\right\},$$

and $\iota_s := \frac{\overline{\epsilon}^2}{2k} + \frac{4k_{\theta}^2}{\underline{a}}\overline{Q}_s^2$. Theorem 4.18 from [80] can be invoked to conclude that provided the gain condition

$$k > \frac{8L_{\sigma}^2}{\underline{a}v}\max\left(\overline{V}_s, \frac{\overline{v}\iota_s}{v}\right), \tag{39}$$

holds, where $\overline{V}_s \geq \left\|V\left(Z\left(T_{s-1}\right), T_{s-1}\right)\right\|$ is a constant, then

$$\dot{V}_s\left(Z\left(t\right), t\right) \leq -\frac{v}{\overline{v}}V_s\left(Z\left(t\right), t\right) + \iota_s, \ \forall t \in \left[T_{s-1}, T_s\right).$$

In particular, by initializing $\mathcal{H}_1$ using arbitrary values of $\hat{x}$ that satisfy $x - \hat{x} \in \tilde{\chi}|_{\tilde{x}}$ for all $x \in \chi|_x$, it can be concluded that $\forall t \in \left[T_0, T_1\right)$,

$$V\left(Z\left(t\right), t\right) \leq \left(\overline{V}_1 - \frac{\overline{v}}{v}\iota_1\right)\mathrm{e}^{-\frac{v}{\overline{v}}(t-T_0)} + \frac{\overline{v}}{v}\iota_1, \tag{40}$$

where $\overline{V}_1 > 0$ is a constant such that $\left|V\left(Z\left(T_0\right), T_0\right)\right| \leq \overline{V}_1$. Using the relationships in (36) and (40), it can further be concluded $\forall t \in \left[T_0, T_1\right)$,

$$\left\|\tilde{\theta}\left(t\right)\right\| \leq \theta_1 := \sqrt{\frac{\overline{v}}{\underline{v}}} \max\left\{\sqrt{\overline{V}_1}, \sqrt{\frac{\overline{v}}{v}\iota_1}\right\}. \tag{41}$$

If it were possible to use the inequality in (40) to conclude that $V\left(Z\left(t\right), t\right) \leq V\left(Z\left(T_0\right), T_0\right)$, then an inductive argument could be used to show that the trajectories decay to a neighborhood of the origin. However, unless the history stack can be selected to have arbitrarily large minimum singular value (which is generally not possible), the constant $\frac{\overline{v}}{v}\iota_1$ cannot be made arbitrarily small using the learning gains.

Since ι_s depends on Q_s, it can be made smaller by reducing the estimation errors and thereby reducing the errors associated with the data stored in the history stack. To that end, consider the candidate Lyapunov function

$$W\left(Y\right) := \frac{\alpha^2}{2}\sum_{j=1}^{2}\tilde{x}_j^T\tilde{x}_j + \frac{1}{2}r^T r + \frac{1}{2}\eta^T\eta. \tag{42}$$

The candidate Lyapunov function satisfies

$$\underline{w}\left\|Y\right\|^2 \leq W\left(Y, t\right) \leq \overline{w}\left\|Y\right\|^2, \tag{43}$$

where $\overline{w} := \frac{1}{2}\max\left\{1, \alpha^2\right\}$, $\underline{w} := \frac{1}{2}\min\left\{1, \alpha^2\right\}$.

The orbital derivative of W is given by[10]

$$\begin{aligned}\dot{W} =& -\alpha^3\sum_{j=1}^{2}\tilde{x}_j^T\tilde{x}_j + \alpha^2\sum_{j=1}^{2}\tilde{x}_j^T\tilde{f}_j - kr^T r - (\beta-\alpha)\eta^T\eta \\ &+ \alpha r^T\tilde{f}_2 + r^T\left(\tilde{f}^o + \left(\theta^T - \tilde{\theta}^T\right)\tilde{\sigma}\right) + r^T\left(\tilde{\theta}^T\sigma + \epsilon\right).\end{aligned}$$

If $\tilde{\theta}(t)$ is bounded over $[T_{s-1}, T_s)$, then using Cauchy-Schwartz inequality, the

[10] $\dot{W}\left(Y, t\right) := \frac{\partial V}{\partial Y}\left(Y, t\right)h_Y\left(Y, t\right) + \frac{\partial V}{\partial t}\left(Y, t\right)$ where $h_Y : \mathbb{R}^{n_1+3n_2} \times \mathbb{R}_{\geq T_0} \to \mathbb{R}^{n_1+3n_2}$ is constructed using (18), (17), and (32) so that $\dot{Y} = h_Y\left(Y, t\right)$.

orbital derivative can be simplified and bounded over $[T_{s-1}, T_s)$ as

$$\begin{aligned}\dot{W}_s \leq & -\alpha^3 \sum_{i=1}^{2} \left\|\tilde{x}_j\right\|^2 + \alpha^2 L \sum_{i=1}^{2} \left\|\tilde{x}_j\right\| \left\|\tilde{x}^-\right\| - k\left\|r\right\|^2 - (\beta - \alpha)\left\|\eta\right\|^2 \\ & + \left(L_f + \left(\overline{\theta} + \theta_s\right) L_\sigma\right) \left\|r\right\| \left\|\tilde{x}\right\| + \alpha \left\|r\right\| \left\|\tilde{x}^-\right\| + \left(\theta_s \overline{\sigma} + \overline{\epsilon}\right) \left\|r\right\|,\end{aligned}$$

where $\theta_s > 0$ is a constant such that

$$\theta_s \geq \sup_{t \in [T_{s-1}, T_s)} \left\|\tilde{\theta}\left(t\right)\right\|.$$

In particular, consider the time interval $[T_0, T_1)$. Using the fact that $\tilde{\theta}(t)$ is bounded over $t \in [T_0, T_1)$, provided

$$\begin{aligned}& k > \frac{3}{\alpha} + \frac{6}{\alpha^3}\left(L_f + \left(\overline{\theta} + \theta_1\right) L_\sigma\right)^2 + 2\left(L_f + \left(\overline{\theta} + \theta_1\right) L_\sigma\right), \\ & \alpha^3 > 8\alpha^2 L + \frac{2}{\alpha} L^2, \\ & \beta > \alpha, \end{aligned} \tag{44}$$

then the time-derivative of W over $[T_0, T_1)$ can be bounded as

$$\dot{W}_1 \leq -\frac{w}{\overline{w}} W_1 + \gamma, \quad \text{where } w := \frac{1}{2} \min\left\{\frac{\alpha^3}{4}, k, 2(\beta - \alpha)\right\},$$

and $\gamma = \frac{(\theta_1 \overline{\sigma} + \overline{\epsilon})^2}{2k}$. That is, for all $t \in [T_0, T_1)$,

$$W\left(Y\left(t\right), t\right) \leq \left(\overline{W}_1 - \frac{\overline{w}}{w}\gamma\right) \mathrm{e}^{-\frac{w}{\overline{w}}(t - T_0)} + \frac{\overline{w}}{w}\gamma, \tag{45}$$

where $\overline{W}_1 > 0$ is a constant such that $\left|W\left(Y\left(T_0\right)\right)\right| \leq \overline{W}_1$. In particular, $\forall t \in [T_0, T_1)$.

$$\left\|Y\left(t\right)\right\| \leq \sqrt{\frac{\overline{w}}{\underline{w}} \max\left(\overline{W}_1, \frac{\overline{w}}{w}\gamma\right)} =: \overline{Y}_1. \tag{46}$$

Provided the dwell time $\mathcal{T}_1$ is large enough so that

$$\begin{aligned}\left(\overline{W}_1 - \frac{\overline{w}}{w}\gamma\right) \mathrm{e}^{-\frac{w}{\overline{w}}\mathcal{T}_1} & \leq \frac{\overline{w}}{w}\gamma, \\ \left(\overline{V}_1 - \frac{\overline{v}}{v}\iota_s\right) \mathrm{e}^{-\frac{v}{\overline{v}}\mathcal{T}_1} & \leq \frac{\overline{v}}{v}\iota_1, \end{aligned} \tag{47}$$

then from (40) and (45), $W\left(Y\left(T_1\right)\right) \leq \frac{2\overline{w}\gamma}{w}$ and $V\left(Z\left(T_1\right), T_1\right) \leq \frac{2\overline{v}\iota_1}{v}$. In particular, $\left\|Y\left(T_1\right)\right\| \leq \sqrt{\frac{2\overline{w}^2\gamma}{\underline{w}w}}$ and $\left\|Z\left(T_1\right)\right\| \leq \sqrt{\frac{2\overline{v}^2\iota_1}{\underline{v}v}}$. Note that the bound on $Y\left(T_1\right)$ can be made arbitrarily small by increasing k, α, and β.

Now the interval $[T_1, T_2)$ is considered. Given any arbitrary bound $\overline{W}_1$, a compact set $\tilde{\chi}$, and the learning gains that satisfy the resulting gain conditions in (38), (39), and (44), can be selected such that $\overline{\mathrm{B}}(0, \overline{Y}_1)$[11]$\subseteq \tilde{\chi}$, and as a result from (46) it follows that $\tilde{x}(t) \in \tilde{\chi}|_{\tilde{x}}$ for all $t \in [T_0, T_1)$. Since the history stack $\mathcal{H}_2$, which is active during $[T_1, T_2)$, is recorded during $[T_0, T_1)$, the bound in (24) can be used to show that $\overline{Q}_2 = O\left(\overline{Y}_1 + \overline{\epsilon}\right)$.

Since $\mathcal{H}_1$ is independent of the system trajectories, $\overline{Q}_1$ can be selected, without loss of generality, such that $\overline{Q}_2 < \overline{Q}_1$, and hence, $\iota_2 < \iota_1$. Thus, provided the constant $\overline{V}_1$ (and as a result, the gain k) is selected large enough so that

$$\frac{2\overline{v}\iota_1}{v} < \overline{V}_1, \tag{48}$$

the gain condition in (39) holds over $[T_1, T_2)$, and hence, a similar Lyapunov-based analysis, along with the bound $\overline{V}_2 = \frac{2\overline{v}\iota_1}{v}$ can be utilized to conclude that $\forall t \in [T_1, T_2)$,

$$\left\|\tilde{\theta}\left(t\right)\right\| \leq \sqrt{\frac{\overline{v}^2}{\underline{v}v}} \max\left\{\sqrt{2\iota_1}, \sqrt{\iota_2}\right\} =: \theta_2. \tag{49}$$

The sufficient condition in (48) implies that $\overline{V}_2 < \overline{V}_1$ and hence, (41) and $\iota_2 < \iota_1$ imply that $\theta_2 < \theta_1$.

Since $\theta_2 < \theta_1$, the gain conditions in (44) hold over the interval $[T_1, T_2)$. A Lyapunov-based analysis similar to (42)-(46) yields $\left\|Y\left(t\right)\right\| \leq \sqrt{\frac{\overline{w}}{\underline{w}} \max\left(\overline{W}_2, \frac{\overline{w}}{w}\gamma\right)}$. From (47), $\overline{W}_2 = \frac{2\overline{w}\gamma}{w}$, and hence, $\forall t \in [T_1, T_2)$,

$$\left\|Y\left(t\right)\right\| \leq \sqrt{\frac{2\overline{w}^2\gamma}{\underline{w}w}} := \overline{Y}_2. \tag{50}$$

Now, the interval $[T_2, T_3)$ is considered. By selecting $\overline{W}_1$ large enough, it can be ensured that $\overline{Y}_2 < \overline{Y}_1$, and as a result, $\tilde{x}(t) \in \tilde{\chi}|_{\tilde{x}}, \forall t \in [T_1, T_2)$. Since the history stack

[11] $\overline{\mathrm{B}}(0, \overline{Y})$ denotes the closed ball of radius $\overline{Y}$ around the origin.

$\mathcal{H}_3$, which is active during $[T_2, T_3)$, is recorded during $[T_1, T_2)$, the bounds in (24) and (50) can be used to show that $\overline{Q}_3 = O\left(\overline{Y}_2 + \overline{\epsilon}\right)$. Since $\overline{Y}_2 < \overline{Y}_1$, it follows that $\overline{Q}_3 < \overline{Q}_2$, which implies $\iota_3 < \iota_2$. Provided $\mathcal{T}_2$ satisfies (47), then $\left(\overline{V}_2 - \frac{\overline{v}}{v}\iota_2\right) \mathrm{e}^{-\frac{v}{\overline{v}}(T_2 - T_1)} \leq \frac{\overline{v}}{v}\iota_2$, which implies $\overline{V}_3 = \frac{2\overline{v}}{v}\iota_2$, and hence, $\overline{V}_3 < \overline{V}_2$ and $\theta_3 < \theta_2$. Therefore, the gain conditions in (38), (39), and (44) are satisfied over $[T_2, T_3)$.

Since the gain conditions are satisfied, a Lyapunov-based analysis similar to (42) - (46) yields $\left\|Y(t)\right\| \leq \sqrt{\frac{2\overline{w}^2\gamma}{\underline{w}w}}, \forall t \in [T_2, T_3)$. Given any $\varepsilon > 0$, the gains α, β, and k can be selected large enough to satisfy $\overline{Y}_2 \leq \varepsilon$, and hence, $\left\|Y(t)\right\| \leq \varepsilon, \forall t \in [T_2, T_3)$. Furthermore, a similar Lyapunov-based analysis as (34) - (40) yields $V\left(Z(t), t\right) \leq \left(\overline{V}_3 - \frac{\overline{v}}{v}\iota_3\right) \mathrm{e}^{-\frac{v}{\overline{v}}(t - T_2)} + \frac{\overline{v}}{v}\iota_3, \forall t \in [T_2, T_3)$. If $T_3 = \infty$ then $\limsup_{t\to\infty} V\left(Z(t), t\right) \leq \frac{2\overline{v}}{v}\iota_3$, which, from $\overline{Q}_3 = O\left(\overline{Y}_2 + \overline{\epsilon}\right)$ and $\iota_3 = \frac{\overline{\epsilon}^2}{2k} + \frac{2k_\theta^2}{\underline{a}}\overline{Q}_3^2$ implies that $\limsup_{t\to\infty} \left\|Z(t)\right\|$ $= O(\varepsilon + \overline{\epsilon})$.

If $T_3 \neq \infty$ then an inductive continuation of the Lyapunov-based analysis to the time intervals $[T_{s-1}, T_s)$ shows that provided the dwell time $\mathcal{T}_s$ satisfies (47), then the gain conditions in (38), (39), and (44) are satisfied for all $t > T_3$, the state Y satisfies

$$\left\|Y(t)\right\| \leq \varepsilon, \forall t > T_1, \tag{51}$$

$\tilde{x}(t) \in \tilde{\chi}|_{\tilde{x}}, \forall\ t \geq T_0$, and $Q_s \leq Q_{s-1}$, $\iota_s \leq \iota_{s-1}$, $\overline{V}_s \leq \overline{V}_{s-1}$, and $\theta_s \leq \theta_{s-1}$, for all $s > 3$.

The bound in (51) and the fact that $\overline{Q}_s = O\left(\overline{Y}_{s-1} + \overline{\epsilon}\right)$ indicate that $\overline{Q}_s = O(\varepsilon + \overline{\epsilon}), \forall s \in \mathbb{N}$. Furthermore, $V\left(Z(t), t\right) \leq \left(\overline{V}_s - \frac{\overline{v}}{v}\iota_s\right) \mathrm{e}^{-\frac{v}{\overline{v}}(t - T_{s-1})} + \frac{\overline{v}}{v}\iota_s$, $\forall t \in [T_{s-1}, T_s)$, $\forall s \in \mathbb{N}$, which, along with the dwell time requirement, implies that $\limsup_{t\to\infty} V\left(Z(t), t\right) \leq \frac{2\overline{v}}{v}\iota_s$, and hence, $\limsup_{t\to\infty} \left\|Z(t)\right\| = O(\varepsilon + \overline{\epsilon})$.

■

3.3 Linear Systems

When the system under consideration is linear, parameter estimation can be directly achieved using measurements of x_1 and without using state estimation. The following section details an output-feedback parameter estimator using x_1 as the output. The accompanying state estimator for linear systems is a trivial application of the estimator in Section 3.2.3, and has been omitted.

3.3.1 Problem Formulation

Consider a linear system of the form

$$\{\dot{x}_i = x_{i+1}\}_{i=1}^{N-1}, \quad \dot{x}_N = Ax + Bu, \quad y = x_1, \tag{52}$$

where $x_1, x_2, \ldots, x_N \in \mathbb{R}^n$ denote the state variables, $x := \begin{bmatrix} x_1^T & x_2^T & \ldots & x_N^T \end{bmatrix}^T$ is the system state, $u \in \mathbb{R}^m$ is the controller, $A \in \mathbb{R}^{n \times Nn}$ and $B \in \mathbb{R}^{n \times m}$ denote the system matrices, and $y \in \mathbb{R}^n$ denotes the output. The objective is to design an adaptive estimator to identify the unknown matrices A and B, online, using input-output measurements.

3.3.2 Error System for Estimation

To obtain an error signal for parameter identification, the system in (52) is expressed in the form

$$\dot{x}_N = A_1 x_1 + A_2 x_2 + \ldots + A_N x_N + Bu, \tag{53}$$

where $A_1 \in \mathbb{R}^{n \times n}$, $A_2 \in \mathbb{R}^{n \times n}$, ..., and $A_N \in \mathbb{R}^{n \times n}$ are constant matrices such that $A = \begin{bmatrix} A_1 & A_2 & \ldots & A_N \end{bmatrix}$. Integrating (53) over the interval $[t - \tau_1, t]$ for some constant $\tau_1 \in \mathbb{R}_{>0}$,

$$\begin{aligned} x_N(t) - x_N(t - \tau_1) = A_1 \int_{t-\tau_1}^{t} x_1(\zeta_1)\,\mathrm{d}\zeta_1 + \ldots \\ + A_N \int_{t-\tau_1}^{t} x_N(\zeta_1)\,\mathrm{d}\zeta_1 + B \int_{t-\tau_1}^{t} u(\zeta_1)\,\mathrm{d}\zeta_1. \end{aligned} \tag{54}$$

Integrating again over the interval $[t-\tau_2, t]$ for some constant $\tau_2 \in \mathbb{R}_{>0}$,

$$\int_{t-\tau_2}^{t} \big(x_N(\zeta_2) - x_N(\zeta_2-\tau_1)\big)\,\mathrm{d}\zeta_2 = A_1 \int_{t-\tau_2}^{t}\int_{\zeta_2-\tau_1}^{\zeta_2} x_1(\zeta_1)\,\mathrm{d}\zeta_1\,\mathrm{d}\zeta_2 + \dots \\ + A_N \int_{t-\tau_2}^{t}\int_{\zeta_2-\tau_1}^{\zeta_2} x_N(\zeta_1)\,\mathrm{d}\zeta_1\,\mathrm{d}\zeta_2 + B\int_{t-\tau_2}^{t}\int_{\zeta_2-\tau_1}^{\zeta_2} u(\zeta_1)\,\mathrm{d}\zeta_1\,\mathrm{d}\zeta_2.$$

Using the Fundamental Theorem of Calculus and the fact that $x_N(t) = \dot{x}_{N-1}(t)$,

$$x_{N-1}(t) - x_{N-1}(t-\tau_1) - x_{N-1}(t-\tau_2) + x_{N-1}(t-\tau_1-\tau_2) \\ = A_1 \int_{t-\tau_2}^{t}\int_{\zeta_2-\tau_1}^{\zeta_2} x_1(\zeta_1)\,\mathrm{d}\zeta_1\,\mathrm{d}\zeta_2 + \dots + A_N \int_{t-\tau_2}^{t} \big(x_{N-1}(\zeta_2) - x_{N-1}(\zeta_2-\tau_1)\big)\,\mathrm{d}\zeta_2 \\ + B\int_{t-\tau_2}^{t}\int_{\zeta_2-\tau_1}^{\zeta_2} u(\zeta_1)\,\mathrm{d}\zeta_1\,\mathrm{d}\zeta_2. \quad (55)$$

Repeating this process $N-1$ more time, results in

$$x_1(t) - x_1(t-\tau_1) - \dots + x_1(t-\tau_1-\tau_2-\dots-\tau_N) \\ = A_1F_1(t) + A_2F_2(t) + \dots + A_NF_N(t) + BU(t), \quad (56)$$

where

$$F_1(t) := \begin{cases} \int_{t-\tau_N}^{t} \dots \int_{\zeta_2-\tau_1}^{\zeta_2} x_1(\zeta_1)\,\mathrm{d}\zeta_1 \dots \mathrm{d}\zeta_{N-1}, & t\in[\mathscr{T},\infty), \\ 0, & t < \mathscr{T}, \end{cases} \quad (57)$$

$$F_2(t) := \begin{cases} \int_{t-\tau_N}^{t} \cdots \int_{\zeta_3-\tau_2}^{\zeta_3} (x_1(\zeta_2) - x_1(\zeta_2-\tau_1))\,\mathrm{d}\zeta_2 \dots \mathrm{d}\zeta_{N-1}, & t \in [\mathscr{T},\infty), \\ 0, & t < \mathscr{T}, \end{cases} \quad (58)$$

$$\vdots$$

$$F_N(t) := \begin{cases} \int_{t-\tau_N}^{t} (x_1(\zeta_{N-1}) + x_1(\zeta_{N-1}-\tau_1) + \dots)\,\mathrm{d}\zeta_{N-1}, & t\in[\mathscr{T},\infty), \\ 0, & t < \mathscr{T}, \end{cases} \quad (59)$$

$$U(t) := \begin{cases} \int_{t-\tau_N}^{t} \ldots \int_{\zeta_2-\tau_1}^{\zeta_2} u(\zeta_1)\, \mathrm{d}\zeta_1 \ldots \mathrm{d}\zeta_{N-1}, & t \in [\mathscr{T}, \infty), \\ 0, & t < \mathscr{T}, \end{cases} \tag{60}$$

and $\mathscr{T} = T_0 + \tau_1 + \ldots + \tau_N$. As opposed to nonlinear systems in Section 3.2.2, where measurements of all states but the final state are required for parameter estimation, the integral form in (56) is independent of the state variables $x_2, \ldots, x_N$, and depends only on the output, $y = x_1$. The expression in (56) can be rearranged to form the linear error system

$$\mathcal{F}(t) = \mathcal{G}(t)\,\theta,\ \forall t \in \mathbb{R}_{\geq T_0}. \tag{61}$$

In (61), θ is a vector of unknown parameters, defined as $\theta := \begin{bmatrix}\mathrm{vec}(A_1)^T & \mathrm{vec}(A_2)^T & \ldots & \mathrm{vec}(A_N)^T & \mathrm{vec}(B)^T\end{bmatrix}^T \in \mathbb{R}^{Nn^2+mn}$, where $\mathrm{vec}(\cdot)$ denotes the vectorization operator and the matrices $\mathcal{F} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\mathcal{G} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n\times\left(Nn^2+mn\right)}$ are defined as

$$\mathcal{F}(t) := \begin{cases} x_1(t) - x_1(t-\tau_1) - \ldots, & t \in [\mathscr{T}, \infty), \\ 0 & t < \mathscr{T}, \end{cases}$$

$$\mathcal{G}(t) := \begin{bmatrix} \left(F_1(t) \otimes \mathrm{I}_n\right)^T & \ldots & \left(F_N(t) \otimes \mathrm{I}_n\right)^T & \left(U(t) \otimes \mathrm{I}_n\right)^T \end{bmatrix},$$

where I_n denotes an $n \times n$ identity matrix, and $\otimes$ denotes the Kronecker product. Note that even though the linear relationship in (61) is valid for all $t \in \mathbb{R}_{\geq T_0}$, it provides useful information about the vector θ only after $t \geq \mathscr{T}$.

The linear error system in (61) motivates the adaptive estimation scheme that follows.

3.3.3 Parameter Estimator Design

To obtain output-feedback concurrent learning update law for the parameter estimates, a history stack denoted by $\mathcal{H}$ is utilized. The history stack is a set of ordered pairs $\left\{(\mathcal{F}_i, \mathcal{G}_i)\right\}_{i=1}^{M}$ such that

$$\mathcal{F}_i = \mathcal{G}_i\theta,\ \forall i \in \{1, \cdots, M\}\,. \tag{62}$$

Note that $\mathcal{E}_i$ from (20) is absent from (62), since there are no estimated state variables in $\mathcal{F}_i$ or $\mathcal{G}_i$.

If a history stack that satisfies (63) is not available a priori, it can be recorded online, using the relationship in (61), by selecting a set of time-instances $\{t_i\}_{i=1}^{M}$ and letting

$$\mathcal{F}_i = \mathcal{F}\left(t_i\right), \quad \mathcal{G}_i = \mathcal{G}\left(t_i\right). \tag{63}$$

Furthermore, a singular value maximization algorithm is used to select the time instances $\{t_i\}_{i=1}^{M}$. That is, a data-point $\left\{\left(\mathcal{F}_j, \mathcal{G}_j\right)\right\}$ in the history stack is replaced by a new data-point $\left\{(\mathcal{F}^*, \mathcal{G}^*)\right\}$, where $\mathcal{F}^* = \mathcal{F}\left(t\right)$ and $\mathcal{G}^* = \mathcal{G}\left(t\right)$, for some t, only if

$$\lambda_{\min}\left\{\sum_{i\neq j}\mathcal{G}_i^T\mathcal{G}_i + \mathcal{G}_j^T\mathcal{G}_j\right\} < \lambda_{\min}\left\{\sum_{i\neq j}\mathcal{G}_i^T\mathcal{G}_i + \mathcal{G}^{*T}\mathcal{G}^*\right\},$$

where $\lambda_{\min}\left\{\cdot\right\}$ denotes the minimum Eigenvalue of a matrix.

Since the time instances, $\{t_i\}_{i=1}^{M}$, vary according to the minimum singular value maximization algorithm, the history stacks, $\mathcal{F}(t)$ and $\mathcal{G}(t)$, are time-varying and piece-wise constant. The following definition establishes a uniform lower bound for the time-varying history stacks to facilitate the analysis that directly follows.

Definition 1 *A history stack* $\left\{(\mathcal{F}_i, \mathcal{G}_i)\right\}_{i=1}^{M}$ *is called* uniformly full rank *if there exists a constant* $\underline{c} \in \mathbb{R}$ *such that*

$$0 < \underline{c} < \lambda_{\min}\left\{\mathscr{G}(t)\right\}, \forall t \geq T_0, \forall T_0 \in \mathbb{R}_{\geq 0}, \tag{64}$$

where the matrix $\mathscr{G} \in \mathbb{R}^{\left(Nn^2+mn\right)\times\left(Nn^2+mn\right)}$ *is defined as* $\mathscr{G} := \sum_{i=1}^{M}\mathcal{G}_i^T\mathcal{G}_i$.

The concurrent learning update law to estimate the unknown parameters is then given by

$$\dot{\hat{\theta}} = k_\theta\Gamma\sum_{i=1}^{M}\mathcal{G}_i^T\left(\mathcal{F}_i - \mathcal{G}_i\hat{\theta}\right), \tag{65}$$

and the least square update law is

$$\dot{\Gamma} = \beta_1 \Gamma - k_\theta \Gamma \mathscr{G} \Gamma. \tag{66}$$

Remark 3 *To facilitate the following Lyapunov analysis, using* (61) *and* (65), *the parameter estimation error can be expressed as*

$$\dot{\tilde{\theta}} = -k_\theta \Gamma \mathscr{G} \tilde{\theta}. \tag{67}$$

Since the function $\mathscr{G} : \mathbb{R}_{\geq T_0} \to \mathbb{R}^{\left(Nn^2+mn\right)\times\left(Nn^2+mn\right)}$ *is piece-wise continuous, the trajectories of (67) and all the subsequent functions involving* $\mathscr{G}$, *are defined in the sense of Carathéodory [54].*

3.3.4 Analysis

The following theorem establishes exponential convergence of the parameter estimates.

Theorem 2 *If there exists a time* T *such that the history stack* $\left\{\left(\mathcal{F}_i(T), \mathcal{G}_i(T)\right)\right\}_{i=1}^{M}$ *is uniformly full rank, then the parameter estimates,* $\hat{\theta}$, *updated using the parameter estimator in (65), converge to* θ^*, *exponentially over the interval* $[T, \infty)$.

Proof. Consider the following candidate Lyapunov function

$$V(\tilde{\theta}, t) = \tilde{\theta}^T \Gamma^{-1}(t) \tilde{\theta}. \tag{68}$$

Using arguments similar to [60, Corollary 4.3.2], it can be shown that provided $\lambda_{\min}\left\{\Gamma^{-1}\left(T_0\right)\right\} > 0$ and Assumption 2 holds, the least squares gain matrix satisfies

$$\underline{\Gamma} \mathrm{I}_{\left(Nn^2+mn\right)} \leq \Gamma(t) \leq \overline{\Gamma} \mathrm{I}_{\left(Nn^2+mn\right)}, \forall t \in \mathbb{R}_{\geq T_0}, \forall T_0 \in \mathbb{R}_{\geq 0}. \tag{69}$$

The candidate Lyapunov function satisfies

$$\underline{\Gamma} \|\tilde{\theta}\|^2 \leq V(\tilde{\theta}, t) \leq \overline{\Gamma} \|\tilde{\theta}\|^2, \tag{70}$$

where (69) implies that the the bounds, $\overline{\Gamma}$ and $\underline{\Gamma}$, in (70) are established independent of T_0.

Using (65) and (66), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, the time-derivative of (68) results in[12]

$$\dot{V}(\tilde{\theta},t) = -2\tilde{\theta}^T\Gamma^{-1}(t)\Big(k_\theta\Gamma\,(t)\sum_{i=1}^{M}\mathcal{G}_i^T\,(t)\Big(\mathcal{F}_i(t) - \mathcal{G}_i(t)\hat{\theta}\Big)\Big)$$
$$- \tilde{\theta}^T\left(\Gamma^{-1}(t)\Big[\beta_1\Gamma(t) - k_\theta\Gamma(t)\mathscr{G}(t)\Gamma(t)\Big]\Gamma^{-1}(t)\right)\tilde{\theta}. \quad (71)$$

Simplifying (71), $\dot{V}(\tilde{\theta},t)$ becomes

$$\dot{V}(\tilde{\theta},t) = -k_\theta\tilde{\theta}^T\mathscr{G}(t)\tilde{\theta} - \beta_1\tilde{\theta}^T\Gamma^{-1}(t)\tilde{\theta}. \quad (72)$$

During the time interval $[T_0, T]$, when $\mathscr{G}$ is not full rank, Theorem 4.8 from [80] can be used to show uniform boundedness of $\tilde{\theta}$. Once the history stack becomes full rank in the sense of Def. 1, using (68) and (72), along with the bounds in (64) and (69), Theorem 4.10 from [80] can be invoked to conclude that $\tilde{\theta}$ converges to the origin, exponentially over the interval $[T, \infty)$. ■

3.4 Simulation

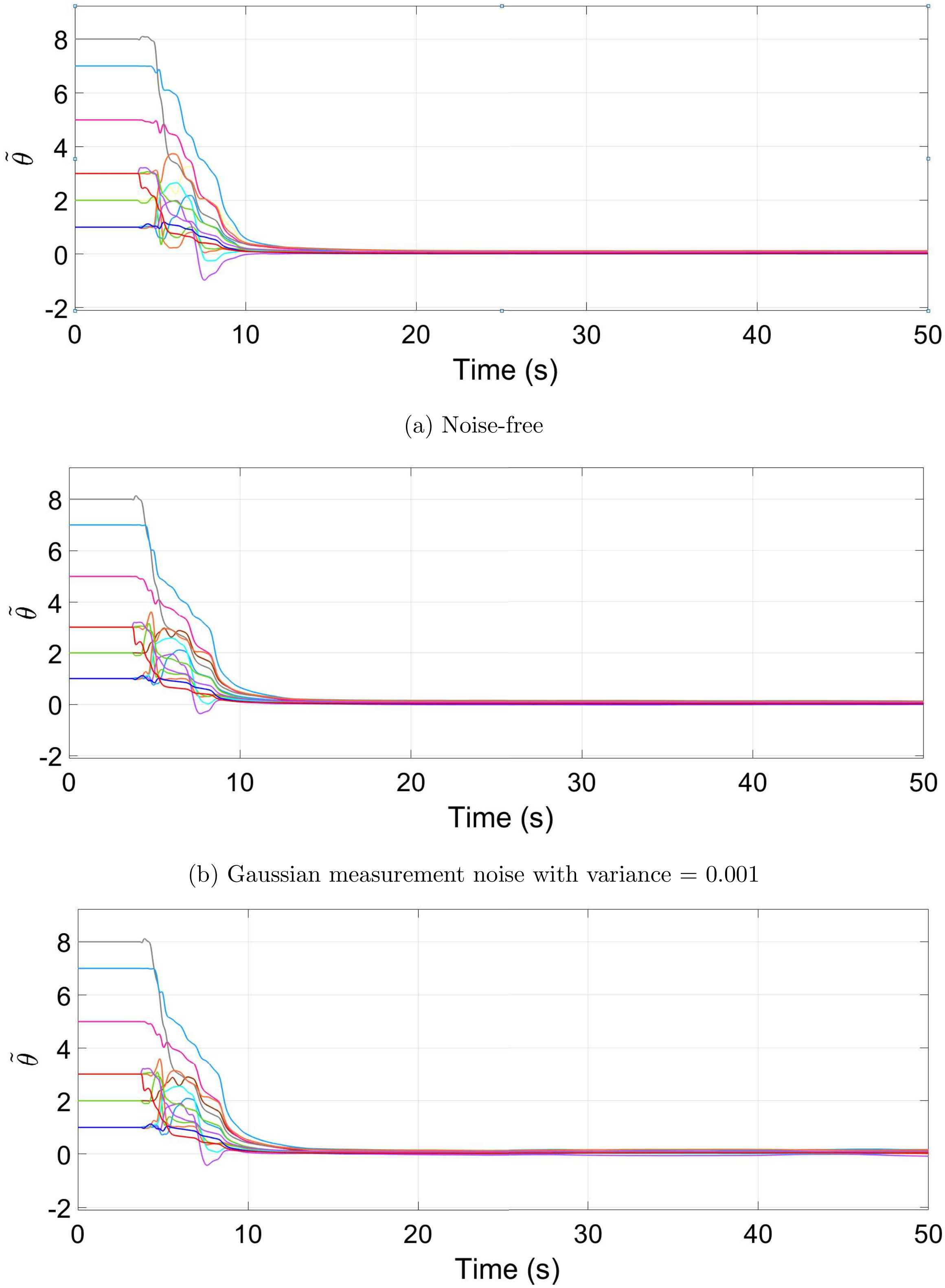
3.4.1 Linear System

The linear system selected for the simulation study is given by

$$\{\dot{x}_i = x_{i+1}\}_{i=1}^2, \dot{x}_3 = \begin{bmatrix} 2 & 3 & 1 & 5 & 7 & 3 \\ 1 & 2 & 1 & 8 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \end{bmatrix} u.$$

To satisfy Assumption 1, a controller that results in a uniformly bounded system response is needed. In this simulation study, the controller, u, is selected to be

[12] $\dot{V}\left(\tilde{\theta},t\right) := \frac{\partial V}{\partial \tilde{\theta}}\left(\tilde{\theta},t\right)h_\theta\left(\tilde{\theta},t\right) + \frac{\partial V}{\partial t}\left(\tilde{\theta},t\right)$ where $h_\theta : \mathbb{R}^{Nn^2+mn} \times \mathbb{R}_{\geq T_0} \to \mathbb{R}^{Nn^2+mn}$ is constructed using (61) and (65) so that $\dot{\tilde{\theta}} = h_\theta\left(\tilde{\theta},t\right)$.



(a) Noise-free

(b) Gaussian measurement noise with variance = 0.001

(c) Gaussian measurement noise with variance = 0.01

Figure 3: Parameter estimation errors for the linear system

a PD controller of the form $u = -k_p\left(x_1 - x_{d1}\right) - k_{d1}\left(x_2 - \dot{x}_{d1}\right) - k_{d2}\left(x_3 - \ddot{x}_{d1}\right)$ so that the system tracks the trajectory $x_{d1_1}\left(t\right) = x_{d1_2}\left(t\right) = -\frac{1}{3}\cos(3t) - \frac{1}{2}\cos(2t) - \cos(t) - \frac{1}{5}\cos(5t) - \frac{1}{7}\cos(7t) - \frac{1}{11}\cos(11t)$, uniformly in T_0, where the notation of x_{i_j} represents the j^{th} element of state x_i. Since there are fourteen unknown parameters, and the desired trajectory contains six distinct frequencies, the closed-loop system is not guaranteed to be persistently excited.

The simulation utilizes Euler forward numerical integration using a sample time of $\Delta_t = 0.001$ seconds. Past $\frac{\tau_1+\tau_2+\tau_3}{\Delta_t}$ values of the state, x_1, and the control input, u, are stored in a buffer. The matrices $\mathcal{F}$ and $\mathcal{G}$ for the parameter update law in (65) are computed using trapezoidal integration of the data stored in the aforementioned buffer. Values of $\mathcal{F}$ and $\mathcal{G}$ are stored in the history stack and are updated so as to maximize the minimum eigenvalue of $\mathscr{G}$.

The initial estimates of the unknown parameters are selected to be zero, and the history stack is initialized so that all the elements of the history stack are zero. Data is added to the history stack using a singular value maximization algorithm. To demonstrate the utility of the developed method, three simulation runs are performed. In the first run, the parameter estimator has access to noise free measurements of the output, x_1. In the second and the third runs, a zero-mean Gaussian noise with variance 0.001 and 0.01, respectively, is added to the output signal to simulate measurement noise. The values of various simulation parameters selected for the three runs are $\tau_1 = 1.5$, $\tau_2 = 1.2$, $\tau_3 = 1.0$, $N = 350$, $\Gamma\left(T_0\right) = \mathrm{I}_{14}$, $\beta_1 = 0.4$, $\alpha = 0.5$, $k = 10$, $\beta = 2$, $\alpha_1 = 1$, and $k_\theta = {}^{2}/_{N}$. Figure 3a demonstrates that in absence of noise, the developed parameter estimator drives the parameter estimation error, $\tilde{\theta}$, to the origin. Figures 3b and 3c indicate that the developed method is robust to measurement noise, and results in convergence rates that are similar to the noise-free case, with a small increase in the steady state error due to measurement noise.

A one-at-a-time sensitivity analysis was performed on the parameters $\tau_1, \tau_2, \tau_3, \beta_1,$

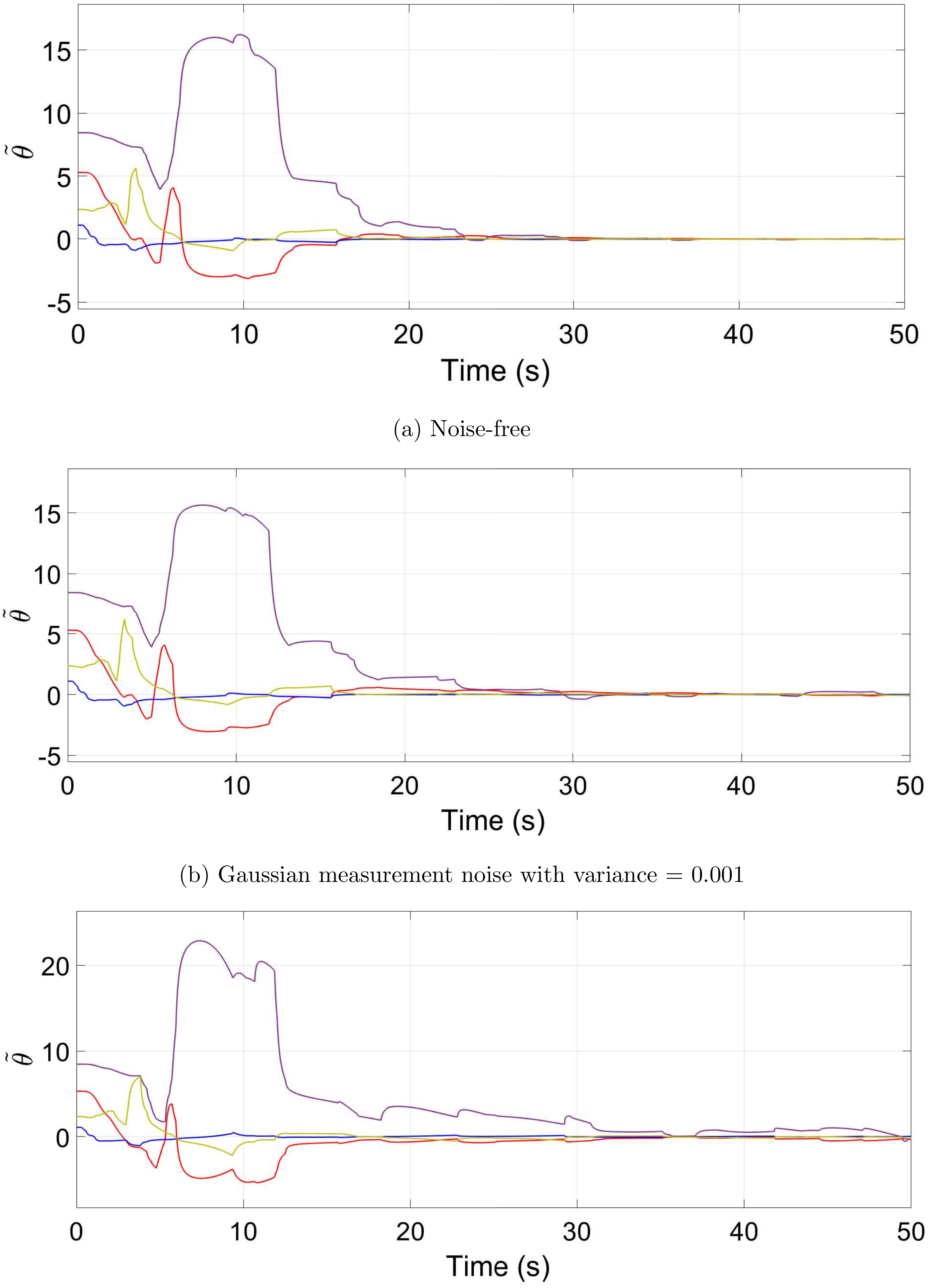
Table 1.: Sensitivity analysis for the linear system. The nominal values of τ_1, τ_2, τ_3, β_1, and k_θ were selected to be $\tau_1 = 1.5, \tau_2 = 1.2, \tau_3 = 1.0, \beta_1 = 0.4$, and $k_\theta = 2/N$. A zero-mean Gaussian noise with variance 0.001 was used with a step size $\Delta_t = 0.001$.

Parameter	Tested Values	RMS Error Variation	Steady-State RMS Error Variation
τ_1	1.1 - 2.0	55.91 - 64.21	0.1255 - 0.1548
τ_2	0.8 - 1.7	56.22 - 65.61	0.1134 - 0.1339
τ_3	0.6 - 1.5	56.98 - 64.98	0.1206 - 0.1337
β_1	0.05 - 0.9	58.50 - 63.04	0.1265 - 0.2509
k_θ	$^{0.5}/_{\mathrm{N}}$ - $^{4}/_{\mathrm{N}}$	58.14 - 62.62	0.1161 - 0.1266

and k_θ to gauge robustness of the developed technique. As demonstrated by the results in Table 1., the developed method is robust to small changes in the integration intervals and learning gains.

3.4.2 Nonlinear System

The developed state and parameter estimator is validated using a simulation study involving a two-link robot manipulator arm, where $x_1 \in \mathbb{R}^2$ denotes the angular position of the two links, $x_2 \in \mathbb{R}^2$ denotes the angular velocities of the two links, and $x = \begin{bmatrix} x_1^T & x_2^T \end{bmatrix}^T$. The selected model belongs to a sub-class of systems in (1), where the function approximation error, ε, is zero. The model is selected because the ideal parameters, θ, are known, and as a result, the model facilitates direct quantitative analysis of the parameter estimation error.



(a) Noise-free

(b) Gaussian measurement noise with variance = 0.001

(c) Gaussian measurement noise with variance = 0.01

Figure 4: Parameter estimation errors for the nonlinear system.

The nonlinear dynamics of the system are described by (1), where

$$f^0(x,u) = -\left(M(x_1)\right)^{-1} V_m(x_1,x_2)\,x_2 + \left(M(x_1)\right)^{-1} u,$$
$$g^T(x,u) = \theta^T \begin{bmatrix} \begin{bmatrix} \left(M(x_1)\right)^{-1} & \left(M(x_1)\right)^{-1} \end{bmatrix} D(x_2) \end{bmatrix}^T. \tag{73}$$

In (73), $u \in \mathbb{R}^2$ is the control input,

$$D(x_2) := \operatorname{diag}\left[\tanh\left(x_{2_1}\right),\ \tanh\left(x_{2_2}\right)\right],$$
$$M(x_1) := \begin{bmatrix} a_1 + 2a_3\mathrm{c}_2(x_1), & a_2 + a_3\mathrm{c}_2(x_1) \\ a_2 + a_3\mathrm{c}_2(x_1), & a_2 \end{bmatrix},$$

and

$$V_m(x_1,x_2) := \begin{bmatrix} -a_3\mathrm{s}_2(x_1)\,x_{2_2}, & -a_3\mathrm{s}_2(x_1)\left(x_{2_1}+x_{2_2}\right) \\ a_3\mathrm{s}_2(x_1)\,x_{2_1}, & 0 \end{bmatrix},$$

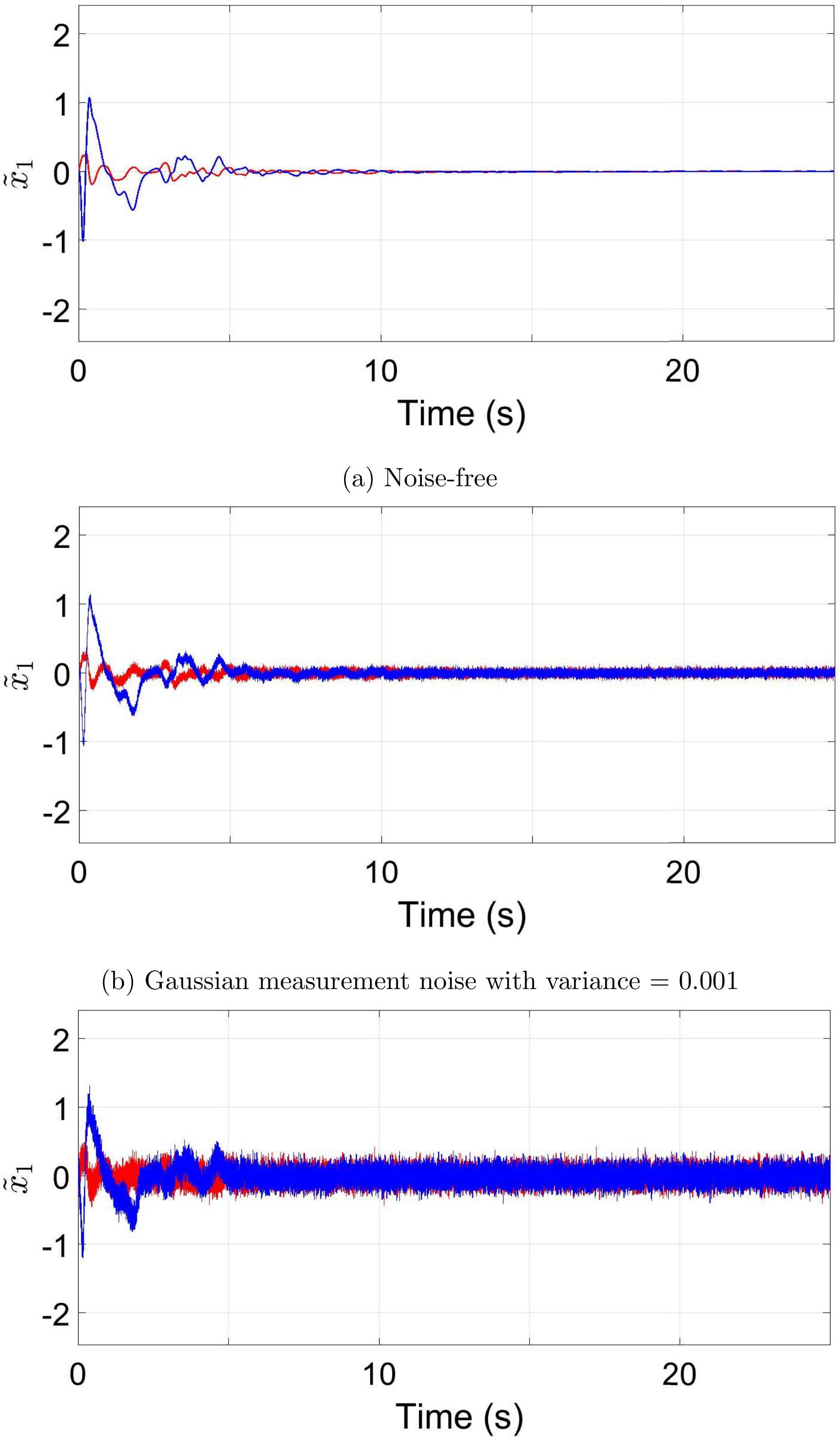
where $\mathrm{c}_2(x_1) = \cos\left(x_{1_2}\right)$, $\mathrm{s}_2(x_1) = \sin\left(x_{1_2}\right)$, and $a_1 = 3.473$, $a_2 = 0.196$, and $a_3 = 0.242$ are known constants. The system has four unknown parameters. The ideal values of the unknown parameters are $\theta = \begin{bmatrix} 5.3 & 1.1 & 8.45 & 2.35 \end{bmatrix}^T$.

To satisfy Assumption 1, a controller that results in a uniformly bounded system response is needed. In this simulation study, the controller, u, is selected to be a PD controller of the form $u = -k_p(x_1 - x_{d1}) - k_d(x_2 - \dot{x}_{d1})$ so that the system tracks the trajectory $x_{d1_1}(t) = x_{d1_2}(t) = -\frac{1}{3}\cos(3t) - \frac{1}{2}\cos(2t)$, uniformly in T_0.

The simulation utilizes Euler forward numerical integration using a sample time of $\Delta_t = 0.001$ seconds. Past $\frac{\tau_1+\tau_2}{\Delta_t}$ values of the output, x_1, state estimates, $\hat{x}$, and the control input, u, are stored in a buffer. The matrices P, $\hat{G}$, and $\hat{F}$ for the parameter update law in (25) are computed using trapezoidal integration of the data stored in the aforementioned buffer. Values of P, $\hat{G}$, and $\hat{F}$ are stored in the history stack and are updated according to the algorithm detailed in Fig. 1.

The initial estimates of the unknown parameters are selected to be zero, and the history stack is initialized so that all the elements of the history stack are zero[13].

[13] It is clear from the simulation results that full rank initialization of the history stack and the

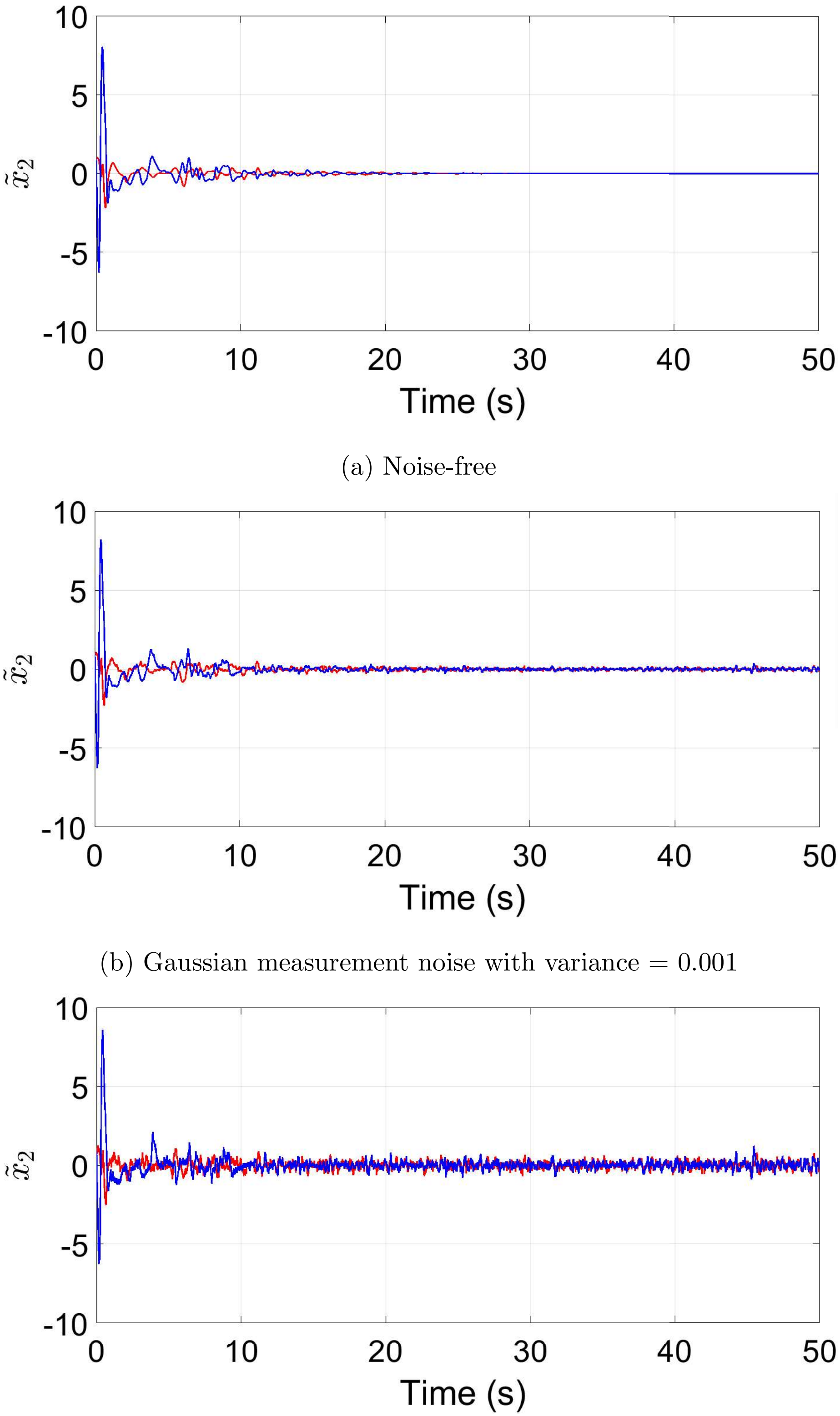


(a) Noise-free

(b) Gaussian measurement noise with variance = 0.001

(c) Gaussian measurement noise with variance = 0.01

Figure 5: x_1 state estimation errors for the nonlinear system.



(a) Noise-free

(b) Gaussian measurement noise with variance = 0.001

(c) Gaussian measurement noise with variance = 0.01

Figure 6: x_2 state estimation errors for the nonlinear system.

Table 2.: Sensitivity analysis for the nonlinear system. The nominal values of τ_1, τ_2, β_1, and k_θ were selected to be $\tau_1 = 1.2, \tau_2 = 0.9, \beta_1 = 0.7$, and $k_\theta = 0.5/N$. The zero-mean Gaussian noise with variance 0.001 was used with a step size $\Delta_t = 0.001$.

Parameter	Tested Values	RMS Error Variation	Steady-State RMS Error Variation
τ_1	0.8 - 1.7	0.998 - 3.848	0.0325 - 0.1339
τ_2	0.5 - 1.4	1.011 - 3.546	0.0294 - 0.1270
β_1	0.1 - 1.2	1.224 - 1.763	0.0324 - 0.3273
k_θ	$^{0.01}/_{\mathrm{N}}$ - $^{2}/_{\mathrm{N}}$	1.090 - 1.684	0.0296 - 0.0515

Data is added to the history stack using a singular value maximization algorithm. To demonstrate the utility of the developed method, three simulation runs are performed. In the first run, the observer is assumed to have access to noise free measurements of the output, x_1. In the second and third runs, a zero-mean Gaussian noise with variance 0.001 and variance 0.01 are added to the output signal to simulate measurement noise. The values of various simulation parameters selected for the three runs are $\tau_1 = 1.2$, $\tau_2 = 0.9$, $N = 150$, $\Gamma(T_0) = \mathrm{I}_4$, $\beta_1 = 0.7$ (0.9 for variance 0.01), $\alpha = 2$, $k = 10$, $\beta = 2$, $\alpha_1 = 1$, $\kappa = 0$, and $k_\theta = {}^{0.5}/_{N}$. Figures 4a - 6a demonstrate that in the absence of noise, the developed method drives the state estimation errors, $\tilde{x}$, and the parameter estimation errors, $\tilde{\theta}$, to a neighborhood of the origin. Figures 4b - 6c indicate that the developed technique can be utilized in the presence of measurement noise, with expected degradation of performance.

One-at-a-time sensitivity analysis was performed on the parameters τ_1, τ_2, β_1, and k_θ to gauge robustness of the developed technique. As demonstrated by the results in

normalization terms in (25) and (26) are sufficient, but not necessary conditions for the analysis in Section 3.2.6.

Table 2., the developed method is robust to small changes in the integration intervals and learning gains.

3.5 Conclusion

This chapter develops a concurrent learning based adaptive observer and parameter estimator to simultaneously estimate the unknown parameters and the states of linear and nonlinear systems using output measurements. The developed technique utilizes a dynamic state observer to generate state estimates necessary for data-driven adaptation. A purging algorithm is developed to improve the quality of the stored data as the state estimates converge to the true states.

The developed state and parameter estimation method allows for simultaneous estimation of the system states and uncertain parameters in the system model without the need for full state feedback, and facilitates parameter convergence without the requirement of PE. Theoretical guarantees for uniform ultimate boundedness of the estimation errors are established in the absence of measurement noise. Simulation results indicate that the developed method is robust to measurement noise and not sensitive to design parameters. For the class of linear systems presented, the parameter estimation can be performed independent of state estimation which facilitates exponential convergence of the parameter estimation errors. Future work will involve analyzing applicability of feedback linearization, along with a theoretical analysis of the developed method under measurement noise and process noise. A theoretical analysis of the effect of the integration intervals, τ_i, on the performance of the developed estimator will also be pursued.

Chapter IV

INVERSE REINFORCEMENT LEARNING IN REAL TIME

In this chapter, an output-feedback model-based inverse reinforcement learning method is developed for a class of linear and nonlinear systems. Real-time reward function estimation with sparse data points is shown to result in a unique solution in the presence of parametric uncertainties in the system dynamics and unmeasureable states.

4.1 Introduction

Based on the premise that the most succinct representation of the behavior of an entity is its reward structure [118], this chapter aims to recover the reward (or cost) function of an agent by observing the agent performing a task and monitoring its state and control trajectories. Methods to estimate the reward function using state and control trajectories fall under the umbrella of inverse reinforcement learning (IRL) (see, for example, [118] and [132]). The IRL method developed in this chapter learns the reward function and the value function of an agent under observation online, and in the presence of modeling uncertainties and unmeasurable states.

While IRL in an *offline* setting has a rich history of literature [3, 5, 92, 93, 116, 118, 126, 132, 143, 154, 157, 159, 160], traditional IRL methods typically require a large amount of training data. As such, *offline* methods are ill-suited for real-time applications such as consistency checking (comparing the estimated reward function to a designed reward function for real-time monitoring) or real-time learning from demonstration. The development of online IRL techniques is motivated by the need for robustness to uncertainties in the system model, the need for adaptation to changes in the system model, and the need for adaptation to changing objectives.

In this chapter, a model-based IRL approach is developed for deterministic systems in continuous time based on the preliminary results in [68], [138], and [139]. The key contribution of this chapter is the development of a novel method for reward function estimation for linear and nonlinear systems using a model-based recursive IRL technique in an online setting, using potentially uncertain agent dynamics, and input-output measurements (as opposed to input-state measurements in results such as [137, 138], and [139]). Using Lyapunov theory, the developed MBIRL technique is shown to result in ultimate boundedness of the reward function estimation error.

The chapter is organized as follows: Section 4.2 introduces the problem formulation. Section 4.3 introduces the IRL algorithm. Section 4.4 is the analysis for convergence of the developed IRL algorithm. Section 4.5 shows the simulations, and Section 4.6 concludes the chapter.

4.2 Problem Formulation

Consider an agent under observation with the dynamics

$$\begin{aligned} \dot{x} &= f(x,u), \\ y &= h(x,u), \end{aligned} \tag{74}$$

where $x \in \mathbb{R}^n$ is the state, $f : \mathbb{R}^{n \times m} \to \mathbb{R}^n$ denotes the uncertain dynamics, $u \in \mathbb{R}^m$ is the control, $y \in \mathbb{R}^l$ is the output, and $h : \mathbb{R}^{n \times m} \to \mathbb{R}^l$ denotes the measurement model. If a nominal dynamic model of the agent is available, then the dynamics in (74) can then be separated into

$$\dot{x} = f^o(x,u) + g(x,u), \tag{75}$$

where $f^o : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the nominal model, $g \in \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the uncertainty[1].

[1] If a nominal model is not available, $f^o(x,u) := 0 \; \forall \; (x,u) \in \mathbb{R}^n \times \mathbb{R}^m$.

The following assumption is required for the proposed methods.

Assumption 3 *The partial derivatives of f in* (75) *with respect to x and u are locally Lipschitz continuous.*

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

$$J(x_0, u(\cdot)) = \int_0^\infty r(x(t; x_0, u_{[0,t)}), u(t))\,\mathrm{d}t, \tag{76}$$

where $x(\cdot; x_0, u_{[0,t)})$ is the trajectory of the agent generated using the control signal $u(\cdot)$, restricted to the time interval $[0, t)$, starting from the initial condition x_0. The main objective of the paper is to estimate the unknown reward function r, in the presence of uncertain dynamics, using measurements of the input $u(\cdot)$ and the output $t \mapsto y(t) = h\left(x(t, x_0, u_{[0,t)}), u(t)\right)$, under the assumption that $u(t)$ is the optimal action in response to the state $x(t, x_0, u_{[0,t)})$.

In the following, the input and the output signals available for measurement will be denoted by $t \mapsto u(t)$ and $t \mapsto y(t)$, respectively, the corresponding unknown true state will be denoted by $t \mapsto x(t)$, and x and u will be used to denote generic elements of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

The following assumptions are used throughout the analysis.

Assumption 4 *The dynamics in* (74) *is affine in control and the optimal control problem defined by* (74), (76), *and* (77) *admits a twice continuously differentiable optimal value function.*

The class of affine systems is large, it includes linear systems and Euler Lagrange systems with invertible inertia matrices. While twice continuous differentiability of the value function is a strict requirement, many optimal control problems of interest, such as linear quadratic problems and nonlinear problems similar to those used for demonstration in Section 4.5.3, meet this requirement.

Assumption 5 *The unknown reward function r is quadratic in control, i.e.,*

$$r(x,u) = Q(x) + u^T R u, \tag{77}$$

where $R \in \mathbb{R}^{m\times m}$ is a positive definite (P.D.) matrix and $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive semi-definite (P.S.D.) continuously differentiable function with a locally Lipschitz continuous gradient.

Remark 4 *Since R can be selected to be symmetric without loss of generality, the developed IRL method only estimates the elements of R that are on and above the main diagonal.*

Assumption 6 *The state and control trajectories are bounded such that $x(t) \in \mathcal{X}$, $u(t) \in \mathcal{U}$ for some compact sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$.*

Under the premise that the observed agent makes optimal decisions, the state and control trajectories, $x(\cdot)$ and $u(\cdot)$, satisfy the Hamilton-Jacobi-Bellman equation[2] [95]

$$H\left(x\left(t\right), \nabla_x \left(V^*\left(x\left(t\right)\right)\right)^T, u\left(t\right)\right) = 0, \forall t \in \mathbb{R}_{\geq 0}, \tag{78}$$

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

The functions V^* and Q can be represented using $P \in \mathbb{N}$ and $L \in \mathbb{N}$ basis functions, respectively, as $V^*(x) = (W_V^*)^T \sigma_V(x) + \epsilon_V(x)$ and $Q(x) = (W_Q^*)^T \sigma_Q(x) + \epsilon_Q(x)$. The vectors $W_V^* := [v_1 \ldots v_P]^T \in \mathbb{R}^P$ and $W_Q^* := [q_1 \ \ldots \ q_L]^T \in \mathbb{R}^L$ denote ideal weights, $\sigma_V : \mathbb{R}^n \to \mathbb{R}^P$ and $\sigma_Q : \mathbb{R}^n \to \mathbb{R}^L$ denote continuously differentiable known features with locally Lipschitz continuous gradients, and $\epsilon_V : \mathbb{R}^n \to \mathbb{R}$ and $\epsilon_Q : \mathbb{R}^n \to \mathbb{R}$ denote approximation errors. Given any constants $\overline{\epsilon}_V, \overline{\epsilon}_Q \in \mathbb{R}_{>0}$, there exist $P, L \in \mathbb{N}$ such that ϵ_V and ϵ_Q satisfy $\sup_{x\in\chi} \left\|\epsilon_V\left(x\right)\right\| < \overline{\epsilon}_V$, $\sup_{x\in\chi} \left\|\nabla\epsilon_V\left(x\right)\right\| < \overline{\epsilon}_V$, $\sup_{x\in\chi} \left\|\epsilon_Q\left(x\right)\right\| < \overline{\epsilon}_Q$, and $\sup_{x\in\chi} \left\|\nabla\epsilon_Q\left(x\right)\right\| < \overline{\epsilon}_Q$ [58,59]. Let $\hat{V} : \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}$,

[2] For brevity, the full dependencies of the state trajectory, $x(t, x_0, u(\cdot))$, will be omitted wherever they are clear from the context and the trajectory will be denoted as $x(t)$.

$\left(x, \hat{W}_V\right) \mapsto \hat{W}_V^T \sigma_V\left(x\right)$ and $\hat{Q} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$, $\left(x, \hat{W}_Q\right) \mapsto \hat{W}_Q^T \sigma_Q(x)$ be parameterized estimates of V^* and Q, respectively, where $\hat{W}_V$ and $\hat{W}_Q$ are estimates of W_V^* and W_Q^*, respectively. Furthermore, let $u^T R u$ be parameterized as $u^T R u = \left(W_R^*\right)^T \sigma_{R1}(u)$ where $\sigma_{R1} : \mathbb{R}^m \to \mathbb{R}^M$, are the basis functions, selected as

$$\sigma_{R1}(u) := [u_1^2, 2u_1u_2, 2u_1u_3, \ldots, 2u_1u_m, u_2^2,$$
$$2u_2u_3, 2u_2u_4, \ldots, u_{m-1}^2, \ldots, 2u_{m-1}u_m, u_m^2]^T,$$

and $W_R^* \in \mathbb{R}^M$, are the ideal weights, given by

$$W_R^* = \left[R_{11}, 2R_1^{(-1)}, R_{22}, 2R_2^{(-2)}, \ldots, 2R_{m-1}^{-(m-1)}, R_{mm}\right]^T,$$

where, for a given matrix $R \in \mathbb{R}^{m \times m}$, R_{ij} denotes the corresponding element in the i-th row and the j-th column of the matrix R, and $R_i^{(-j)}$ denotes the i-th row of the matrix E with the first j elements removed, i.e., $R_3^{(-3)} := \left[R_{34}, R_{35}, \ldots, R_{3(m-1)}, R_{3m}\right]$.

Using $\hat{W}_V$ and $\hat{W}_Q$, along with estimates $\hat{W}_R$ of W_R^*, in (78), a parametric estimate of the Hamiltonian called the inverse Bellman error $\delta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+M} \to \mathbb{R}$ is obtained as

$$\delta\left(x, u, \hat{W}'\right) = \hat{W}_V^T \nabla_x \sigma_V\left(x\right) f(x, u) + \hat{W}_Q^T \sigma_Q(x) + \hat{W}_R^T \sigma_{R1}\left(u\right), \tag{79}$$

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

Since (79) utilizes the agent's dynamics, the IRL technique developed in this paper is model-based, and as such, an accurate model is required to estimate the unknown reward function. To facilitate estimation under modeling uncertainties, a system identifier is utilized that estimates the unknown model parameters.

The unknown function g in (75) can be represented using basis functions as

$$g\left(x, u, \theta\right) = \theta^T \sigma\left(x, u\right) + \epsilon\left(x, u\right), \tag{80}$$

where $\sigma \in \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and $\epsilon : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denote the basis vector and the approximation error, respectively, and $\theta \in \mathbb{R}^{p \times n}$ is a constant matrix of unknown parameters. Given any constant $\overline{\epsilon}$, there exist $p \in \mathbb{N}$ and $\overline{\sigma}, \overline{\theta} \in \mathbb{R}_{>0}$ such that $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\sigma\left(x,u\right)\right\| < \overline{\sigma}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\nabla\sigma\left(x,u\right)\right\| < \overline{\sigma}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\epsilon\left(x,u\right)\right\| < \overline{\epsilon}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})} \left\|\nabla\epsilon\left(x,u\right)\right\| < \overline{\epsilon}$, and $\|\theta\| < \overline{\theta}$.

To focus the discussion on the key contributions of the work, it is assumed that a state and parameter estimator that satisfies the following properties is available.

Assumption 7 *There exists a state and parameter estimator that yields a time instance, $\overline{T}$, such that the state and parameter estimation errors, $\tilde{x}$ and $\tilde{\theta}$, converge exponentially for all $t < \overline{T}$ and*

$$\overline{\Theta} \geq \left\|\tilde{\theta}(t)\right\|, \quad \overline{X} \geq \left\|\tilde{x}(t)\right\|, \quad \forall t \geq \overline{T}, \tag{81}$$

where $\overline{\Theta}, \overline{X} \in \mathbb{R}_{\geq 0}$ denote ultimate bounds for the parameter estimation errors and state estimation errors, respectively, $\tilde{\theta} := \theta - \hat{\theta}$ and $\tilde{x} = x - \hat{x}$, where $\hat{\theta}$ and $\hat{x}$ denote estimates of the parameters and states, respectively.

For examples of such state and parameter estimators, see [66, 67]. The state and parameter estimator is implemented synchronously with inverse reinforcement learning, and in real-time. Assumption 7 also implies existence of compact sets $\hat{\mathcal{X}} \subseteq \mathbb{R}^n$ and $\hat{\Theta} \subseteq \mathbb{R}^p$, such that $\hat{x}(t) \in \hat{\mathcal{X}}$ and $\hat{\theta}(t) \in \hat{\Theta}$, $\forall t \in \mathbb{R}_{\geq 0}$.

4.3 Inverse Reinforcement Learning Utilizing Trajectory Information

In this section, the state and parameter estimates are utilized to formulate an indirect error metric, called the approximate inverse Bellman error, that facilitates IRL.

Utilizing $\hat{x}$ and $\hat{\theta}$ from Assumption 7, and the parametric dynamics from (80), the inverse Bellman error from (79) can be approximated as

$$\hat{\delta}\left(\hat{x}, u, \hat{W}', \hat{\theta}\right) = \hat{W}_V^T \nabla_x \sigma_V\left(\hat{x}\right) \hat{Y}(\hat{x}, u, \hat{\theta}) + \hat{W}_Q^T \sigma_Q(\hat{x}) + \hat{W}_R^T \sigma_{R1}\left(u\right), \tag{82}$$

where $\hat{Y}(\hat{x}, u, \hat{\theta}) = \left[f^o(\hat{x}, u) + \hat{\theta}^T \sigma(\hat{x}, u)\right]$. Rearranging, we get

$$\hat{\delta}\left(\hat{x}, u, \hat{W}', \hat{\theta}\right) = \left(\hat{W}'\right)^T \sigma'\left(\hat{x}, u, \hat{\theta}\right), \tag{83}$$

where

$$\sigma'\left(\hat{x},u,\hat{\theta}\right):=\left[\left(\nabla_x\sigma_V\left(\hat{x}\right)\hat{Y}(\hat{x},u,\hat{\theta})\right)^T,\left(\sigma_Q\left(\hat{x}\right)\right)^T,\left(\sigma_{R1}\left(u\right)\right)^T\right]^T.$$

If $x(\cdot)$ and $u(\cdot)$ are optimal with respect to the reward function in (77), and $\tilde{x}$ and $\tilde{\theta}$ are equal to zero, then the inverse Bellman error is equal to zero whenever $\hat{W}'=\left[\left(W_V^*\right)^T,\left(W_Q^*\right)^T,\left(W_R^*\right)^T\right]^T$. Therefore, the inverse Bellman error is an indirect metric that helps gauge the quality of a given set of weight estimates.

The IRL problem can now be solved by minimizing $\|\hat{\delta}\|$. It can be seen that $\hat{W}'=0$ trivially minimizes $\|\hat{\delta}\|$. Existence of the trivial solution is expected because minimization of any positive constant multiple of a reward function generates identical optimal trajectories, and as such, the IRL problem can only be solved up to a scaling factor. As a result, there is no loss of generality in arbitrarily assigning a value to one of the reward function weights.

Taking the first element, R_{11}, of $\hat{W}_R$ to be known, the approximate inverse Bellman error in (83) can be expressed as

$$\hat{\delta}'\left(\hat{x},u,\hat{W},\hat{\theta}\right)=\hat{W}^T\sigma''\left(\hat{x},u,\hat{\theta}\right)+R_{11}u_1^2, \tag{84}$$

where $\hat{W}:=\left[\hat{W}_V^T,\hat{W}_Q^T,\left(\hat{W}_R^{(-1)}\right)^T\right]$, and

$$\sigma''\left(\hat{x},u,\hat{\theta}\right):=\left[\left(\nabla_x\sigma_V\left(\hat{x}\right)\hat{Y}(\hat{x},u,\hat{\theta})\right)^T,\left(\sigma_Q(\hat{x})\right)^T,\left(\sigma_{R1}^{(-1)}\left(u\right)\right)^T\right]^T.$$

To estimate the unknown weights using the approximate inverse Bellman error, one could update the weight estimates using

$$\dot{\hat{W}}=-K\sigma''\left(\hat{x},u,\hat{\theta}\right)\left(\left(\sigma''\left(\hat{x},u,\hat{\theta}\right)\right)^T\hat{W}+R_{11}u_1^2\right), \tag{85}$$

where K is a gain matrix. The dynamics of the state estimation error can then be expressed as a perturbed linear time-varying system with $\sigma''\left(\hat{x},u,\hat{\theta}\right)\left(\sigma''\left(\hat{x},u,\hat{\theta}\right)\right)^T$ as the system matrix that requires persistency of excitation for boundedness and convergence of the estimation error [51, 60, 112, 134]. The features of the inverse Bellman error are nonlinear, and as such, ensuring persistency of excitation a priori and monitoring PE online are generally difficult.

To relax the PE requirement and help ensure boundedness of the weight estimation errors under loss of excitation, the IRL method developed in this paper borrows the idea of history stacks from concurrent learning (CL) adaptive control [40, 44, 79]. A history stack at time t, denoted by $\mathcal{H}^{IRL}(t)$, is a collection of values of $\hat{x}(\cdot)$ and $u(\cdot)$, measured at judiciously selected time instances $t_1(t) < t_2(t) < \ldots < t_N(t) \leq t$.

The approximate inverse Bellman errors, evaluated along the trajectories $\hat{x}(\cdot)$ and $u(\cdot)$ at time instances $t_1(t), t_2(t), \ldots, t_N(t)$, can be compiled in the matrix form

$$\Delta'(t, \hat{W}) = \hat{\Sigma}'(t)\hat{W} + R_{11}\left[u_1^2(t_1(t)), \ldots, u_1^2(t_N(t))\right]^T, \tag{86}$$

where

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

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

In addition to the approximate inverse Bellman error, further information is gained by leveraging the fact that if u is the optimal action in response to state x, then[3]

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

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

$$\begin{aligned}-2Ru = &\Big(\nabla_u f^o(x) + \theta^T \nabla_u \sigma(x)\Big)^T (\nabla_x \sigma_V(x))^T W_V^* \\ &+ \Big(\nabla_u f^o(x) + \theta^T \nabla_u \sigma(x)\Big)^T \left(\nabla_x \epsilon_V(x)\right)^T \\ &+ \left(\nabla_u \epsilon(x)\right)^T \left(\left(\nabla_x \sigma_V(x)\right)^T W_V^* + \left(\nabla_x \epsilon_V(x)\right)^T\right). \qquad (87)\end{aligned}$$

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

$$\sigma_{R2}(u) = \begin{bmatrix} u^T & 0_{1\times m-1} & \dots & 0 \\ 0_{1\times m} & \left(u^{(-1)}\right)^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1\times m} & 0_{1\times m-1} & \dots & \left(u^{-(m-1)}\right)^T \end{bmatrix}. \qquad (88)$$

Using the estimates $\hat{W}_R$ and $\hat{W}_V$ in (87) for W_R^* and W_V^*, respectively, a control residual error $\Delta_u' : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+M} \to \mathbb{R}^m$ is obtained as

$$\Delta_u'(x, u, \hat{W}') = \Big(\nabla_u f^o(x) + \theta^T \nabla_u \sigma(x)\Big)^T \left(\nabla_x \sigma_V(x)\right)^T \hat{W}_V + 2\sigma_{R2}(u)\hat{W}_R. \qquad (89)$$

Utilizing the estimates $\hat{x}$ and $\hat{\theta}$ in (89), subtracting

$$0 = H(x(t_i(t)), \left(\nabla_x, V(x(t_i(t)))\right)^T, u(t_i(t))),$$

and appending (89) evaluated at $t_1(t), \dots, t_N(t)$ to (86), with the known weight R_{11} removed, results in the linear system of equations

$$-\Sigma_{R1}(t) - \hat{\Sigma}(t)\hat{W} = \hat{\Sigma}(t)\tilde{W} + \Delta(t), \qquad (90)$$

where the weight estimation error is defined as $\tilde{W} = W^* - \hat{W}$ with $W^* := \Big[\left(W_V^*\right)^T, \Big(W_Q^*\Big)^T, \Big(\left(W_R^*\right)^{(-1)}\Big)^T\Big]^T$,

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

$$\Sigma_{R1}(t) := \Big[R_{11}\left(u_1(t_1(t))\right)^2, 2R_{11}u_1(t_1(t)), 0_{1\times(m-1)}, \\ \cdots, R_{11}\left(u_1(t_N(t))\right)^2, 2R_{11}u_1(t_N(t)), 0_{1\times(m-1)}\Big]^T,$$

$$\left(\sigma'''\left(\hat{x}\left(t_i(t)\right), u\left(t_i(t)\right), \hat{\theta}\left(t_i(t)\right)\right)\right)^T := \\ \begin{bmatrix} \left(\sigma''\left(\hat{x}\left(t_i(t)\right), u\left(t_i(t)\right), \hat{\theta}\left(t_i(t)\right)\right)\right)^T \\ \begin{bmatrix} G\left(\hat{x}\left(t_i(t)\right), \hat{\theta}\left(t_i(t)\right)\right) & 0_{m\times L} & 2\sigma_{R2}^{(-1)}\left(u(t_i(t))\right)\end{bmatrix}\end{bmatrix},$$

$$G\left(\hat{x}\left(t_i(t)\right), \hat{\theta}\left(t_i(t)\right)\right) := \\ \left(\nabla_u f^o(\hat{x}(t_i(t)))\right)^T \left(\nabla_x \sigma_V(\hat{x}(t_i(t)))\right)^T \\ + \left(\left(\hat{\theta}(t_i(t))\right)^T \nabla_u \sigma(\hat{x}(t_i(t)))\right)^T \left(\nabla_x \sigma_V(\hat{x}(t_i(t)))\right)^T,$$

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

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

$$\|\Delta(t)\| \leq \overline{\Delta}_\epsilon + \bar{\tilde{x}}(t)\overline{\Delta}_{\tilde{x}} + \bar{\tilde{\theta}}(t)\overline{\Delta}_{\tilde{\theta}}, \tag{91}$$

where $\bar{\tilde{x}}(t) := \max_{i=1,2,\ldots,N} \|\tilde{x}(t_i(t))\|$ and $\bar{\tilde{\theta}}(t) := \max_{i=1,2,\ldots,N} \|\tilde{\theta}(t_i(t))\|$. Since $t \mapsto x(t), t \mapsto \hat{x}(t), t \mapsto \hat{\theta}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 7, the bounds $\bar{\Delta}_\epsilon, \bar{\Delta}_{\tilde{x}}$, and $\overline{\Delta}_{\tilde{\theta}}$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{x}$ currently stored in the history stack.

The relationship in (90) suggests the following update law for estimation of the unknown reward function weights

$$\dot{\hat{W}} = \alpha\Gamma(t)\hat{\Sigma}^T(t)\left(-\hat{\Sigma}(t)\hat{W} - \Sigma_{R1}(t)\right), \tag{92}$$

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

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\hat{\Sigma}^T(t)\hat{\Sigma}(t)\Gamma, \tag{93}$$

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

The update law in (92) is motivated by the fact that the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}} = -\alpha\Gamma(t)\hat{\Sigma}^T(t)\left(\hat{\Sigma}(t)\tilde{W} + \Delta(t)\right), \tag{94}$$

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

Analyzing (94), it can be seen that the rate of decay for the weight estimation errors is proportional to the minimum singular value of the matrix $\hat{\Sigma}^T(t)\hat{\Sigma}(t)$. In order to promote faster convergence, a minimum singular value maximization algorithm (similar to Fig. 1 in [67]) is utilized to select the time instances $t_1(t), \ldots, t_N(t)$. More specifically, a new data point $(x(t), u(t))$ replaces an existing data point $(x(t_i(t)), u(t_i(t)))$, for some $i \in \{1, \ldots, N\}$, if the replacement results in the largest increase in the minimum singular value of $\hat{\Sigma}^T(t)\hat{\Sigma}(t)$ among all N possible replacements.

Since the size of the perturbation depends on the quality of the state and parameter estimates in the history stack $\mathcal{H}^{IRL}(t)$, a time-based purging algorithm is utilized to purge poor estimates $\hat{x}$ and $\hat{\theta}$ from the history stack. Since the state and parameter estimation errors decay exponentially to a bound, newer estimates of $\hat{x}$ and $\hat{\theta}$ are assumed to be better. Therefore, a time interval, $\tau \in \mathbb{R}_{>0}$, is selected so that a new purging event occurs only after τ seconds have passed since the previous purge.

The developed purging technique maintains an additional transient history stack, labeled $\mathcal{G}^{IRL}$ populated using minimum singular value maximization. As soon as the transient history stack is full rank according to (95) and τ seconds have passed, $\mathcal{H}^{IRL}$ is emptied and $\mathcal{G}^{IRL}$ is copied into $\mathcal{H}^{IRL}$. The history stack $\mathcal{H}^{IRL}$ is kept constant in between purging instances.

Due to purging, the time instances $\{t_1, \cdots t_N\}$, the matrices $\hat{\Sigma}$ and Σ_{R1}, and consequently $\mathcal{H}^{IRL}$, are piecewise constant in time.

4.4 Analysis of the Developed MBIRL Technique

Convergence of the estimation error to a neighborhood of the origin follow under the following condition on the history stack.

Definition 2 *The history stack $\mathcal{H}^{IRL}$, is called full rank, uniformly in t, if there exists $\underline{\sigma} \in \mathbb{R}_{>0}$ such that*[4] $\forall t \in \mathbb{R}_{\geq 0}$,

$$\underline{\sigma} < \lambda_{\min}\left\{\left(\hat{\Sigma}(t)\right)^T \hat{\Sigma}(t)\right\}. \tag{95}$$

Definition 3 *The signal $(\hat{x}, u)$ is called* finitely informative *(FI) if there exist time instances $0 \leq t_1 < t_2 < \cdots < t_N$ such that the resulting history stack is full rank and* persistently informative *(PI) if for any $T \geq 0$, there exist time instances $T \leq t_1 < t_2 < \cdots < t_N$ such that the resulting history stack is full rank.*

The stability result is summarized in the following theorem.

Theorem 3 *If the unknown states and parameters are estimated using a state and parameter estimator that satisfies Assumption 7, the signal $(\hat{x}, u)$ is FI, the time instances $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that $\mathcal{H}^{IRL}$ is full rank, uniformly in t, and $\mathcal{H}^{IRL}$ is refreshed using a time-based purging algorithm, then $t \mapsto \tilde{W}(t)$ is ultimately bounded (UB).*

Proof. Consider the candidate Lyapunov function

$$V(\tilde{W}, t) = \frac{1}{2}\tilde{W}^T \Gamma^{-1}(t)\, \tilde{W}. \tag{96}$$

Using arguments similar to [60, Corollary 4.3.2], it can be shown that provided $\lambda_{\min}\left\{\Gamma^{-1}(0)\right\} > 0$, the least squares gain matrix satisfies

$$\underline{\Gamma}\mathrm{I}_{L+P+m-1} \leq \Gamma(t) \leq \overline{\Gamma}\mathrm{I}_{L+P+m-1}, \forall\, t \geq 0, \tag{97}$$

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

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

Using the bounds in (97), the candidate Lyapunov function satisfies

$$\frac{1}{2\overline{\Gamma}}\left\|\tilde{W}\right\|^2 \leq V\left(\tilde{W},t\right) \leq \frac{1}{2\underline{\Gamma}}\left\|\tilde{W}\right\|^2. \tag{98}$$

Using (93), (94), (95), and (97), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, and the Cauchy-Schwartz inequality, the time-derivative $\dot{V}$ can be bounded by

$$\dot{V}(\tilde{W},t) \leq -\frac{1}{2}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta\right)\left\|\tilde{W}\right\|^2 + \alpha\left\|\tilde{W}\right\|\left\|\hat{\Sigma}(t)\right\|\left\|\Delta(t)\right\|. \tag{99}$$

Using (91), $\dot{V}$ can be bounded as

$$\dot{V}(\tilde{W},t) \leq -\frac{1}{4}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta\right)\left\|\tilde{W}\right\|^2, \ \forall\|\tilde{W}\| \geq \rho\left(\|\mu\|\right), \tag{100}$$

where $\mu = \left[\sqrt{\overline{\Delta}_\epsilon}, \sqrt{\overline{\tilde{x}}}, \sqrt{\overline{\tilde{\theta}}}\right]^T$, $\rho(\|\mu\|) = \left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+{}^{\beta}/_{\overline{\Gamma}}}\right)\|\mu\|^2$, and $\overline{\Sigma}$ satisfies $\left\|\hat{\Sigma}(t)\right\| \leq \overline{\Sigma}$, $\forall t \geq 0$. Since $t \mapsto x(t), t \mapsto \hat{x}(t), t \mapsto \hat{\theta}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 7, the bound $\overline{\Sigma}$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{x}$ currently stored in the history stack. Using (98) and (100), [80, Theorem 4.19] can be invoked to conclude that (94) is input-to-state stable (ISS) with state $\tilde{W}$ and input μ.

If a time-based purging algorithm is implemented and if the signal $(\hat{x}, u)$ is FI, there exists a time instance T_s, such that for all $t \geq T_s$, the history stack $\mathcal{H}^{IRL}(t)$ remains unchanged. As a result, using Exercise 4.58 from [80], it can be concluded that the ultimate bound on $\tilde{W}$ can be expressed as

$$\begin{aligned}\limsup_{t\to\infty}\|\tilde{W}(t)\| \leq &\sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+{}^{\beta}/_{\overline{\Gamma}}}\right)\overline{\Delta}_\epsilon \\ &+ \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+{}^{\beta}/_{\overline{\Gamma}}}\right)\left(\overline{\tilde{x}}(T_s) + \overline{\tilde{\theta}}(T_s)\right),\end{aligned} \tag{101}$$

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

Furthermore, if $(\hat{x}, u)$ is PI, then $\limsup_{t\to\infty} \overline{\tilde{x}}(t) \to \overline{X}$ and $\limsup_{t\to\infty} \overline{\tilde{\theta}}(t) \to \overline{\Theta}$. In that case, (101) reduces to

$$\begin{aligned}\limsup_{t\to\infty}\|\tilde{W}(t)\| \leq & \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+{}^{\beta}/{}_{\overline{\Gamma}}}\right)\overline{\Delta}_\epsilon \\ & + \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_x,\overline{\Delta}_\theta\}}{\alpha\underline{\sigma}+{}^{\beta}/{}_{\overline{\Gamma}}}\right)\left(\overline{X}+\overline{\Theta}\right). \quad (102)\end{aligned}$$

■

The ultimate bound for the estimation error, $\tilde{W}$, has a direct relationship to the approximation errors for both the reward function and the value function, along with the ultimate bounds for the state and parameter estimates. As such, the ultimate bound can be reduced by reducing those errors. This observation motivates the following corollary.

Corollary 1 *If $\overline{\Theta}$, $\overline{X}$, ϵ_Q, and ϵ_V are zero, the signal $(\hat{x}, u)$ is PI, the time instance $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that $\mathcal{H}^{IRL}$ is full rank, uniformly in t, and $\mathcal{H}^{IRL}$ is refreshed using a time-based purging algorithm, then as $t \to \infty$, $\|\tilde{W}(t)\| \to 0$.*

Proof. Immediate from Theorem 3. ■

Remark 5 *If the full state is measurable, the restrictions on the dynamics for the agent and the basis functions can be relaxed to continuous differentiability.*

4.5 Simulation

This section presents simulations for the IRL method developed in this chapter. The first simulation demonstrates the IRL method detailed in Section 4.3 for output-feedback linear systems. The second simulation demonstrates the same output-feedback linear system, however, the reward function changes halfway through the simulation to show the IRL method can adapt to this change. The last simulation shows an output-feedback nonlinear optimal control problem that is selected with a known value function. The simultaneous state and parameter estimator developed in Chapter III is used to satisfy the conditions of Assumption 7.

4.5.1 Output-Feedback IRL for Linear Systems

To verify the performance of the developed method, a linear quadratic optimal control problem is selected since it has a known optimal value function for comparison. The linear system is

$$\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & 1 \\ 5 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} u.$$

The weighing matrices in the reward function are selected as $Q = \text{diag}\left([1,\ 2,\ 3,\ 6]\right)$ and $R = [20,\ 10]$, where $R(1,1)$ is assumed to be known. The observed input-output trajectories, along with a prerecorded history stack are used to implement the simultaneous state and parameter estimation algorithm in Chapter III. The design parameters in the system identification algorithm are selected using trial and error as $M = 150$, $T_1 = 1s$, $T_2 = 0.8s$, $k = 100$, $\alpha = 20$, $\beta = 10$, $\beta_1 = 5$, $k_\theta = 0.3/M$, and $\Gamma(0) = 0.1 * \text{I}_{L+P+m-1}$.

The behavior of the system under the optimal controller is observed, and at each time step, a random state vector x^* is selected and the optimal action u corresponding to the random state vector is queried from the entity under observation. The queried state-action pairs (x^*, u) are utilized in conjunction with the estimated state-action pairs $\left(\hat{x}(t), u(t)\right)$ to implement the IRL algorithm developed.

Figs. 7 and 8 demonstrate the performance of the developed state estimator and Fig. 9 illustrates the performance of the developed parameter estimator. Fig. 10

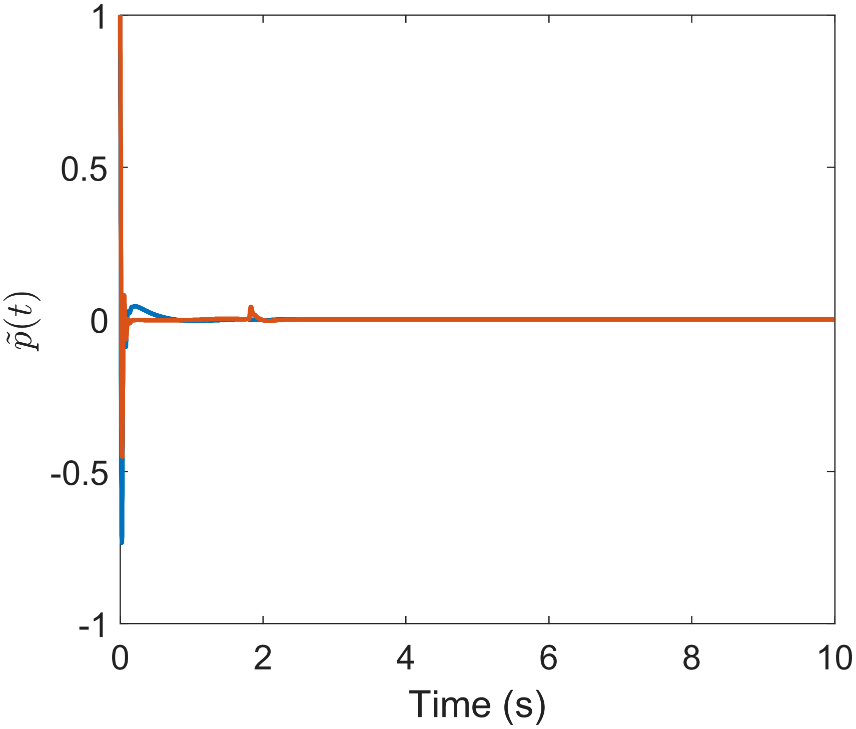


Figure 7: Generalized position estimation error.

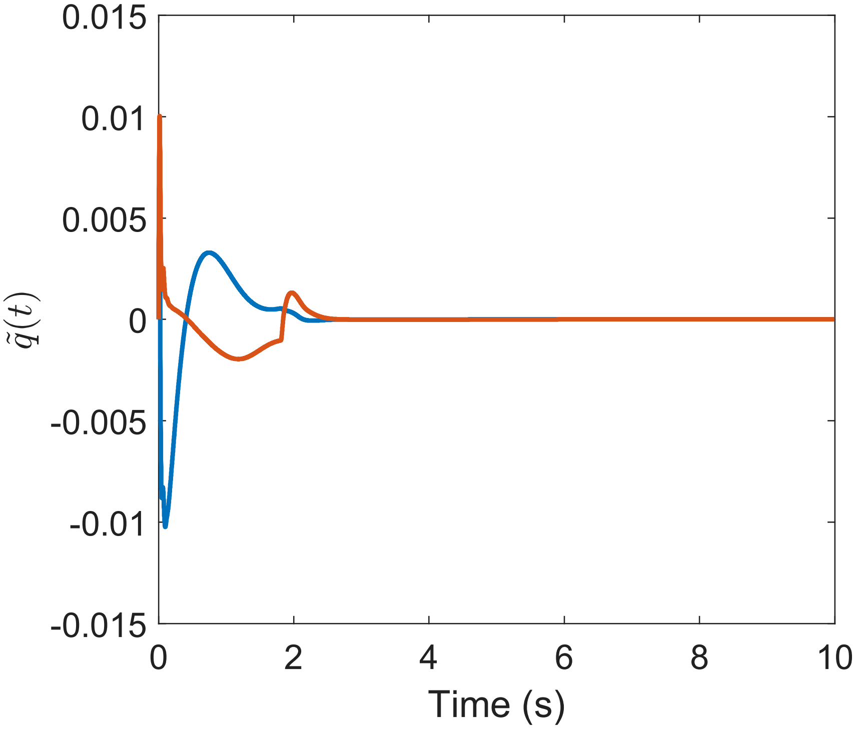


Figure 8: Generalized velocity estimation error.

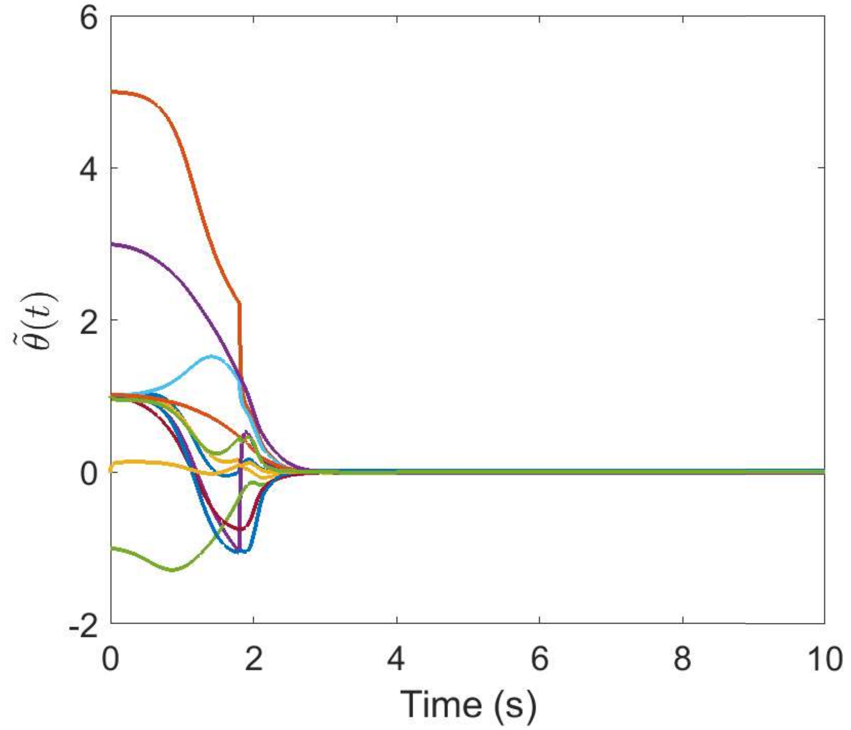


Figure 9: Estimation error for the unknown parameters in the system dynamics.

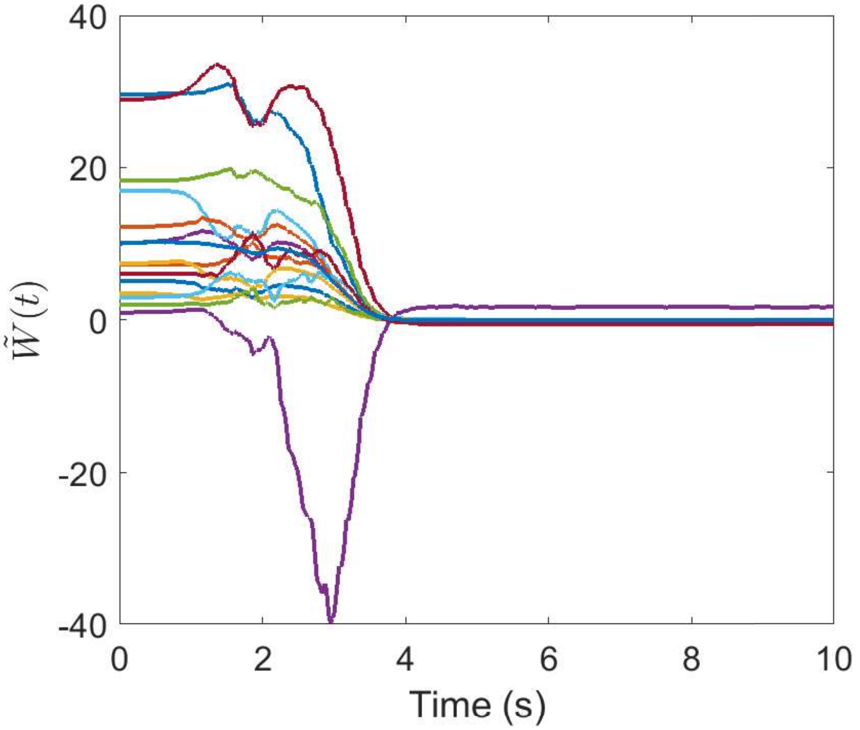


Figure 10: Estimation error for the unknown parameters in the reward function.

indicates that the developed IRL technique can be successfully utilized to estimate the reward function of an entity under observation within a bound.

4.5.2 Output-Feedback IRL for Linear Systems with a Change in the Reward Function

To further validate the performance of the developed method, a linear quadratic optimal control problem is selected and the reward function is chosen to change at 10 seconds. The linear system is

$$\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & 1 \\ 5 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} u.$$

The weighing matrices in the reward function for $t < 10$ seconds are selected as $Q = \operatorname{diag}\left([1,\ 2,\ 3,\ 6]\right)$ and $R = [20,\ 10]$, and the weighing matrices in the reward function for $t \geq 10$ seconds are selected as $Q = \operatorname{diag}\left([3,\ 4,\ 2,\ 10]\right)$ and $R = [2,\ 8]$, where $R\left(1,1\right)$ is assumed to be known throughout the simulation. The observed input-output trajectories, along with a prerecorded history stack are used to implement the simultaneous state and parameter estimation algorithm in Chapter III. The design parameters in the system identification algorithm are selected using trial and error as $M = 150$, $T_1 = 1s$, $T_2 = 0.8s$, $k = 100$, $\alpha = 20$, $\beta = 10$, $\beta_1 = 5$, $k_\theta = 0.3/M$, and $\Gamma\left(0\right) = 0.1 * \mathrm{I}_{L+P+m-1}$.

Figs. 11 and 12 demonstrate the performance of the developed state estimator and Fig. 13 illustrates the performance of the developed parameter estimator.

Fig. 14 indicates even with a change in the reward function weights in real-time, the IRL method developed can estimate the unknown weights within a bound of the origin.

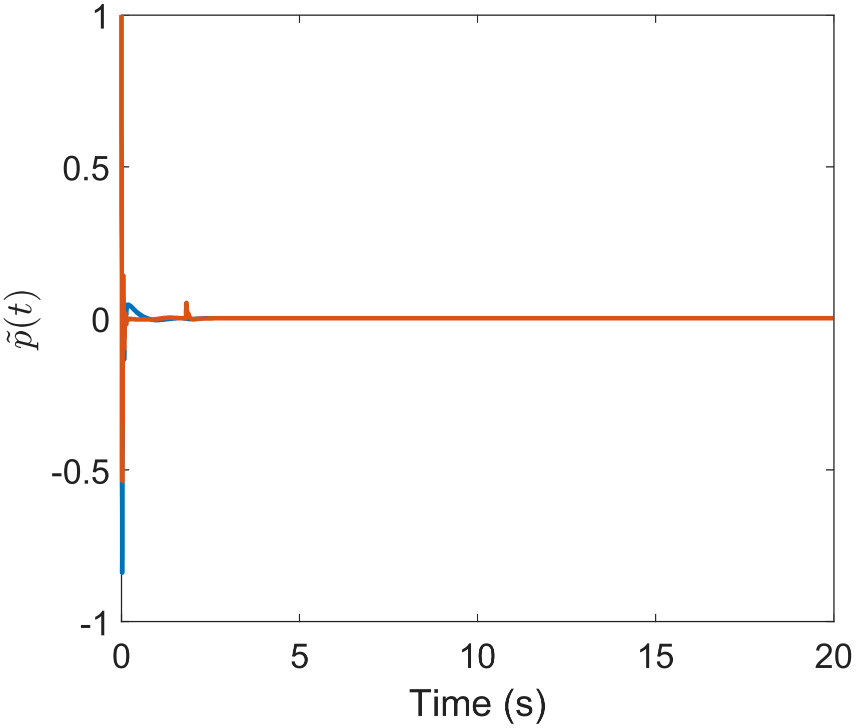


Figure 11: Generalized position estimation error.

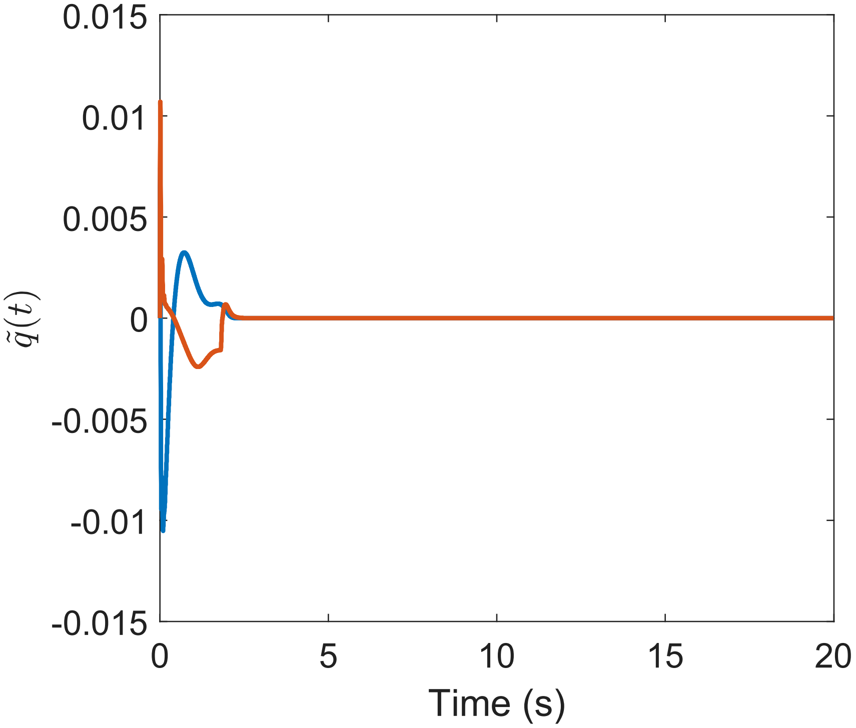


Figure 12: Generalized velocity estimation error.

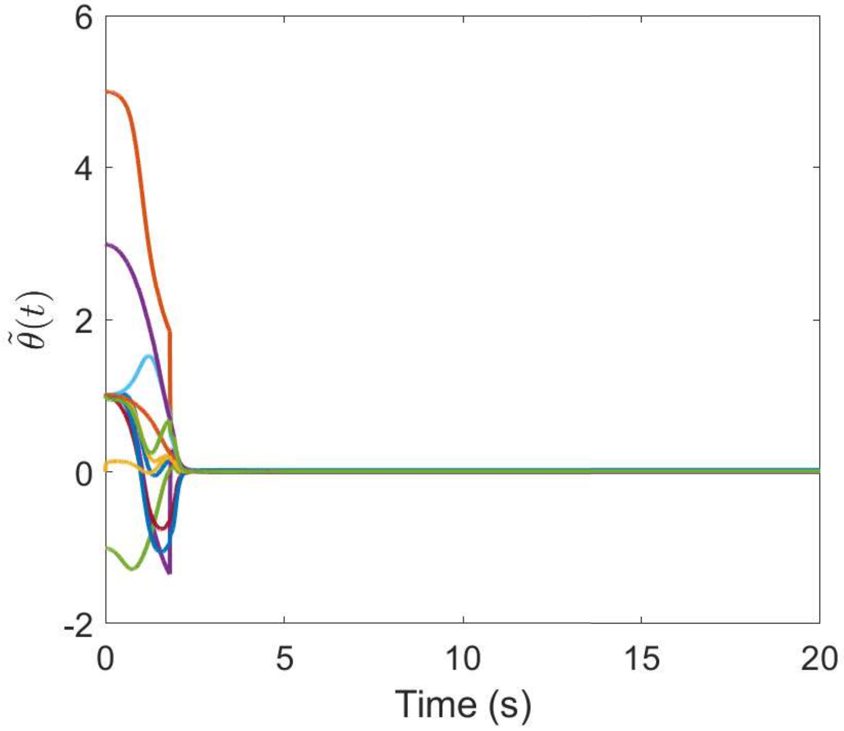


Figure 13: Estimation error for the unknown parameters in the system dynamics.

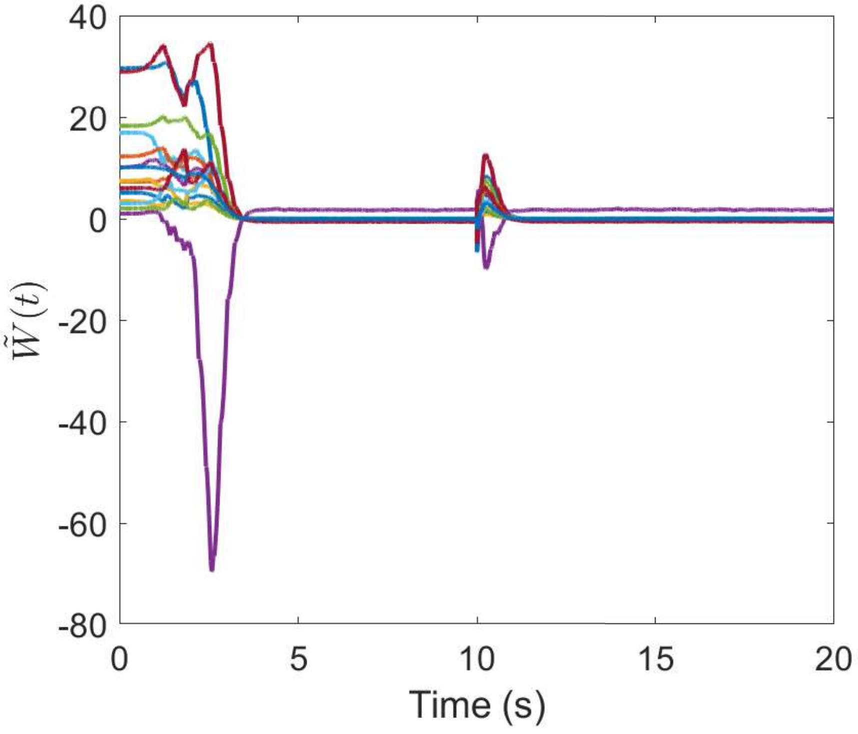


Figure 14: Estimation error for the unknown parameters in the reward function.

4.5.3 Output-Feedback IRL for Nonlinear Systems

The agent under observation has the following nonlinear dynamics

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= \theta_1 x_1\Big(\frac{\pi}{2} + \tan^{-1}(5x_1)\Big) + \frac{\theta_2 x_1^2}{1+25x_1^2} + \theta_3 x_2 + 3u,\end{aligned} \tag{103}$$

where the parameters θ_1, θ_2, and θ_3 are unknown constants to be estimated. The exact values of these parameters are $\theta_1 = -1, \theta_2 = -\frac{5}{2}$, and $\theta_3 = 4$.

The agent is trying to minimize the cost function in (76) with $r(x,u) = x_2^2 + u^2$, resulting in the reward function weights $Q = \operatorname{diag}(\begin{bmatrix} W_{Q_1}, & W_{Q_2}\end{bmatrix}) = \operatorname{diag}([0,1])$ and $R = 1$. The observed output and control trajectories are used in the estimation of unknown parameters in the dynamics, the system state, the optimal value function parameters and the reward function weights.

The closed form optimal controller is

$$u^* = -\frac{1}{2}R^{-1}\left(\nabla_u f(x)\right)^T \left(\nabla_x V(x)\right)^T = -3x_2,$$

with the corresponding optimal value function

$$V^* = x_1^2\left(W_{V_1} + W_{V_2}\tan^{-1}(5x_1)\right) + W_{V_3}x_2^2,$$

resulting in the ideal value function weights $W_{V_1} = \frac{\pi}{2}$, $W_{V_2} = 1$, and $W_{V_3} = 1$. It is assumed that the controller that the agent under observation is utilizing is a combination of the optimal controller and a known exciting controller, that is,

$$u(t) = -3x_2(t) + 9\cos(3t) + 6\cos(2t) + 3\cos(t) + 15\cos(5t).$$

The history stack $\mathcal{H}^{IRL}$ is initialized so that all the elements in the history stack are zero[5]. Data is added to the history stack using a minimum singular value maximization algorithm. A time-based purging technique is utilized with $\tau = 1$. The

[5] It is clear from the simulation results that full rank initialization of the history stack is a sufficient, but not necessary condition for the analysis in Section 4.4.

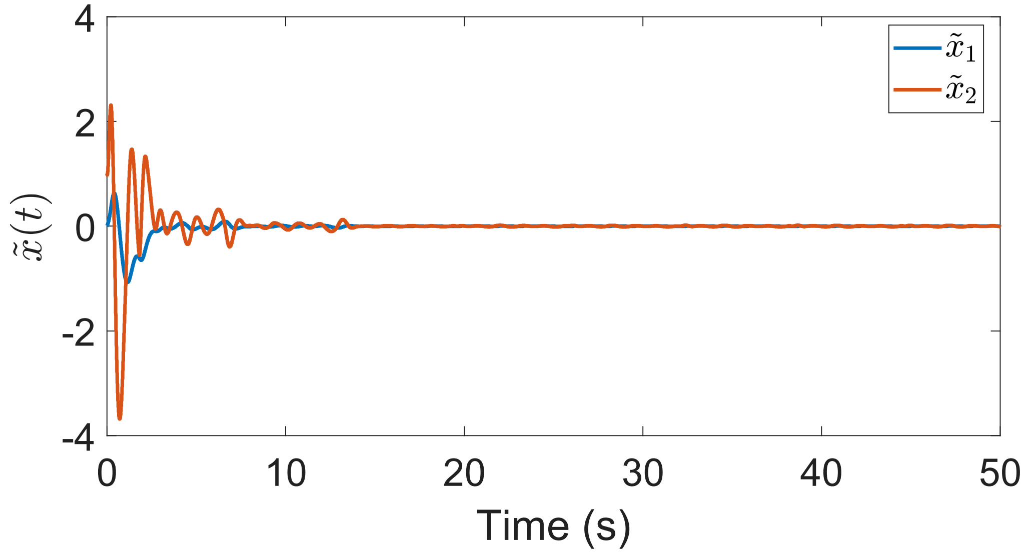


Figure 15: State estimation errors for the system in (103).

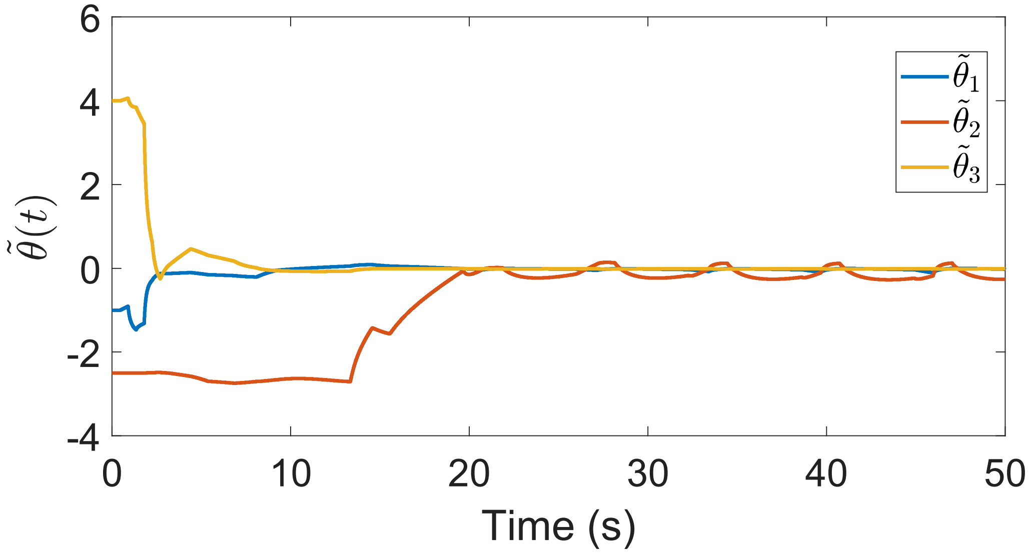


Figure 16: Parameter estimation errors for the uncertain dynamics in (103).

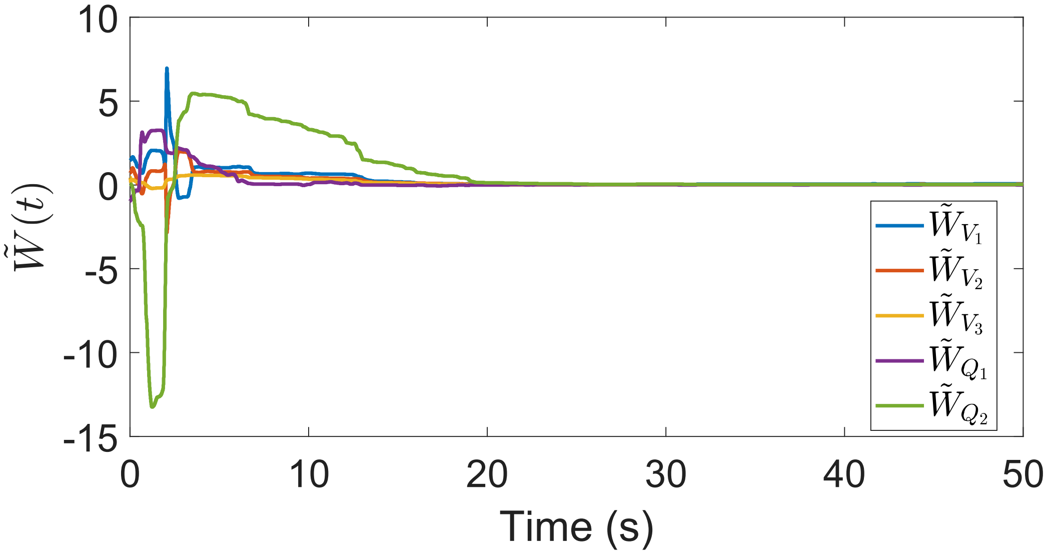


Figure 17: Reward and value function weight estimation errors using direct MBIRL in Section 4.3 for the optimal control problem in (76) with $r(x, u) = x_2^2 + u^2$.

parameters used for the simulation are: $\alpha = 0.0033$, $N = 100$, $\beta = 0.5$, and the simulation time step size is set to $T_s = 0.003$s.

Figs. 15 - 17 show the performance of the developed MBIRL method. As seen in Figs. 15 and 16, the uncertain parameters and system state estimates converge to a small bound near the origin. As seen in Fig. 17, the MBIRL approach is able to estimate the ideal values of the reward and value functions online even with a non-zero ultimate bound on the state and parameter estimates.

4.6 Conclusion

In this chapter, an online model-based IRL method is developed that facilitate reward function estimation utilizing a single demonstration. Theoretical guarantees using Lyapunov theory are established to show error convergence to a bound. Further chapters will build off the foundation built in this chapter.

Chapter V

INVERSE REINFORCEMENT LEARNING WITH INCONSISTENT OBSERVATIONS

The method illustrated so far utilize the assumption that the agent under observation is acting optimally with respect to an unknown reward function. In general, the requirement of optimal demonstrations is a strong assumption since agents in real environments are often affected by unknown disturbances. These external disturbances result in the agent's demonstrations being suboptimal, and these suboptimal demonstrations make model-free IRL methods challenging because model-free IRL, in general, require either optimal or near optimal demonstrations. Model-based IRL methods can be used to compensate for the disturbance-induced sub-optimality if a dynamic model of the agent under observation can be learned. However, the disturbances make system identification challenging, and the resulting models are typically poor. Therefore, this chapter focuses on IRL techniques in real-time that can handle suboptimal demonstrations from the agent under observation due to external disturbances affecting the agent's otherwise optimal performance.

5.1 Introduction

IRL [3,5,92,93,111,116,118,126,132,143,157,159] and inverse optimal control (IOC) methods [65] are extensively utilized to teach autonomous machines to perform specific tasks in an *offline* setting. However, the offline approaches are, in general, prohibitively computationally intensive for real-time implementation, or require more data than is typically available in applications that require real-time learning. Inspired by the success of model-based real-time reinforcement learning methods in, e.g., [149] and [151], and the online IRL/IOC results for linear [68], [110] and nonlinear systems [138], this paper presents an online IRL technique for systems where the observed trajectories are inconsistent with its internal reward function due to external disturbances.

Model-free IRL methods, in general, are entirely trajectory driven, and require the observed trajectories to be either optimal/near-optimal or require sub-optimal trajectories to be rare occurrences [124,159]. However, if the agent under observation is experiencing external disturbances, then the observed trajectories are unlikely to be optimal, which makes model-free IRL difficult. Even if the unknown disturbances can be estimated, removing the effects of these disturbances from the observed trajectories is nontrivial in a model-free IRL setting.

Recently, there has been some work done to develop IRL techniques to help alleviate the difficulties in reward function estimation with sub-optimal trajectories. In [27], the authors consider using preference ranked demonstrations in order to uncover the reward functions that extrapolate beyond the set of ranked demonstrations, [38] considers the problem of utilizing unlabeled demonstrations for different demonstrators with potentially varying skill levels, and [156] develops an approach aimed at separating out noisy or sub-optimal demonstrations in order to extract ideal reward functions. In [53], the authors research if helpful information can be extracted from potentially failed, or severally sub-optimal attempts, instead of disregarding them. The aforementioned methods attempt to extract reward functions with inconsistent observations through a variety of different approaches. However, these methods require multiple trajectories, are computationally expensive, and as such, are not suitable for online implementation.

The novelty of the technique developed in this chapter is the use of a model to compensate for disturbance-induced sub-optimality. Addressing the complexity resulting from sub-optimality of the demonstrations is a major technical contribution of this chapter.

This chapter builds on and extends the preliminary work in [137], where the disturbances affecting both agents, learner and demonstrator, are assumed to be equal. This strong assumption facilitates the analysis in [137], which utilizes the fact that disturbance estimation error is exponentially convergent to zero. Instead, in this chapter, the disturbances of the two agents are allowed to be different, which results in ultimately bounded disturbance estimates.

In order to implement model-based IRL, if a dynamic model of the demonstrator is unavailable, it needs to be identified from the data. However, the disturbances make system identification challenging, and the resulting models are typically poor. To overcome this challenge, it is assumed that the learner and demonstrator are co-located and, as a result, experience similar disturbances, such as teams of quadrotors in a constant wind-field or autonomous watercraft affected by a stream. One can then estimate the disturbance using its effects on the learner and use the resulting estimates to identify the dynamic model of the demonstrator. A model-based IRL method can then be deployed to learn the unknown reward function.

The chapter is organized as follows: Section 5.2 details the problem formulation and how the additional challenges related to disturbances are addressed. Section 5.3 details the disturbance estimator for this method. Section 5.4 shows the developed parameter estimator. Section 5.5 explains the IRL algorithm. Section 5.6 shows a simulation example for the proposed method and Section 5.7 concludes the chapter.

5.2 Problem Formulation

Consider two agents, Agent 1 and Agent 2, where Agent 1 is monitoring the behavior of Agent 2. Agent 1 has the dynamics

$$\dot{x}_1 = f_1(x_1, u_1) + d_1, \tag{104}$$

where $x_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the state, $u_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ is the control, $f_1 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a locally Lipschitz continuous function, and $d_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is a disturbance acting on Agent 1. The dynamics for Agent 2 are

$$\dot{x}_2 = f_2(x_2, u_2) + d_2, \tag{105}$$

where $x_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the state, $u_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ is the control, $f_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a locally Lipschitz continuous function, and $d_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is a disturbance acting on the Agent 2.

Assume that Agent 2 is using a controller that minimizes the performance index

$$J(x_2^o, u_2(\cdot)) = \int_0^\infty r(x_2(t; x_2^o, u_2(\cdot)), u_2(t))\, \mathrm{d}t, \tag{106}$$

where $x_2\left(\cdot; x_2^o, u_2(\cdot)\right)$ is the trajectory generated by the control signal $u_2(\cdot)$, for the undisturbed dynamics, starting at x_2^o. The objective is to estimate the unknown reward function, r, in the presence of uncertainties in the dynamics and sub-optimality of the measured trajectories due the to unknown disturbance d_2.

If Agent 1 and Agent 2 are co-located and have similar dynamic models, then the disturbances affecting them can be reasonably assumed to be similar.

Assumption 8 *The disturbances affecting both agents are similar, i.e.* $\left\| d_1(t) - d_2(t) \right\| \leq \epsilon_d(t), \forall t_{\geq 0}$ *where* $\overline{\epsilon}_d := \sup_{t \in \mathbb{R}_{\geq 0}} \{\epsilon_d(t)\} < \infty$.

The following assumptions are used throughout the analysis.

Assumption 9 *The unknown reward function r is quadratic in the control, i.e.,*

$$r(x,u) = Q(x) + u^T R u, \tag{107}$$

where $R \in \mathbb{R}^{m\times m}$ is a positive definite (P.D.) matrix, such that $R = \operatorname{diag}([r_1, \cdots, r_m])$ and $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive semi-definite (P.S.D.) function.

Assumption 10 *The state and control trajectories of Agent 2 are bounded such that $x_2(t, x_2^o, u_2(\cdot)) \in \mathcal{X}, u_2(t) \in \mathcal{U}$ for some compact sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \in \mathbb{R}^m$.*

The function Q can be represented using $L \in \mathbb{N}$ basis functions as $Q(x) = (W_Q^*)^T \sigma_Q(x) + \epsilon_Q(x)$. The vector $W_Q^* := [q_1 \ \ldots \ q_L]^T \in \mathbb{R}^L$ denotes the ideal weights, $\sigma_Q : \mathbb{R}^n \to \mathbb{R}^L$ denotes continuously differentiable known features, and $\epsilon_Q : \mathbb{R}^n \to \mathbb{R}$ denotes the approximation error. Given any constant $\overline{\epsilon}_Q \in \mathbb{R}_{>0}$ [58, 59], there exist $L \in \mathbb{N}$ such that ϵ_Q satisfies $\sup_{x\in\chi} \left\| \epsilon_Q(x) \right\| < \overline{\epsilon}_Q$, and $\sup_{x\in\chi} \left\| \nabla \epsilon_Q(x) \right\| < \overline{\epsilon}_Q$.

Assumption 11 *The dynamics of Agent 2 in* (105) *are affine in control and the optimal control problem defined by* (105)*,* (106)*, and* (107) *admits a continuously differentiable optimal value function.*

The class of affine systems is large, it includes linear systems and Euler Lagrange systems with invertible inertia matrices. Many optimal control problems of interest satisfy continuously differentiable optimal value functions, such as linear quadratic problems and nonlinear problems similar to those used for demonstration in Section 5.6.1, meet this requirement.

The dynamics for Agent 2 in (105) can be represented as

$$\dot{x}_2 = f_2^o(x_2, u_2) + \theta_2^T \sigma_2(x_2, u_2) + \epsilon_2(x_2, u_2) + d_2, \tag{108}$$

where $f_2^o : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the nominal dynamics, $\theta_2^T \sigma_2$ is a parameterized estimate of the uncertain part of the dynamics, where $\theta_2 \in \mathbb{R}^{p\times n}$ are unknown parameters and $\sigma_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ are known continuously differentiable features, and $\epsilon_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the approximation error.

Due to the unknown disturbance d_2 acting on Agent 2, in spite of the optimal feedback policy employed, the trajectories of Agent 2 will not be optimal with respect to the reward function in (106). As a result, a purely data-driven implementation of IRL would yield incorrect reward function estimates. Instead, in this paper, the reward function is estimated using a model-based approach that compensates for the trajectory deviations. The unknown disturbance, d_1, is estimated by Agent 1 using their known internal model. Agent 1 also implements a parameter estimator that uses the disturbance estimates to calculate the unknown parameters in the dynamics of Agent 2. Finally, both the disturbance and parameter estimates are used by Agent 1 to estimate the unknown reward function that Agent 2 is attempting to optimize. Disturbance estimation, parameter estimation, and inverse reinforcement learning are performed by Agent 1 synchronously and in real-time.

A block diagram showing the Agent 1 and Agent 2 architecture is shown in Fig. 18.

5.3 Disturbance Estimation

To implement the IRL method discussed in Section 5.5, Agent 1 can utilize any disturbance estimator that satisfies the following Assumption.

Assumption 12 *There exists a time instance, T_d, such that the disturbance estimation error, $\tilde{d}_1 = d_1 - \hat{d}$, where $\hat{d}$ is the estimate of the unknown disturbance, converges exponentially for all $t < T_d$ and*

$$\overline{d}_1 \geq \left\| \tilde{d}_1(t) \right\|, \quad \forall t \geq T_d, \tag{109}$$

where $\overline{d}_1 \geq 0$ is the ultimate bound.

Note that, since the difference between the disturbances acting on the two agents is assumed to be bounded, the disturbance estimation error, $\tilde{d}_2 = d_2 - \hat{d}$, for the unknown

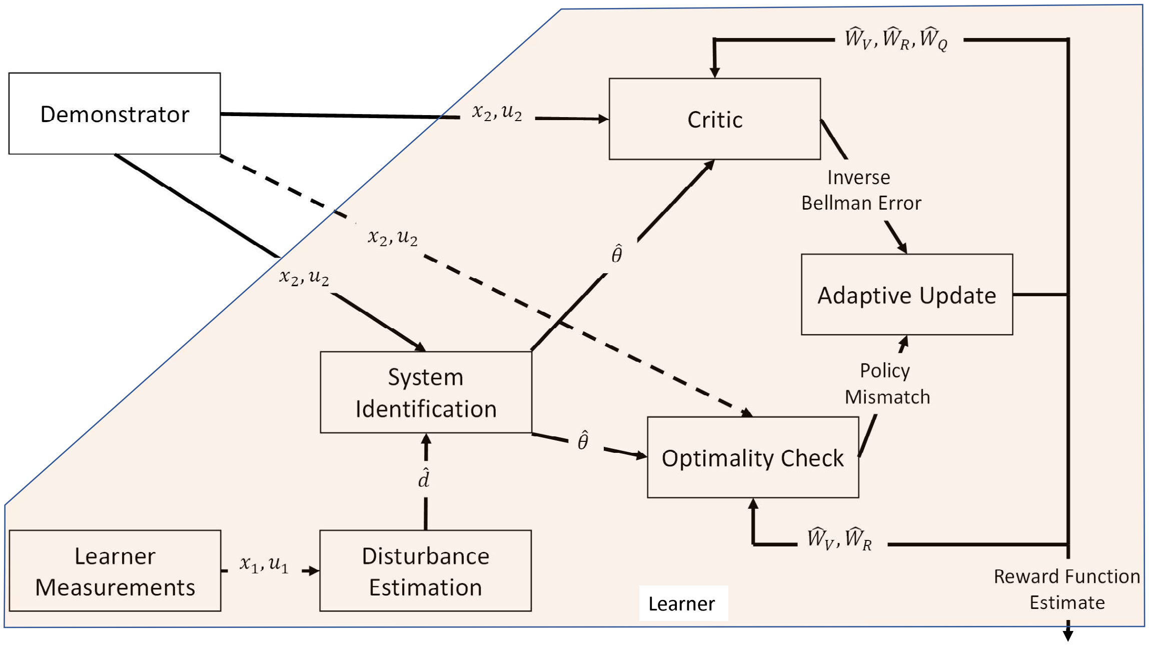


Figure 18: Learner (Agent 1) and Demonstrator (Agent 2) signal block diagram.

disturbance acting on Agent 2 is also UB. The ultimate bound of $\tilde{d}_2$, denoted as $\overline{d}_2$, is bounded from above by $\overline{d}_1 + \overline{\epsilon}_d$, i.e. $0 \leq \|\tilde{d}_2(t)\| \leq \overline{d}_2 \leq \overline{d}_1 + \overline{\epsilon}_d, \ \forall t \geq T_d$.

Examples of a disturbance estimator that satisfies Assumption 12 are available in results such as [31, 32].

5.4 Parameter Estimation

Due to disturbance estimation, the parameter estimator developed in [67] is adapted in the following to compensate for this additional disturbance term.

In Sections 5.4-5.5, the subscripts that denote the agent number in the dynamics of Agent 2 will be omitted for brevity. The specific disturbance estimation terms referring to Agent 1 will be denoted with a subscript 1.

5.4.1 Design

Integrating (108) over the interval $[t-T, t]$ for some constant $T \in \mathbb{R}_{>0}$, yields

$$x\left(t\right) - x\left(t-T\right) = \int_{t-T}^{t} f^{o}\left(x(\gamma), u(\gamma)\right) \mathrm{d}\gamma + \theta^{T} \int_{t-T}^{t} \sigma\left(x(\gamma), u(\gamma)\right) \mathrm{d}\gamma + \int_{t-T}^{t} \epsilon\left(x(\gamma), u(\gamma)\right) \mathrm{d}\gamma + \int_{t-T}^{t} d\left(\gamma\right) \mathrm{d}\gamma. \quad (110)$$

The expression in (110) can be rearranged to form the affine system

$$X\left(t\right) = F\left(t\right) + \theta^{T} S\left(t\right) + E\left(t\right) + D\left(t\right), \ \forall t \in \mathbb{R}_{\geq 0}, \quad (111)$$

where

$$X\left(t\right) := \begin{cases} x(t) - x\left(t-T\right), & t \in [T, \infty), \\ 0, & t < T, \end{cases} \quad (112)$$

$$F\left(t\right) := \begin{cases} \int_{t-T}^{t} f^{o}\left(x(\gamma), u(\gamma)\right) \mathrm{d}\gamma, & t \in [T, \infty), \\ 0, & t < T, \end{cases} \quad (113)$$

$$S\left(t\right) := \begin{cases} \int_{t-T}^{t} \sigma\left(x(\gamma), u(\gamma)\right) \mathrm{d}\gamma, & t \in [T, \infty), \\ 0, & t < T, \end{cases} \tag{114}$$

$$E\left(t\right) := \begin{cases} \int_{t-T}^{t} \epsilon\left(x(\gamma), u(\gamma)\right) \mathrm{d}\gamma, & t \in [T, \infty), \\ 0, & t < T, \end{cases} \tag{115}$$

and

$$D\left(t\right) := \begin{cases} \int_{t-T}^{t} d\left(\gamma\right) \mathrm{d}\gamma, & t \in [T, \infty), \\ 0, & t < T. \end{cases} \tag{116}$$

The affine error system in (111) motivates the adaptive estimation scheme that follows, in which a *concurrent learning*-like technique [41] is developed that utilizes recorded data stored in a history stack to drive parameter estimation.

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

$$X_i = F_i + \theta^T S_i + \hat{D}_i + \mathcal{E}_i, \ \forall i \in \{1, \cdots, M\}, \tag{117}$$

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

$$\hat{D}\left(t\right) := \begin{cases} \int_{t-T}^{t} \hat{d}\left(\gamma\right) \mathrm{d}\gamma, & t \in [T, \infty), \\ 0, & t < T. \end{cases} \tag{118}$$

Definition 4 *A history stack $\mathcal{H}^{PE}$ is called* full rank *if there exists a constant $\underline{c} \in \mathbb{R}$ such that*

$$0 < \underline{c} < \lambda_{\min}\left\{\mathscr{S}\right\}, \tag{119}$$

where the matrix $\mathscr{S} \in \mathbb{R}^{p \times p}$ is defined as $\mathscr{S} := \sum_{i=1}^{M} S_i S_i^T$.

The history stack $\mathcal{H}^{PE}$, if time-varying, is called full-rank, uniformly in t, if $\underline{c}$ in (119) is independent of t.

The concurrent learning update law to estimate the unknown parameters is then given by

$$\dot{\hat{\theta}} = \alpha_\theta \Gamma_\theta \sum_{i=1}^{M} S_i \left(X_i - F_i - \hat{\theta}^T S_i - \hat{D}_i \right)^T, \tag{120}$$

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

$$\dot{\Gamma}_\theta = \beta_\theta \Gamma_\theta - \alpha_\theta \Gamma_\theta \mathscr{S} \Gamma_\theta, \tag{121}$$

where $\beta_\theta \in \mathbb{R}_{>0}$ is a forgetting factor. Using arguments similar to [60, Corollary 4.3.2], it can be shown that provided $\lambda_{\min}\left\{\Gamma_\theta^{-1}(0)\right\} > 0$, the least squares gain matrix satisfies

$$\underline{\Gamma}_\theta \mathrm{I}_p \leq \Gamma_\theta(t) \leq \overline{\Gamma}_\theta \mathrm{I}_p, \forall t \geq 0, \tag{122}$$

where $\underline{\Gamma}_\theta$ and $\overline{\Gamma}_\theta$ are positive constants. If a full rank history stack that satisfies (117) is not available *a priori*, then the data points can be recorded online using the relationship in (111), by selecting an increasing set of time-instances $\{t_i\}_{i=1}^{M}$ and letting

$$X_i = X(t_i), F_i = F(t_i), S_i = S(t_i), \hat{D}_i = \hat{D}(t_i). \tag{123}$$

Motivated by the observation that the rate of decay of the parameter estimation errors (see (124)) is proportional to the minimum singular value of $\mathscr{S}$, a singular value maximization algorithm is used to select the time instances $\{t_i\}_{i=1}^{M}$. That is, a data-point $\left(X_j, F_j, S_j, \hat{D}_j\right)$ in the history stack is replaced with a new data-point $\left(X^*, F^*, S^*, \hat{D}^*\right)$, where $F^* = F(t)$, $X^* = X(t)$, $S^* = S(t)$, and $\hat{D}^* = \hat{D}(t)$, for some t, only if

$$\lambda_{\min}\left(\sum_{i \neq j} S_i S_i^T + S_j S_j^T\right) \leq \lambda_{\min}\left(\sum_{i \neq j} S_i S_i^T + S^* S^{*T}\right),$$

where $\lambda_{\min}(\cdot)$ denotes the minimum singular value of a matrix.

The update law in (121) is motivated by the fact that the dynamics for the weight estimation error can be described by

$$\dot{\tilde{\theta}} = -\alpha_\theta \Gamma_\theta \mathscr{S} \tilde{\theta} - \alpha_\theta \Gamma_\theta \sum_{i=1}^{M} S_i \mathcal{E}_i^T, \tag{124}$$

where $\tilde{\theta} := \theta - \hat{\theta}$, which can be shown to be a perturbed stable linear time-varying system under conditions detailed in the following section.

The magnitude of the perturbation can be decreased by reducing the norm of $\mathcal{E}_i$, which can be reduced by improving the disturbance estimates $\hat{D}_i$ stored in the history stack. Since the disturbance estimation errors are assumed to converge exponentially to an ultimate bound, a time-based purging technique, where the history stack is periodically purged and replaced, is employed to leverage better estimates of $\hat{d}$ when they become available in order to yield more accurate estimates $\hat{\theta}$.

The purging technique utilizes two history stacks, a main history stack and a transient history stack, labeled $\mathcal{H}^{PE}$ and $\mathcal{G}^{PE}$, respectively. As soon as the transient history stack is full rank according to (119), $\mathcal{H}^{PE}$ is emptied and $\mathcal{G}^{PE}$ is copied into $\mathcal{H}^{PE}$. The history stack $\mathcal{H}^{PE}$ is kept constant in between purging instances.

Due to purging, the time instances $\{t_1, \cdots t_M\}$ and the matrices $\hat{D}$ and $\mathcal{E}$, and consequently $\mathcal{H}^{PE}$, are piecewise constant in time.

5.4.2 Analysis

A Lyapunov based analysis, summarized in the following theorem, shows convergence of the parameter estimator developed in Section 5.4.1.

Definition 5 *The signal* (x, u) *is called* finitely informative *(FI) if there exist time instances* $0 \leq t_1 < t_2 < \cdots < t_N$ *such that the resulting history stack is full rank and* persistently informative *(PI) if for any* $T \geq 0$*, there exist time instances* $T \leq t_1 < t_2 < \cdots < t_N$ *such that the resulting history stack is full rank.*

The stability result is summarized in the following theorem.

Theorem 4 *If a disturbance estimation technique that satisfies Assumption 12 is employed to generate estimates of d_2, the signal (x, u) is FI, the time instances $t_1, \ldots, t_M$ are selected using minimum singular value maximization so that the history stack, $\mathcal{H}^{PE}$, is full rank, uniformly in t, and $\mathcal{H}^{PE}$ is refreshed using a time-based purging algorithm, then $t \mapsto \tilde{\theta}(t)$ is ultimately bounded.*

Proof. Consider the candidate Lyapunov function

$$V_\theta(\tilde{\theta}, t) = \frac{1}{2}\tilde{\theta}^T \Gamma_\theta^{-1}(t)\tilde{\theta}. \tag{125}$$

Using the bounds in (122), the candidate Lyapunov function satisfies

$$\frac{1}{2\overline{\Gamma}_\theta}\left\|\tilde{\theta}\right\|^2 \leq V_\theta\left(\tilde{\theta}, t\right) \leq \frac{1}{2\underline{\Gamma}_\theta}\left\|\tilde{\theta}\right\|^2. \tag{126}$$

The time-derivative of (125) results in

$$\dot{V}_\theta(\tilde{\theta}, t) = \tilde{\theta}^T \Gamma_\theta^{-1}(t)\dot{\tilde{\theta}} + \frac{1}{2}\tilde{\theta}^T \dot{\Gamma}_\theta^{-1}(t)\tilde{\theta}. \tag{127}$$

Using (121) and (124), along with the identity $\dot{\Gamma}_\theta^{-1} = -\Gamma_\theta^{-1}\dot{\Gamma}_\theta\Gamma_\theta^{-1}$, $\dot{V}_\theta$ can be expressed as

$$\dot{V}_\theta(\tilde{\theta}, t) = -\frac{1}{2}\alpha_\theta\tilde{\theta}^T\mathscr{S}(t)\tilde{\theta} - \frac{1}{2}\beta_\theta\tilde{\theta}^T\Gamma_\theta^{-1}(t)\tilde{\theta} - \alpha_\theta\tilde{\theta}^T\sum_{i=1}^{M} S_i(t)\mathcal{E}_i^T(t). \tag{128}$$

Using the Cauchy-Schwartz inequality, and bounds in (119) and (122), $\dot{V}_\theta$ can be bounded by

$$\dot{V}_\theta(\tilde{\theta}, t) \leq -\frac{1}{2}\left(\alpha_\theta\underline{c} + \frac{\beta_\theta}{\overline{\Gamma}_\theta}\right)\left\|\tilde{\theta}\right\|^2 + \alpha_\theta\left\|\tilde{\theta}\right\|\sum_{i=1}^{M}\left\|S_i(t)\right\|\left\|\mathcal{E}_i(t)\right\|. \tag{129}$$

Since the states and controls are bounded, $\left\|S_i(t)\right\|$ is bounded for all i and for all $t \geq 0$. The upper bound is defined as $\overline{S} := \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \left\|\sigma\left(x, u\right)\right\|$. Using this upper bound, $\dot{V}_\theta$ can be rewritten as

$$\dot{V}_\theta(\tilde{\theta}, t) \leq -\frac{1}{4}\left(\alpha_\theta\underline{c} + \frac{\beta_\theta}{\overline{\Gamma}_\theta}\right)\left\|\tilde{\theta}\right\|^2, \ \forall \left\|\tilde{\theta}\right\| \geq \rho(\|\mu\|). \tag{130}$$

where $\mu = \sum_{i=1}^{M} \|\mathcal{E}_i\|$ and $\rho\left(\|\mu\|\right) = \left(\frac{4\alpha_\theta \overline{S}}{\alpha_\theta \underline{c} + \frac{\beta_\theta}{\overline{\Gamma}_\theta}}\right) \|\mu\|$. Using (126) and (130), [80, Theorem 4.19] can be invoked to conclude that the system in (124) is input-to-state stable with state $\tilde{\theta}$ and input μ.

If a time-based purging algorithm is implemented and if the signal (x, u) is FI, there exists a time instance T_s, such that for all $t \geq T_s$, the history stack $\mathcal{H}^{PE}$ remains unchanged. As a result, using Exercise 4.58 from [80], the ultimate bound on $\tilde{\theta}$ can be expressed as

$$\limsup_{t\to\infty} \left\|\tilde{\theta}(t)\right\| \leq \left(\sqrt{\frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta}} \frac{4\alpha_\theta \overline{S} T \left(\overline{\epsilon} + \overline{\tilde{d}}(T_s)\right)}{\alpha_\theta \underline{c} + \frac{\beta_\theta}{\overline{\Gamma}_\theta}}\right), \tag{131}$$

where $\overline{\epsilon} := \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \|\epsilon\left(x, u\right)\|$ and $\overline{\tilde{d}}(T_s)$ denotes a bound on the disturbance estimation error in the history stack $\mathcal{H}^{PE}(t)$ for all $t \geq T_s$.

Furthermore, if (x, u) is PI, then (131) reduces to

$$\limsup_{t\to\infty} \left\|\tilde{\theta}(t)\right\| \leq \left(\sqrt{\frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta}} \frac{4\alpha_\theta \overline{S} T \left(\overline{\epsilon} + \overline{d}_1 + \overline{\epsilon}_d\right)}{\alpha_\theta \underline{c} + \frac{\beta_\theta}{\overline{\Gamma}_\theta}}\right) := \overline{\theta}. \tag{132}$$

■

The ultimate bound for the estimation error, $\tilde{\theta}$, has a direct relationship to the approximation errors for the agent's dynamics, along with the disturbance estimation error. As such, the ultimate bound can be reduced by reducing those errors. This observation motivates the following corollary.

Corollary 2 *If the agent's dynamics in* (108) *are linearly parameterizable, both agents experience the same disturbance, (i.e.* $\overline{\epsilon}_d = 0$*), the signal* (x, u) *is PI, the time instances* $t_1, \ldots, t_M$ *are selected using minimum singular value maximization so that the history stack* $\mathcal{H}^{PE}$ *is full rank, uniformly in* t*,* $\mathcal{H}^{PE}$ *is refreshed using a time-based purging algorithm, and* $\tilde{d}$ *converges to zero exponentially, then* $\lim_{t\to\infty} \|\tilde{\theta}(t)\| = 0$.

Proof. Immediate from Theorem 4. ■

5.5 Inverse Reinforcement Learning

In this section, parameter estimates from Section 5.4 are utilized to form an error metric for IRL. The formulation of IRL in the following two subsections, builds off of the authors' previous work in [137].

5.5.1 Inverse Bellman Error

Under the premise that Agent 2 implements a feedback controller that would be optimal in a disturbance-free environment, the state and control trajectories, $x(\cdot)$ and $u(\cdot)$, satisfy the Hamilton-Jacobi-Bellman (HJB) equation[1] [95]

$$H\left(x\left(t\right), \nabla_x\left(V^*\left(x\left(t\right)\right)\right)^T, u\left(t\right)\right) = 0, \forall t \in \mathbb{R}_{\geq 0}, \tag{133}$$

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

The function V^* can be represented using $P \in \mathbb{N}$ basis functions as $V^*(x) = (W_V^*)^T \sigma_V(x) + \epsilon_V(x)$. The vector $W_V^* \in \mathbb{R}^P$ denotes ideal weights, $\sigma_V : \mathbb{R}^n \to \mathbb{R}^P$ denotes continuously differentiable known features, and $\epsilon_V : \mathbb{R}^n \to \mathbb{R}$ denotes approximation error. Given any constant $\overline{\epsilon}_V \in \mathbb{R}_{>0}$ [58, 59], there exist $P \in \mathbb{N}$ such that ϵ_V satisfies $\sup_{x \in \chi} \left\|\epsilon_V\left(x\right)\right\| < \overline{\epsilon}_V$ and $\sup_{x \in \chi} \left\|\nabla \epsilon_V\left(x\right)\right\| < \overline{\epsilon}_V$. Let $\hat{V} : \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}$, $\left(x, \hat{W}_V\right) \mapsto \hat{W}_V^T \sigma_V\left(x\right)$ and $\hat{Q} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$, $\left(x, \hat{W}_Q\right) \mapsto \hat{W}_Q^T \sigma_Q(x)$ be parameterized estimates of V^* and Q, respectively, where $\hat{W}_V \in \mathbb{R}^P$ are the estimates of W_V^* and $\hat{W}_Q$ are the estimates of W_Q^*. Furthermore, let $\hat{W}_R$ be the estimates of $W_R^* := [r_1 \ \ldots \ r_m]^T$. Using $\hat{\theta}$, $\hat{W}_V$, $\hat{W}_Q$, and $\hat{W}_R$, which are the estimates of θ, W_V^*, W_Q^*, and W_R^*, respectively, in (133), a parametric estimate of the Hamiltonian called the inverse Bellman error $\delta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+m} \times \mathbb{R}^p \to \mathbb{R}$ is obtained as

$$\delta\left(x, u, \hat{W}, \hat{\theta}\right) = \hat{W}_V^T \nabla_x \sigma_V\left(x\right) \hat{Y}(x, u, \hat{\theta}) + \hat{W}_Q^T \sigma_Q\left(x\right) + \hat{W}_R^T \sigma_u(u), \tag{134}$$

[1] For brevity, the full dependencies of the state trajectory, $x(t, x_0, u(\cdot))$, will be omitted wherever they are clear from the context and the trajectory will be denoted as $x(t)$.

where $\sigma_u(u) := \left[u_1^2, \ldots, u_m^2\right]^T$ and $\hat{Y}(x, u, \hat{\theta}) = \left[f^o(x, u) + \hat{g}(x, u, \hat{\theta})\right]$ and $\hat{g}(x, u, \hat{\theta}) := \hat{\theta}^T \sigma(x, u)$ from (108) with parameter estimates, $\hat{\theta}$. Rearranging, we get

$$\delta\left(x, u, \hat{W}', \hat{\theta}\right) = \left(\hat{W}'\right)^T \sigma'\left(x, u, \hat{\theta}\right), \tag{135}$$

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

$$\sigma'\left(x, u, \hat{\theta}\right) := \left[\left(\nabla_x \sigma_V\left(x\right) \hat{Y}(x, u, \hat{\theta})\right)^T, \left(\sigma_Q(x)\right)^T, \left(\sigma_u\left(u\right)\right)^T\right]^T.$$

5.5.2 Formulation of IRL

Using control signals, trajectories, and parameter estimates stored in a history stack, denoted as $\mathcal{H}^{IRL}$, the inverse Bellman error in Section 5.5.1, evaluated along the trajectories $x(\cdot)$ and $u(\cdot)$, at time instances $t_1(t), t_2(t), \ldots, t_N(t)$, can be formulated into the matrix form

$$\Delta'(t, \hat{W}') = \hat{\Sigma}'(t)\hat{W}', \tag{136}$$

where

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

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

Since the HJB equation in (133) is equal to zero along the optimal state and control trajectories, utilizing the current estimate of $\hat{\theta}$, candidate solutions for $\hat{W}'$ can be obtained by minimizing $\|\Delta'\|$ in (136). It can be seen that the solution $\hat{W}' = 0$ trivially minimizes $\|\Delta'\|$, which is expected due to the fact that optimal trajectories that result from minimization of all positive multiples of r are identical. As a result, r can only be identified up to a scaling factor using $x(\cdot)$ and $u(\cdot)$. To remove the scaling ambiguity without loss of generality, one reward weight will be assigned a fixed known value. In the following, it is assumed that the first element of $\hat{W}_R$ is known, denoted as r_1.

The inverse Bellman error in (135) can then be expressed as

$$\delta'\left(x, u, \hat{W}, \hat{\theta}\right) = \hat{W}^T \sigma''\left(x, u, \hat{\theta}\right) + r_1 \sigma_{u1}\left(u\right), \tag{137}$$

where $\hat{W} := \left[\hat{W}_V^T, \hat{W}_Q^T, \left(\hat{W}_R^-\right)^T\right]^T$, the vector $\hat{W}_R^-$ denotes $\hat{W}_R$ with the first element removed, $\sigma_{ui}\left(u\right)$ denotes the i-th element of the vector $\sigma_u\left(u\right)$, the vector σ_u^- denotes σ_u with the first element removed, and

$$\sigma''\left(x, u, \hat{\theta}\right) := \left[\left(\nabla_x \sigma_V\left(x\right) \hat{Y}(x, u, \hat{\theta})\right)^T, \left(\sigma_Q(x)\right)^T, \left(\sigma_u^-\left(u\right)\right)^T\right]^T.$$

Provided Assumption 9 is true, the closed-form optimal controller corresponding to the reward structure in (106) provides the relationship

$$-2Ru = \left(g'(x)\right)^T (\nabla_x \sigma_V\left(x\right))^T W_V^* + \left(g'(x)\right)^T \left(\nabla_x \epsilon_V\left(x\right)\right)^T, \tag{138}$$

which can be expressed as

$$\begin{aligned} -2r_1 u_1 + \Delta_{u1} &= \sigma_{g1} W_V^* \\ \Delta_{u^-} &= \sigma_g^- W_V^* + 2\mathrm{diag}\left([u_2, \ldots, u_m]\right)(W_R^*)^-, \end{aligned} \tag{139}$$

where $g'(x) := \left[\nabla_u f^o\right](x) + \theta^T \left[\nabla_u \sigma\right](x)$ ($\nabla_u f^o$ and $\nabla_u \sigma$ are independent of u since the dynamics are affine in control (Assumption 11)), σ_{g1} and Δ_{u1} denote the first rows and σ_g^- and Δ_{u^-} denote all but the first rows of $\sigma_g(x) := \left(g'(x)\right)^T \left(\nabla_x \sigma_V\left(x\right)\right)^T$ and $\Delta_u(x) := \left(g'(x)\right)^T \left(\nabla_x \epsilon_V(x)\right)^T$, respectively. For simplification, let $\sigma := \begin{bmatrix} \sigma'', & \begin{bmatrix} \sigma_g^T \\ \Theta \end{bmatrix} \end{bmatrix}$, where

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

Substituting $\hat{\theta}$, $\hat{W}_V$, and $\hat{W}_R$, in (139) and updating the history stack in (136) by removing the known reward weight element results in the linear system

$$-\Sigma_{u1}(t) - \hat{\Sigma}(t)\hat{W} = \hat{\Sigma}(t)\tilde{W} + \Delta(t), \tag{140}$$

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

$$\hat{\Sigma}(t) := \begin{bmatrix} \left(\sigma\left(x\left(t_1(t)\right), u\left(t_1(t)\right), \hat{\theta}\left(t_1(t)\right)\right)\right)^T \\ \vdots \\ \left(\sigma\left(x\left(t_N(t)\right), u\left(t_N(t)\right), \hat{\theta}\left(t_N(t)\right)\right)\right)^T \end{bmatrix},$$

$$\Sigma_{u1}(t) := \left[\left(\sigma'_{u1}\left(u\left(t_1(t)\right)\right)\right)^T, \cdots, \left(\sigma'_{u1}\left(u\left(t_N(t)\right)\right)\right)^T\right]^T,$$

$$\Delta(t) := \begin{bmatrix} \Delta_\delta\left(x(t_1(t)), u(t_1(t)), \tilde{\theta}(t_1(t))\right) \\ \Delta_m\left(x(t_1(t)), u(t_1(t)), \tilde{\theta}(t_1(t))\right) \\ \vdots \\ \Delta_\delta\left(x(t_N(t)), u(t_N(t)), \tilde{\theta}(t_N(t))\right) \\ \Delta_m\left(x(t_N(t)), u(t_N(t)), \tilde{\theta}(t_N(t))\right) \end{bmatrix},$$

where

$$\sigma'_{u1}(u(t_i(t))) := \left[r_1\sigma_{u1}(u(t_i(t))), 2r_1u_1(t_i(t)), 0_{1\times(m-1)}\right]^T,$$

$$\begin{aligned}\Delta_\delta\left(x(t_i(t)), u(t_i(t)), \tilde{\theta}(t_i(t))\right) := \Big[&\epsilon_Q(x(t_i(t))) \\ &+ \left(\sigma(x(t_i(t)), u(t_i(t)))\right)^T\tilde{\theta}(t_i(t))\left(\nabla_x\sigma_V(x(t_i(t)))\right)^T W_V^* \\ &+ \left(f^o(x(t_i(t)), u(t_i(t)))\right)^T\left(\nabla_x\epsilon_V(x(t_i(t)))\right)^T \\ &+ \left(\theta^T\sigma(x(t_i(t)), u(t_i(t)))\right)^T\left(\nabla_x\epsilon_V(x(t_i(t)))\right)^T\Big],\end{aligned}$$

$$\Delta_m\Big(x(t_i(t)),u(t_i(t)),\tilde{\theta}(t_i(t))\Big) := \Big[\Big(\nabla_u f^o(x(t_i(t)),u(t_i(t)))\Big)^T\Big(\nabla_x \epsilon_V(x(t_i(t)))\Big)^T + \Big(\nabla_u \sigma(x(t_i(t)),u(t_i(t)))\Big)^T \tilde{\theta}(t_i(t))\Big(\nabla_x \sigma_V(x(t_i(t)))\Big)^T W_V^* + \Big(\nabla_u \sigma(x(t_i(t)),u(t_i(t)))\Big)^T \theta\Big(\nabla_x \epsilon_V(x(t_i(t)))\Big)^T\Big].$$

The relationship in (140) suggests the following update law for estimation of the unknown reward function weights

$$\dot{\hat{W}} = \alpha\Gamma(t)\hat{\Sigma}^T(t)\left(-\hat{\Sigma}(t)\hat{W} - \Sigma_{u1}(t)\right), \tag{141}$$

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

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\hat{\Sigma}^T(t)\hat{\Sigma}(t)\Gamma, \tag{142}$$

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

The update law in (141) is motivated by the fact that, the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}} = -\alpha\Gamma(t)\hat{\Sigma}^T(t)\left(\hat{\Sigma}(t)\tilde{W} + \Delta(t)\right), \tag{143}$$

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

5.5.3 Analysis

A Lyapunov based analysis is used to show convergence of the IRL method in Section 5.5.2.

Convergence of the estimation error to a neighborhood of the origin follow under the following condition on the regressor, $\hat{\Sigma}$.

Definition 6 *The time-varying history stack, $\mathcal{H}^{IRL}$, is called full rank, uniformly in t, if there exists $\underline{\sigma} \in \mathbb{R}_{>0}$ such that*[2] $\forall t \in \mathbb{R}_{\geq 0}$,

$$\underline{\sigma} < \lambda_{\min}\left\{\hat{\Sigma}^{T}(t)\hat{\Sigma}(t)\right\}. \tag{144}$$

Using arguments similar to [60, Corollary 4.3.2], it can be shown that if $\lambda_{\min}\left\{\Gamma^{-1}\left(0\right)\right\} > 0$, and if H^{IRL} is full rank, uniformly in t, then the least squares gain matrix satisfies

$$\underline{\Gamma}\mathrm{I}_{L+P+m-1} \leq \Gamma\left(t\right) \leq \overline{\Gamma}\mathrm{I}_{L+P+m-1}, \forall t > 0, \tag{145}$$

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

The stability result is summarized in the following theorem.

Theorem 5 *If there exists a disturbance estimation technique that satisfies Assumption 12, the signal (x, u) is FI, the time instances $t_1, \ldots, t_M$ and $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that the history stacks $\mathcal{H}^{PE}$ and $\mathcal{H}^{IRL}$ are full rank, uniformly in t, and $\mathcal{H}^{PE}$ and $\mathcal{H}^{IRL}$ are refreshed using a time-based purging algorithm, then $t \mapsto \tilde{W}\left(t\right)$ is ultimately bounded.*

Proof. Consider the positive definite candidate Lyapunov function

$$V(\tilde{W}, t) = \frac{1}{2}\tilde{W}^{T}\Gamma^{-1}\left(t\right)\tilde{W}. \tag{146}$$

Using the bounds in (145), the candidate Lyapunov function satisfies

$$\underline{v}\left\|\tilde{W}\right\|^{2} \leq V\left(\tilde{W}, t\right) \leq \overline{v}\left\|\tilde{W}\right\|^{2}. \tag{147}$$

where $\underline{v} := {}^{1}/_{2\overline{\Gamma}}$ and $\overline{v} := {}^{1}/_{2\underline{\Gamma}}$.

The time-derivative of (146) results in

$$\dot{V}(\tilde{W}, t) = \tilde{W}^{T}\Gamma^{-1}\left(t\right)\dot{\tilde{W}} + \frac{1}{2}\tilde{W}^{T}\dot{\Gamma}^{-1}\left(t\right)\tilde{W}. \tag{148}$$

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

Using (142) and (143), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, after simplifying the time-derivative can be expressed as

$$\dot{V}(\tilde{W},t) = -\frac{1}{2}\alpha\tilde{W}^T\hat{\Sigma}^T(t)\hat{\Sigma}(t)\tilde{W} - \alpha\tilde{W}^T\hat{\Sigma}^T(t)\Delta(t) - \frac{1}{2}\beta\tilde{W}^T\Gamma^{-1}\left(t\right)\tilde{W}. \tag{149}$$

Substituting in $\hat{\Sigma} = \Sigma - \tilde{\Sigma}$, yields

$$\begin{aligned}\dot{V}(\tilde{W},t) = -\frac{1}{2}\alpha\tilde{W}^T\hat{\Sigma}^T(t)\hat{\Sigma}(t)\tilde{W} - \frac{1}{2}\beta\tilde{W}^T\Gamma^{-1}\left(t\right)\tilde{W}& \\ - \alpha\tilde{W}^T\Sigma^T(t)\Delta(t) + \alpha\tilde{W}^T\tilde{\Sigma}^T(t)\Delta(t).&\end{aligned} \tag{150}$$

Using the Cauchy-Schwartz inequality, and bounds in (145) and (144), $\dot{V}$ can be bounded by

$$\begin{aligned}\dot{V}(\tilde{W},t) \leq -\frac{1}{2}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\overline{\Gamma}}}\beta\right)\left\|\tilde{W}\right\|^2 + \alpha\left\|\tilde{W}\right\|\left\|\Sigma(t)\right\|\left\|\Delta(t)\right\|& \\ + \alpha\left\|\tilde{W}\right\|\left\|\tilde{\Sigma}(t)\right\|\left\|\Delta(t)\right\|.&\end{aligned} \tag{151}$$

The term $\left\|\tilde{\Sigma}\right\|$ can be expressed in terms of $\tilde{\theta}$ as

$$\left\|\tilde{\Sigma}(t)\right\| \leq \overline{\overline{\tilde{\theta}}}(t)\ \overline{\Sigma}, \tag{152}$$

where $\overline{\overline{\tilde{\theta}}}(t) = \max_{i=1,2,\ldots,N}\|\tilde{\theta}(t_i(t))\|$. Since $t \mapsto x(t), t \mapsto \hat{\theta}(t)$, and $t \mapsto u(t)$ are bounded, the coefficient $\overline{\Sigma}$ can be selected independent of t_i and the specific trajectories of x and u currently stored in the history stack as

$$\overline{\Sigma} := \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}}\left\{\left\|\nabla_x\sigma_V(x)\right\|\left\|\sigma(x,u)\right\|, \left\|\nabla_x\sigma_V(x)\right\|\left\|\nabla_u\sigma(x,u)\right\|\right\}. \tag{153}$$

The term $\|\Sigma\|$, which contains true values of the unknown parameters, is bounded above since it is a function of only true parameters, θ, and bounded states and controls, x and u. Let the upper bound on $\|\Sigma\|$ be denoted as

$$\left\|\Sigma(t)\right\| \leq \overline{\Sigma}_\theta,\ \forall\, t \geq 0. \tag{154}$$

The residual $\|\Delta\|$ can be bounded above by

$$\|\Delta(t)\| \leq \overline{\overline{\tilde{\theta}}}(t)\ \overline{\Delta} + \overline{\Delta}_\epsilon, \tag{155}$$

where

$$\overline{\Delta} := \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \Big\{ \|\Delta_\delta\|, \|\Delta_m\| \Big\}$$
$$\overline{\Delta}_\epsilon := \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \Big\{ \|\Delta_{\delta\epsilon}\|, \|\Delta_{m\epsilon}\| \Big\}.$$

and

$$\begin{aligned}
\|\Delta_\delta\| :=& \|\sigma(x,u)\|\|\nabla_x\sigma_V(x)\|\|W_V^*\|,\\
\|\Delta_{\delta\epsilon}\| :=& \|\epsilon_Q(x)\| + \|\nabla_x\epsilon_V(x)\|\|f^o(x,u)\|\\
&+ \|\nabla_x\epsilon_V(x)\|\|\theta\|\|\sigma(x,u)\|,\\
\|\Delta_m\| :=& \|\nabla_u\sigma(x,u)\|\|\nabla_x\sigma_V(x)\|\|W_V^*\|,\\
\|\Delta_{m\epsilon}\| :=& \|\nabla_x\epsilon_V(x)\|\|\nabla_u f^o(x,u)\|\\
&+ \|\nabla_x\epsilon_V(x)\|\|\theta\|\|\nabla_u\sigma(x,u)\|.
\end{aligned}$$

Using (152), (154) and (155), $\dot{V}$ becomes

$$\dot{V}(\tilde{W},t) \le -\frac{1}{4}\left(\alpha\underline{\sigma} + \frac{\beta}{\overline{\overline{\Gamma}}}\right)\left\|\tilde{W}\right\|^2, \forall \left\|\tilde{W}\right\| \ge \rho(\|\mu\|), \tag{156}$$

where $\mu = \left[\overline{\Delta}_\epsilon, \overline{\Delta}_\epsilon\overline{\tilde{\theta}}, \overline{\tilde{\theta}}, \overline{\tilde{\theta}}^2\right]^T$ and

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

Using (147) and (156), [80, Theorem 4.19] can be invoked to conclude that the system in (143) is input-to-state stable with state $\tilde{W}$ and input μ.

If a time-based purging algorithm is implemented and if the signal (x,u) is FI, there exists a time instance T_s, such that for all $t \ge T_s$, the history stack $\mathcal{H}^{IRL}(t)$ remains unchanged. As a result, using Exercise 4.58 from [80], the ultimate bound on $\tilde{W}$ can be expressed as

$$\limsup_{t\to\infty}\left\|\tilde{W}(t)\right\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\max\{\overline{\Sigma}_\theta,\overline{\Sigma},\overline{\Delta}\ \overline{\Sigma}_\theta,\overline{\Delta}\ \overline{\Sigma}\}}{\alpha\underline{\sigma}+\frac{\beta}{\overline{\Gamma}}}\right)\overline{\Delta}_\epsilon$$
$$+\sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\max\{\overline{\Sigma}_\theta,\overline{\Sigma},\overline{\Delta}\ \overline{\Sigma}_\theta,\overline{\Delta}\ \overline{\Sigma}\}}{\alpha\underline{\sigma}+\frac{\beta}{\overline{\Gamma}}}\right)\left(\overline{\Delta}_\epsilon\overline{\tilde{\theta}}(T_s)+\overline{\tilde{\theta}}(T_s)\right)$$
$$+\sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\max\{\overline{\Sigma}_\theta,\overline{\Sigma},\overline{\Delta}\ \overline{\Sigma}_\theta,\overline{\Delta}\ \overline{\Sigma}\}}{\alpha\underline{\sigma}+\frac{\beta}{\overline{\Gamma}}}\right)\left(\overline{\tilde{\theta}}(T_s)\right)^2, \quad (157)$$

where $\overline{\tilde{\theta}}(T_s)$ denotes a bound on the parameter estimation error in the history stack $\mathcal{H}^{IRL}(t)$ for all $t \geq T_s$.

Furthermore, if (x, u) is PI, then $\limsup_{t\to\infty}\overline{\tilde{\theta}}(t) \to \overline{\theta}$. In that case, the ultimate bound reduces to

$$\limsup_{t\to\infty}\left\|\tilde{W}(t)\right\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\max\{\overline{\Sigma}_\theta,\overline{\Sigma},\overline{\Delta}\ \overline{\Sigma}_\theta,\overline{\Delta}\ \overline{\Sigma}\}}{\alpha\underline{\sigma}+\frac{\beta}{\overline{\Gamma}}}\right)\overline{\Delta}_\epsilon$$
$$+\sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\max\{\overline{\Sigma}_\theta,\overline{\Sigma},\overline{\Delta}\ \overline{\Sigma}_\theta,\overline{\Delta}\ \overline{\Sigma}\}}{\alpha\underline{\sigma}+\frac{\beta}{\overline{\Gamma}}}\right)\left(\overline{\Delta}_\epsilon\overline{\theta}+\overline{\theta}+\overline{\theta}^2\right). \quad (158)$$

■

The ultimate bound for the estimation error, $\tilde{W}$, has a direct relationship to the approximation errors for the agent's dynamics, along with both the reward and value function approximation errors. As such, the ultimate bound can be reduced by reducing those errors. This observation motivates the following corollary.

Corollary 3 *If the agent's dynamics in* (108) *are linearly parameterizable, the approximation error terms ϵ_Q, ϵ_V are equal to zero, both agents experience the same disturbance, i.e. $\overline{\epsilon}_d = 0$, the signal (x, u) is PI, the time instances $t_1, \ldots, t_M$ and $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that $\mathcal{H}^{PE}$ and $\mathcal{H}^{IRL}$ are full rank, uniformly in t, and $\mathcal{H}^{PE}$ and $\mathcal{H}^{IRL}$ are refreshed using a time-based purging algorithm, then* $\lim_{t\to\infty}\|\tilde{W}(t)\| = 0$.

Proof. Immediate from Theorem 5. ■

If the agent's dynamics are known, then the developed IRL method in Section 5.5.2 does not require any disturbance estimation. Since the IRL method is model-based and only requires optimal state-action pairs, not optimal state-action trajectories. IRL with exact model knowledge does not require disturbance estimation, and as such, does not require Assumptions 8 or 12.

Theorem 6 *If the dynamics of Agent 2 in* (105) *are known, along with an exact basis for approximation of Q and V^* (i.e., $\epsilon_Q = 0$ and $\epsilon_V = 0$), the signal (x, u) is FI, the time instances $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that $\mathcal{H}^{IRL}$ is full rank, uniformly in t, then as $t \to \infty$, $\|\tilde{W}(t)\| \to 0$, exponentially.*

Proof. Consider the positive definite candidate Lyapunov function

$$V(\tilde{W}, t) = \frac{1}{2}\tilde{W}^T \Gamma^{-1}(t)\, \tilde{W}. \tag{159}$$

Using the bounds in (145), the candidate Lyapunov function satisfies

$$\underline{v} \left\|\tilde{W}\right\|^2 \leq V\left(\tilde{W}, t\right) \leq \overline{v} \left\|\tilde{W}\right\|^2. \tag{160}$$

where $\underline{v} := {}^1\!/_{2\overline{\Gamma}}$ and $\overline{v} := {}^1\!/_{2\underline{\Gamma}}$.

The time-derivative of (159) results in

$$\dot{V}(\tilde{W}, t) = \tilde{W}^T \Gamma^{-1}(t)\, \dot{\tilde{W}} + \frac{1}{2}\tilde{W}^T \dot{\Gamma}^{-1}(t)\, \tilde{W}. \tag{161}$$

Using (142) and (143), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, after simplifying the time-derivative can be expressed as

$$\dot{V}(\tilde{W}, t) = -\frac{1}{2}\alpha \tilde{W}^T \hat{\Sigma}^T(t)\hat{\Sigma}(t)\tilde{W} - \alpha \tilde{W}^T \hat{\Sigma}^T(t)\Delta(t) - \frac{1}{2}\beta \tilde{W}^T \Gamma^{-1}(t)\, \tilde{W}. \tag{162}$$

Since the dynamics are known, there is no approximation error, i.e. $\Delta = 0$. Then (162) becomes

$$\dot{V}(\tilde{W}, t) = -\frac{1}{2}\alpha \tilde{W}^T \hat{\Sigma}^T(t)\hat{\Sigma}(t)\tilde{W} - \frac{1}{2}\beta \tilde{W}^T \Gamma^{-1}(t)\, \tilde{W}. \tag{163}$$

Using the Cauchy-Schwartz inequality, and bounds in (145) and (144), $\dot{V}$ can be bounded by

$$\dot{V}(\tilde{W},t) \leq -\frac{1}{2}\left(\alpha\underline{\sigma} + \frac{1}{\overline{\Gamma}}\beta\right)\left\|\tilde{W}\right\|^2 \tag{164}$$

Using (159) and (164), [80, Theorem 4.10] can be invoked to conclude that $\tilde{W}$ converges exponentially to 0. ■

5.6 Simulation

To demonstrate the performance of the developed method, a nonlinear optimal control problem that has a known optimal value function is constructed using [65].

5.6.1 Uncertain Agent Dynamics

Agent 1 has the following nonlinear dynamics

$$\dot{x}_{1_1} = x_{1_2}, \qquad \dot{x}_{1_2} = x_{1_1}x_{1_2} + 3x_{1_2}^2 + 5u_1 + d_1.$$

Agent 2 under observation has the following nonlinear dynamics

$$\begin{aligned}
\dot{x}_{2_1} &= x_{2_2}, \\
\dot{x}_{2_2} &= \theta_1 x_{2_1}\left(\frac{\pi}{2} + \tan^{-1}(5x_{2_1})\right) + \frac{\theta_2 x_{2_1}^2}{1+25x_{2_1}^2} + \theta_3 x_{2_2} + 3u_2 + d_2,
\end{aligned} \tag{165}$$

where x_{i_j} denotes state j for Agent i. The parameters θ_1, θ_2, and θ_3 are unknown constants to be estimated and d_i is the unknown disturbance acting on Agent i. The exact values of these parameters are $\theta_1 = -1, \theta_2 = -\frac{5}{2}$, and $\theta_3 = 4$. Inspired by [34], the disturbance acting on the Agent 1 is assumed to be generated from the exogenous linear system

$$\dot{\zeta} = A\zeta, \tag{166}$$

$$d_1 = C\zeta, \tag{167}$$

where $\zeta : \mathbb{R}_{\geq 0} \to \mathbb{R}^N, A \in \mathbb{R}^{N \times N}, C \in \mathbb{R}^{n \times N}$, and $d : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the disturbance, and where $A = [0, 1; -1, 0]$ and $C = [0, 0; 1, 0]$. The disturbance estimator[3] is designed as

$$\dot{\hat{\zeta}} = A\hat{\zeta} + K\left(\dot{x}_1 - \left(f_1\left(x_1, u_1\right) + \hat{d}_1\right)\right), \tag{168}$$

and

$$\hat{d}_1 = C\hat{\zeta}, \tag{169}$$

where $K \in \mathbb{R}^{N \times n}$ is a gain matrix and chosen as $K = [1, 0.5; 0, 5]$. The disturbance acting on Agent 2 is the same disturbance, d_1, with an additive zero-mean Gaussian noise with variance 0.1.

The performance index that Agent 2 is trying to minimize is

$$J(x_2^o, u_2(\cdot)) = \int_0^\infty (x_{2_2}^2 + u_2^2)\,\mathrm{d}t, \tag{170}$$

resulting in the ideal reward function weights $Q = \text{diag}([q_1, q_2]) = \text{diag}([0, 1])$ and $R = 1$. The observed state and control trajectories and the disturbance estimates are used to estimate the unknown parameters in the dynamics of Agent 2, along with the optimal value function parameters and the reward function weights. The optimal controller is $u_2^* = -3x_{2_2}$, while the optimal value function is $V^* = x_{2_1}^2(v_1 + v_2 \tan^{-1}(5x_{2_1})) + v_3 x_{2_2}^2$, resulting in the ideal optimal value function parameters $v_1 = \frac{\pi}{2}$, $v_2 = 1$, and $v_3 = 1$.

Figs. 19 - 21 show the performance of the proposed method. As seen in Figs. 19 and 21, the estimation errors of unknown part of the dynamics of Agent 2 and the unknown disturbance affecting Agent 2 converge to a bound near the origin. As seen in Fig. 20, the IRL approach is able to estimate the ideal values of the reward and optimal value functions online even with non-zero ultimate bounds on the disturbance and parameter estimates. The parameters used for the simulation

[3] Since the disturbance estimation error using (168) exponentially converges to the origin, it trivially satisfied Assumption 12.

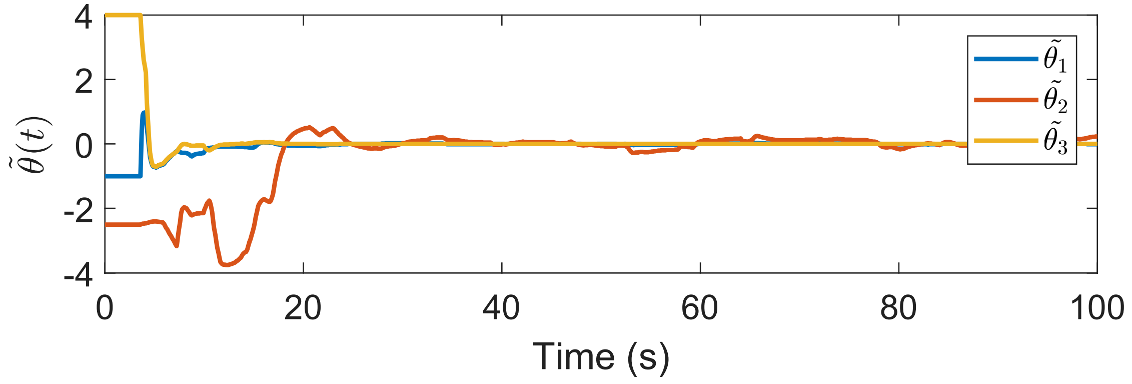


Figure 19: Estimation error for the unknown parameters in the dynamics of Agent 2.

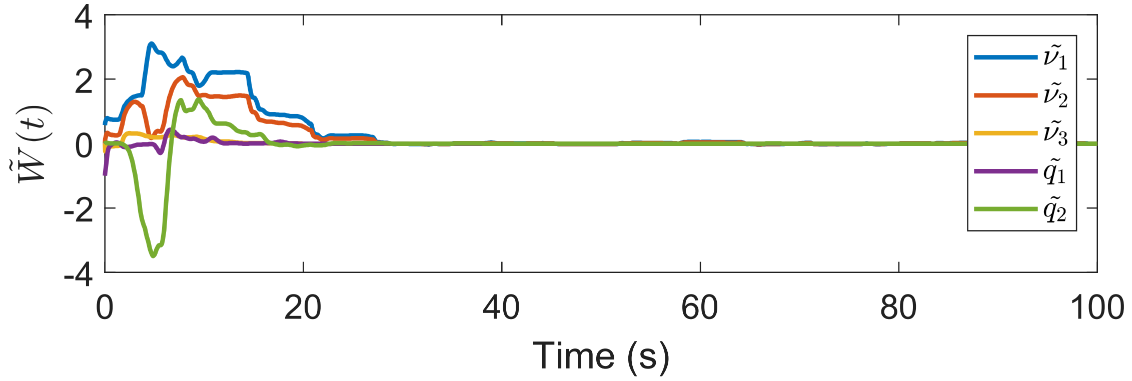


Figure 20: Estimation error for the unknown parameters in the reward function for Agent 2.

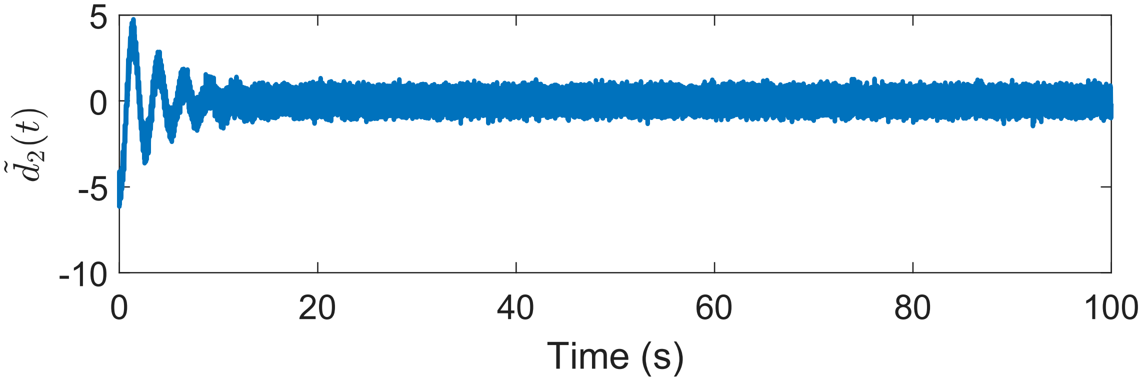


Figure 21: Estimation error for the unknown disturbance acting on Agent 2.

are: $T = 1.2s$, $N = 100$, $M = 150$, $\beta = \beta_\theta = 0.5$, $\alpha = \alpha_\theta = {}^1\!/_N$, and a time step of $0.0005s$.

5.6.2 Exact Model Knowledge

The second simulation is the same as in Section 5.6.1, however, this simulation utilizes known dynamics of Agent 2. Since Agent 2 is trying to minimize the performance index in (170), even though Agent 2 is still affected by an unknown disturbance, each action that Agent 2 takes is the optimal instantaneous action for a given state corresponding to the reward function in (170). Therefore, since the state-action pairs are optimal, and there are no estimation errors for the dynamics of Agent 2, the resulting error term Δ in (140) is equal to 0. Therefore, disturbance estimation is not required in this situation to estimate the unknown reward function. Fig. 22 shows the performance of the proposed method.

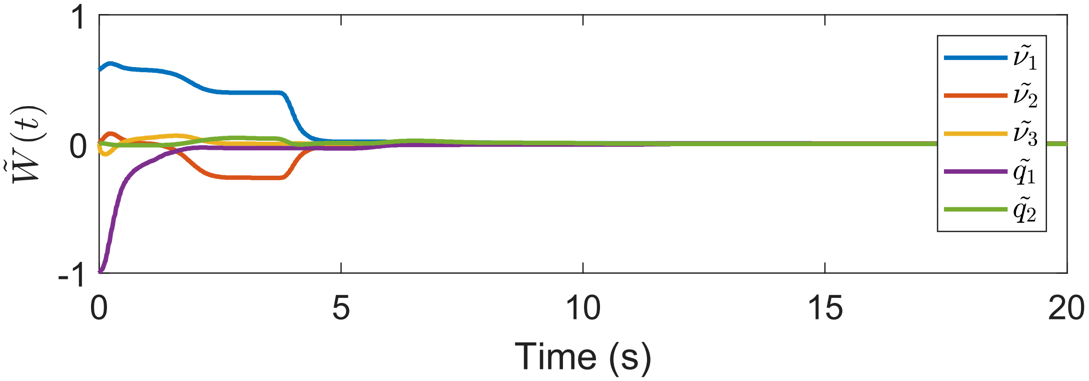


Figure 22: Estimation error for the unknown parameters in the reward function for Agent 2 with exact model knowledge.

As seen in Fig. 22, the reward function and optimal value function estimation result in perfect estimates of the unknown weights without the need for disturbance estimation. This would potentially be useful for real-world implementation, as certain situations may facilitate better knowledge of the system dynamics than accurate disturbance estimation, such as quadcopters flying in highly turbulent and unpredictable wind fields.

5.7 Conclusion

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

Chapter VI

INVERSE REINFORCEMENT LEARNING WITH LIMITED DATA

Inverse reinforcement learning has a large dependency on the information contained in observed demonstrations. However, this data dependency becomes an issue for trajectories with sparse data, and attempting to recover reward functions from a single demonstration under these sparse data conditions is challenging. Most current methods developed for IRL require multiple trajectories to uncover reward functions. This chapter aims to resolve IRL for trajectories with sparse data.

6.1 Introduction

While IRL in an *offline* setting has a rich history of literature [3,5,92,93,116,118,126, 132,143,154,157,159], these offline approaches to IRL are ill-suited for adaptation in real-time, and as a result, cannot handle changes to task objectives. The development of online IRL is motivated by the need for robustness to uncertainties in the system model and responsiveness to adapt to changing reward structures. However, little work has been done to address IRL in an online setting and one reason for this is the limited data provided by a single demonstration.

Preliminary results on online IRL are available for linear systems, in results such as [68] and [110], and for nonlinear systems, in results such as in [138], [137], and Chapters IV and V. However, [68], [138] and Chapter IV exploit access to demonstrator's feedback policy, [110] requires exact model knowledge, and [137] and Chapter V exploit similar disturbances to provide sufficient excitation. The main contribution of this chapter is the development of a novel method for reward function estimation for an agent in situations where estimation of the demonstrator's optimal feedback law is less data-intensive than direct estimation of its reward function.

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

The chapter is organized as follows: Section 6.2 introduces the problem formulation. Section 6.3 develops the update law for the optimal controller. Section 6.4 details the analysis for the optimal controller estimator. Section 6.5 introduces the IRL algorithm. Section 6.6 shows the IRL convergence analysis. Section 6.7 shows simulation examples, and Section 6.8 concludes the chapter.

6.2 Problem Formulation

Consider an agent under observation with the dynamics

$$\begin{aligned}\dot{x} &= f(x,u),\\ y &= h(x,u), \qquad (171)\end{aligned}$$

where $x \in \mathbb{R}^n$ is the state, $f : \mathbb{R}^{n\times m} \to \mathbb{R}^n$ denotes the uncertain dynamics, $u \in \mathbb{R}^m$ is the control, $y \in \mathbb{R}^l$ is the output, and $h : \mathbb{R}^{n\times m} \to \mathbb{R}^l$ denotes the measurement model. If a nominal dynamic model of the agent is available, then the dynamics in

(171) can then be separated into

$$\dot{x} = f^o(x,u) + g(x,u), \tag{172}$$

where $f^o : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the nominal model, $g \in \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the uncertainty[1].

The following assumption is required for the proposed methods.

Assumption 13 *The partial derivative of f in* (172) *with respect to x and u are locally Lipschitz continuous.*

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

$$J(x_0, u(\cdot)) = \int_0^\infty r(x(t; x_0, u_{[0,t)}), u(t))\,\mathrm{d}t, \tag{173}$$

where $x(\cdot; x_0, u_{[0,t)})$ is the trajectory of the agent generated using the control signal $u(\cdot)$, restricted to the time interval $[0,t)$, starting from the initial condition x_0. The main objective of the paper is to estimate the unknown reward function r, in the presence of uncertain dynamics, using measurements of the input $u(\cdot)$ and the output $t \mapsto y(t) = h\left(x(t, x_0, u_{[0,t)}), u(t)\right)$, under the assumption that $u(t)$ is the optimal action in response to the state $x(t, x_0, u_{[0,t)})$.

In the following, the input and the output signals available for measurement will be denoted by $t \mapsto u(t)$ and $t \mapsto y(t)$, respectively, the corresponding unknown true state will be denoted by $t \mapsto x(t)$, and x and u will be used to denote generic elements of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

The following assumptions are used throughout the analysis.

Assumption 14 *The dynamics in* (171) *is affine in control and the optimal control problem defined by* (171), (173), *and* (77) *admits a twice continuously differentiable optimal value function.*

[1] If a nominal model is not available, $f^o(x,u) := 0 \;\forall\; (x,u) \in \mathbb{R}^n \times \mathbb{R}^m$.

The class of affine systems is large, it includes linear systems and Euler Lagrange systems with invertible inertia matrices. While twice continuous differentiability of the value function is a strict requirement, many optimal control problems of interest, such as linear quadratic problems and nonlinear problems similar to those used for demonstration in Section 4.5.3, meet this requirement.

Assumption 15 *The unknown reward function r is quadratic in control, i.e.,*

$$r(x,u) = Q(x) + u^T R u, \tag{174}$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite (P.D.) matrix and $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive semi-definite (P.S.D.) continuously differentiable function with a locally Lipschitz continuous gradient.

Remark 6 *Since R can be selected to be symmetric without loss of generality, the developed IRL method only estimates the elements of R that are on and above the main diagonal.*

Assumption 16 *The state and control trajectories are bounded such that $x(t) \in \mathcal{X}$, $u(t) \in \mathcal{U}$ for some compact sets $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$.*

Under the premise that the observed agent makes optimal decisions, the state and control trajectories, $x(\cdot)$ and $u(\cdot)$, satisfy the Hamilton-Jacobi-Bellman equation[2] [95]

$$H\left(x\left(t\right), \nabla_x \left(V^*\left(x\left(t\right)\right)\right)^T, u\left(t\right)\right) = 0, \forall t \in \mathbb{R}_{\geq 0}, \tag{175}$$

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

The functions V^* and Q can be represented using $P \in \mathbb{N}$ and $L \in \mathbb{N}$ basis functions, respectively, as $V^*(x) = (W_V^*)^T \sigma_V(x) + \epsilon_V(x)$ and $Q(x) = (W_Q^*)^T \sigma_Q(x) +$

[2] For brevity, the full dependencies of the state trajectory, $x(t, x_0, u(\cdot))$, will be omitted wherever they are clear from the context and the trajectory will be denoted as $x(t)$.

$\epsilon_Q(x)$. The vectors $W_V^* := [v_1 \dots v_P]^T \in \mathbb{R}^P$ and $W_Q^* := [q_1 \, \dots \, q_L]^T \in \mathbb{R}^L$ denote ideal weights, $\sigma_V : \mathbb{R}^n \to \mathbb{R}^P$ and $\sigma_Q : \mathbb{R}^n \to \mathbb{R}^L$ denote continuously differentiable known features with locally Lipschitz continuous gradients, and $\epsilon_V : \mathbb{R}^n \to \mathbb{R}$ and $\epsilon_Q : \mathbb{R}^n \to \mathbb{R}$ denote approximation errors. Given any constants $\overline{\epsilon}_V, \overline{\epsilon}_Q \in \mathbb{R}_{>0}$, there exist $P, L \in \mathbb{N}$ such that ϵ_V and ϵ_Q satisfy $\sup_{x\in\chi} \left\|\epsilon_V(x)\right\| < \overline{\epsilon}_V$, $\sup_{x\in\chi} \left\|\nabla\epsilon_V(x)\right\| < \overline{\epsilon}_V$, $\sup_{x\in\chi} \left\|\epsilon_Q(x)\right\| < \overline{\epsilon}_Q$, and $\sup_{x\in\chi} \left\|\nabla\epsilon_Q(x)\right\| < \overline{\epsilon}_Q$ [58,59]. Let $\hat{V} : \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}$, $\left(x, \hat{W}_V\right) \mapsto \hat{W}_V^T \sigma_V(x)$ and $\hat{Q} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$, $\left(x, \hat{W}_Q\right) \mapsto \hat{W}_Q^T \sigma_Q(x)$ be parameterized estimates of V^* and Q, respectively, where $\hat{W}_V$ and $\hat{W}_Q$ are estimates of W_V^* and W_Q^*, respectively. Furthermore, let $u^T R u$ be parameterized as $u^T R u = \left(W_R^*\right)^T \sigma_{R1}(u)$ where $\sigma_{R1} : \mathbb{R}^m \to \mathbb{R}^M$, are the basis functions, selected as

$$\begin{aligned}\sigma_{R1}(u) := [&u_1^2, 2u_1u_2, 2u_1u_3, \dots, 2u_1u_m, u_2^2,\\ &2u_2u_3, 2u_2u_4, \dots, u_{m-1}^2, \dots, 2u_{m-1}u_m, u_m^2]^T,\end{aligned}$$

and $W_R^* \in \mathbb{R}^M$, are the ideal weights, given by

$$W_R^* = \left[R_{11}, 2R_1^{(-1)}, R_{22}, 2R_2^{(-2)}, \dots, 2R_{m-1}^{-(m-1)}, R_{mm}\right]^T,$$

where, for a given matrix $R \in \mathbb{R}^{m\times m}$, R_{ij} denotes the corresponding element in the i-th row and the j-th column of the matrix R, and $R_i^{(-j)}$ denotes the i-th row of the matrix E with the first j elements removed, i.e., $R_3^{(-3)} := \left[R_{34}, R_{35}, \dots, R_{3(m-1)}, R_{3m}\right]$.

Using $\hat{W}_V$ and $\hat{W}_Q$, along with estimates $\hat{W}_R$ of W_R^*, in (175), a parametric estimate of the Hamiltonian called the inverse Bellman error $\delta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{L+P+M} \to \mathbb{R}$ is obtained as

$$\delta\left(x, u, \hat{W}'\right) = \hat{W}_V^T \nabla_x \sigma_V(x) f(x,u) + \hat{W}_Q^T \sigma_Q(x) + \hat{W}_R^T \sigma_{R1}(u), \qquad (176)$$

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

Since (176) utilizes the agent's dynamics, the IRL technique developed in this paper is model-based, and as such, an accurate model is required to estimate the unknown reward function. To facilitate estimation under modeling uncertainties, a system identifier is utilized that estimates the unknown model parameters.

The unknown function g in (172) can be represented using basis functions as

$$g\left(x,u,\theta\right)=\theta^{T}\sigma\left(x,u\right)+\epsilon\left(x,u\right), \tag{177}$$

where $\sigma\in\mathbb{R}^{n}\times\mathbb{R}^{m}\to\mathbb{R}^{p}$ and $\epsilon:\mathbb{R}^{n}\times\mathbb{R}^{m}\to\mathbb{R}^{n}$ denote the basis vector and the approximation error, respectively, and $\theta\in\mathbb{R}^{p\times n}$ is a constant matrix of unknown parameters. Given any constant $\overline{\epsilon}$, there exist $p\in\mathbb{N}$ and $\overline{\sigma},\overline{\theta}\in\mathbb{R}_{>0}$ such that $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\Vert \sigma\left(x,u\right)\right\Vert <\overline{\sigma}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\Vert \nabla\sigma\left(x,u\right)\right\Vert <\overline{\sigma}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\Vert \epsilon\left(x,u\right)\right\Vert <\overline{\epsilon}$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\Vert \nabla\epsilon\left(x,u\right)\right\Vert <\overline{\epsilon}$, and $\left\Vert \theta\right\Vert <\overline{\theta}$.

To focus the discussion on the key contributions of the work, it is assumed that a state and parameter estimator that satisfies the following properties is available.

Assumption 17 *There exists a state and parameter estimator that yields a time instance, $\overline{T}$, such that the state and parameter estimation errors, $\tilde{x}$ and $\tilde{\theta}$, converge exponentially for all $t<\overline{T}$ and*

$$\overline{\Theta}\geq\left\Vert \tilde{\theta}(t)\right\Vert ,\quad\overline{X}\geq\left\Vert \tilde{x}(t)\right\Vert ,\quad\forall t\geq\overline{T}, \tag{178}$$

where $\overline{\Theta},\overline{X}\in\mathbb{R}_{\geq0}$ denote ultimate bounds for the parameter estimation errors and state estimation errors, respectively, $\tilde{\theta}:=\theta-\hat{\theta}$ and $\tilde{x}=x-\hat{x}$, where $\hat{\theta}$ and $\hat{x}$ denote estimates of the parameters and states, respectively.

For examples of such state and parameter estimators, see [66, 67]. The state and parameter estimator is implemented synchronously with inverse reinforcement learning, and in real-time. Assumption 17 also implies existence of compact sets $\hat{\mathcal{X}}\subseteq\mathbb{R}^{n}$ and $\hat{\Theta}\subseteq\mathbb{R}^{p}$, such that $\hat{x}(t)\in\hat{\mathcal{X}}$ and $\hat{\theta}(t)\in\hat{\Theta}$, $\forall t\in\mathbb{R}_{\geq0}$.

6.3 Optimal Policy Estimation

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

The closed-form nonlinear optimal policy corresponding to the reward structure in (173) is

$$u = -\frac{1}{2}R^{-1}\Big(\nabla_u f(x)\Big)^T\Big(\nabla_x V^*(x)\Big)^T. \tag{179}$$

To promote estimation, u will be represented as

$$u = -\left(W_u^*\right)^T \sigma_u\left(x\right) + \epsilon_u(x), \tag{180}$$

where $W_u^* \in \mathbb{R}^{K\times m}$ is a matrix of unknown ideal constant parameters, $\sigma_u : \mathbb{R}^n \to \mathbb{R}^K$ are known continuously differentiable features, and $\epsilon_u : \mathbb{R}^n \to \mathbb{R}^m$ is the resulting approximation error. Given any constant $\overline{\epsilon}_u$, there exist $K \in \mathbb{N}$ and $\overline{\sigma}_u \in \mathbb{R}_{>0}$ such that $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\sigma_u\left(x,u\right)\right\| < \overline{\sigma}_u$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\nabla\sigma_u\left(x,u\right)\right\| < \overline{\sigma}_u$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}$ $\left\|\epsilon_u\left(x,u\right)\right\| < \overline{\epsilon}_u$, $\sup_{(x,u)\in(\mathcal{X}\times\mathcal{U})}\left\|\nabla\epsilon_u\left(x,u\right)\right\| < \overline{\epsilon}_u$ [58, 59]. Collecting values of the state estimates and the control signals over time instances, $t_1^u(t), t_2^u(t), \cdots, t_M^u(t)$, in a history stack, denoted as $\mathcal{H}^u(t)$, (180) can be reformulated into the matrix form

$$-\Sigma_u(t) - \hat{\Sigma}_\sigma(t)\hat{W}_u = \hat{\Sigma}_\sigma(t)\tilde{W}_u - \Delta_u(t), \tag{181}$$

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

$$\Sigma_u(t) := \left[u(t_1(t)), \cdots, u(t_M(t))\right]^T,$$

$$\hat{\Sigma}_\sigma(t) := \left[\sigma_u(\hat{x}(t_1(t))), \cdots, \sigma_u(\hat{x}(t_M(t)))\right]^T,$$

the residual Δ_u depends on ϵ_u and $\tilde{x}$, and the time instances $t_1^m, \ldots, t_M^u$ are selected according to minimum singular value maximization.

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

$$\|\Delta_u(t)\| \leq \overline{\Delta}_u + L_u \bar{\tilde{x}}(t), \tag{182}$$

where $\bar{\tilde{x}}(t) = \max_{i=1,2,\ldots,M} \|\tilde{x}(t_i(t))\|$. Since $t \mapsto x(t), t \mapsto \hat{x}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 17, the bound $\bar{\Delta}_u$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{x}$ currently stored in the history stack.

The relationship in (181) suggests the following update law for estimation of the unknown weights

$$\dot{\hat{W}}_u = \alpha_u \Gamma_u(t) \hat{\Sigma}_\sigma^T(t) \left(-\Sigma_u(t) - \hat{\Sigma}_\sigma(t) \hat{W}_u \right), \tag{183}$$

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

$$\dot{\Gamma}_u = \beta_u \Gamma_u - \alpha_u \Gamma_u \hat{\Sigma}_\sigma^T(t) \hat{\Sigma}_\sigma(t) \Gamma_u, \tag{184}$$

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

The update law in (181) is motivated by the fact that the dynamics of the weight estimation error can be described by

$$\dot{\tilde{W}}_u = -\alpha_u \Gamma_u(t) \hat{\Sigma}_\sigma^T(t) \left(\hat{\Sigma}_\sigma(t) \tilde{W}_u - \Delta_u(t) \right), \tag{185}$$

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

6.4 Analysis of the Optimal Policy Estimator

Convergence of the estimation error to a neighborhood of the origin follow under the following condition on the regressor, $\hat{\Sigma}_\sigma$.

Definition 7 *The time-varying history stack, $\mathcal{H}^u$, is called full rank, uniformly in t, if there exists a $\underline{k} > 0$ such that*[3] $\forall t \in \mathbb{R}_{\geq 0}$,

$$\underline{k} < \lambda_{\min}\left\{\hat{\Sigma}_\sigma^T(t)\hat{\Sigma}_\sigma(t)\right\}. \tag{186}$$

Using arguments similar to [60, Corollary 4.3.2], it can be shown that if $\lambda_{\min}\left\{\Gamma_u^{-1}(0)\right\} > 0$, and if $\mathcal{H}^u$ is full rank, uniformly in t, then the least squares gain matrix satisfies

$$\underline{\Gamma}_u \mathrm{I}_K \leq \Gamma_u(t) \leq \overline{\Gamma}_u \mathrm{I}_K, \forall\, t \geq 0, \tag{187}$$

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

Theorem 7 *If there exists a state and parameter estimator that satisfies Assumption 17, the signal $(\hat{x}, u)$ is FI, the time instances $t_1^u, \ldots, t_M^u$ are selected using minimum singular value maximization so that $\mathcal{H}^u$ is full rank, uniformly in t, and $\mathcal{H}^u$ is refreshed using a time-based purging algorithm, then $t \mapsto \tilde{W}_u(t)$ is ultimately bounded.*

Proof. Consider the following positive definite candidate Lyapunov function

$$V_u(\tilde{W}_u, t) = \mathrm{tr}(\tilde{W}_u^T \Gamma_u^{-1}(t)\tilde{W}_u), \tag{188}$$

Using the bounds in (187), the candidate Lyapunov function satisfies

$$\frac{1}{\overline{\Gamma}_u}\left\|\tilde{W}_u\right\|^2 \leq V_u\left(\tilde{W}_u, t\right) \leq \frac{1}{\underline{\Gamma}_u}\left\|\tilde{W}_u\right\|^2. \tag{189}$$

Taking the time derivative of (188), using (184), (185), (186) and (187), along with the identity $\dot{\Gamma}_u^{-1} = -\Gamma_u^{-1}\dot{\Gamma}_u\Gamma_u^{-1}$ and using the Cauchy-Schwartz inequality, $\dot{V}_u$ can be bounded by

$$\dot{V}_u(\tilde{W}_u, t) \leq -\left(\alpha_u \underline{k} + \frac{\beta_u}{\overline{\Gamma}_u}\right)\left\|\tilde{W}_u\right\|^2 + 2\alpha_u\left\|\tilde{W}_u\right\|\left\|\hat{\Sigma}_\sigma(t)\right\|\left\|\Delta_u(t)\right\|. \tag{190}$$

Using (182), $\dot{V}_u$ can be bounded as

$$\dot{V}_u(\tilde{W}_u, t) \leq -\frac{1}{2}\left(\alpha_u \underline{k} + \frac{\beta_u}{\overline{\Gamma}_u}\right)\left\|\tilde{W}_u\right\|^2, \forall \|\tilde{W}_u\| \geq \rho\left(\|\mu\|\right), \tag{191}$$

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

where $\mu = \left[\sqrt{\overline{\Delta}_u}, \sqrt{\overline{\tilde{x}}}\right]^T$, $\rho\left(\|\mu\|\right) = \left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u}\right)\|\mu\|^2$, and $\overline{\Sigma}_\sigma$ is an upper bound of $\|\hat{\Sigma}_\sigma(t)\|$, $\forall t \geq 0$. Since $t \mapsto \hat{x}(t)$, and $t \mapsto u(t)$ are bounded by Assumption 17, the bound $\overline{\Sigma}_\sigma$ can be selected independent of t_i and the specific trajectories of u and $\hat{x}$ currently stored in the history stack. Using (189) and (191), [80, Theorem 4.19] can be invoked to conclude that (185) is input-to-state stable with state $\tilde{W}_u$ and input μ.

If a time-based purging algorithm is implemented and if the signal $(\hat{x}, u)$ is FI, there exists a time instance T_s, such that for all $t \geq T_s$, the history stack $\mathcal{H}^u(t)$ remains unchanged. As a result, using Exercise 4.58 from [80], it can be concluded that the ultimate bound on $\tilde{W}_u$ can be expressed as

$$\limsup_{t\to\infty}\|\tilde{W}_u(t)\| \leq \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}}\left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u}\right)\overline{\Delta}_u + \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}}\left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u}\right)\overline{\tilde{x}}(T_s). \quad (192)$$

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

$$\limsup_{t\to\infty}\|\tilde{W}_u(t)\| \leq \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}}\left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u}\right)\overline{\Delta}_u + \sqrt{\frac{\overline{\Gamma}_u}{\underline{\Gamma}_u}}\left(\frac{4\alpha_u \overline{\Sigma}_\sigma \max\{1, L_u\}}{\alpha_u \underline{k} + \frac{1}{\overline{\Gamma}_u}\beta_u}\right)\overline{X} := \overline{\gamma}_u. \quad (193)$$

■

Remark 7 *Theorem 7 implies existence of a compact set* $\hat{\mathcal{U}} \subseteq \mathbb{R}^m$*, such that* $\hat{u}(t) \in \hat{\mathcal{U}}$*,* $\forall t \in \mathbb{R}_{\geq 0}$.

Remark 8 *If the full state is measurable, the optimal controller estimate converges exponentially, see [139].*

6.5 Inverse Reinforcement Learning Formulation

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

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

$$\delta''\Big(x_i, \hat{u}_i, \hat{W}', \hat{\theta}(t_i)\Big) = \Big(\hat{W}'\Big)^T \sigma'\Big(x_i, \hat{u}_i, \hat{\theta}(t_i)\Big), \tag{194}$$

where

$$\hat{W}' := \Big[\hat{W}_V^T, \hat{W}_Q^T, \hat{W}_R^T\Big]^T,$$

and

$$\sigma'\Big(x_i, \hat{u}_i, \hat{\theta}(t_i)\Big) := \Big[\big(\sigma(x_i, \hat{u}_i))\big)^T \hat{\theta}(t_i)\left(\nabla_x \sigma_V(x_i)\right)^T + \big(f^o(x_i, \hat{u}_i)\big)^T \left(\nabla_x \sigma_V(x_i)\right)^T, \left(\sigma_Q(x_i)\right)^T, \left(\sigma_{R1}(\hat{u}_i)\right)^T\Big]^T.$$

Taking the first element, R_{11}, of $\hat{W}_R$ to be known, the inverse Bellman error in (194) can be expressed as

$$\delta''\Big(x_i, \hat{u}_i, \hat{W}, \hat{\theta}(t_i)\Big) = \hat{W}^T \sigma''\Big(x_i, \hat{u}_i, \hat{\theta}(t_i)\Big) + R_{11}\hat{u}_{i1}^2, \tag{195}$$

where $\hat{W} := \left[\hat{W}_V^T, \hat{W}_Q^T, \left(\hat{W}_R^{(-1)}\right)^T\right]^T$, $\hat{u}_{i1}$ denotes the first element of the vector $\hat{u}_i$, and

$$\sigma''\Big(x_i, \hat{u}_i, \hat{\theta}(t_i)\Big) := \Big[\big(\sigma(x_i, \hat{u}_i))\big)^T \hat{\theta}(t_i)\left(\nabla_x \sigma_V(x_i)\right)^T + \big(f^o(x_i, \hat{u}_i)\big)^T \left(\nabla_x \sigma_V(x_i)\right)^T, \left(\sigma_Q(x_i)\right)^T, \left(\sigma_{R1}^{(-1)}(\hat{u}_i)\right)^T\Big]^T. \tag{196}$$

A history stack, denoted as $\mathcal{H}^{IRL}$, is a set of ordered pairs of parameter estimates, $\hat{\theta}(t_i)$, and data pairs, $(x_i, \hat{u}_i)$, collected over time instance $t_1, t_2, \ldots, t_N$ into matrices $\Big(\hat{\Sigma}, \hat{\Sigma}_{R1}\Big)$. Similar to Section 4.3, the history stack here contains potentially poor estimates $\hat{u}_i$ and $\hat{\theta}(t_i)$. Since the control estimation error and the parameter estimation error both decay exponentially to an ultimate bound, a time-based purging algorithm similar to Section 4.3 is needed to remove the erroneous estimates from the history stack once newer estimates become available. As a result, the data points $(x_i, \hat{u}_i)$ and the time instance t_i are time-varying.

Utilizing estimates $\hat{\theta}(t_i)$ and data pairs $(x_i, \hat{u}_i)$ in (179), subtracting

$$0 = H\big(x_i, \left(\nabla_x V(x_i)\right)^T, u_i\big),$$

from (195), where u_i denotes the ideal value of $\hat{u}_i$, evaluating (195) and at time instances $\{t_i\}_{i=1}^N$, and stacking the results in a matrix form, we get

$$-\hat{\Sigma}(t)\hat{W} - \hat{\Sigma}_{R1}(t) = \hat{\Sigma}(t)\tilde{W} - \Delta(t), \tag{197}$$

where the weight estimation error is defined as $\tilde{W} = W^* - \hat{W}$ with $W^* := \Big[\left(W_V^*\right)^T, \left(W_Q^*\right)^T, \left(\left(W_R^*\right)^{(-1)}\right)^T \Big]^T$,

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

$$\hat{\Sigma}_{R1}(t) := \left[R_{11}\hat{u}_{11}^2(t), 2R_{11}\hat{u}_{11}(t), 0_{1\times(m-1)}, \cdots, R_{11}\hat{u}_{N1}^2(t), 2R_{11}\hat{u}_{N1}(t), 0_{1\times(m-1)}\right]^T,$$

where

$$\left(\sigma''' \left(x_i(t), \hat{u}_i(t), \hat{\theta}(t_i(t))\right)\right)^T := \begin{bmatrix} \left(\sigma'' \left(x_i(t), \hat{u}_i(t), \hat{\theta}(t_i(t))\right)\right)^T \\ \left[G\left(x_i(t), \hat{\theta}(t_i(t))\right) \quad 0_{m\times L} \quad 2\hat{\sigma}_{R2}^{(-1)}(\hat{u}_i(t)) \right] \end{bmatrix}, \tag{198}$$

$$G\left(x_i(t), \hat{\theta}(t_i(t))\right) := \left(\nabla_u f^o(x_i(t)) + \left(\hat{\theta}(t_i(t))\right)^T \nabla_u \sigma(x_i(t))\right)^T \left(\nabla_x \sigma_V(x_i(t))\right)^T, \tag{199}$$

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

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

$$\left\|\Delta(t)\right\| \le \overline{\Delta}_\epsilon + \overline{\tilde{u}}(t)\ \overline{\Delta}_{\tilde{u}} + \overline{\tilde{\theta}}(t)\ \overline{\Delta}_{\tilde{\theta}}, \tag{200}$$

where $\overline{\tilde{u}}(t) = \max_{i=1,2,\ldots,N} \|\tilde{u}_i(t)\|$ and $\overline{\tilde{\theta}}(t) = \max_{i=1,2\ldots,N} \|\tilde{\theta}(t_i(t))\|$. Since $t \mapsto x(t), t \mapsto \hat{u}(t), t \mapsto u(t)$ and $t \mapsto \hat{\theta}(t)$ are bounded by Assumption 17, the bounds $\overline{\Delta}_\epsilon, \overline{\Delta}_{\tilde{u}}$, and $\overline{\Delta}_{\tilde{\theta}}$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{u}$ currently stored in the history stack.

The relationship in (197) suggests the following update law for estimation of the unknown reward function weights

$$\dot{\hat{W}} = \alpha\Gamma(t)\hat{\Sigma}^T(t)\left(-\hat{\Sigma}(t)\hat{W} - \hat{\Sigma}_{R1}(t)\right), \tag{201}$$

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

$$\dot{\Gamma} = \beta\Gamma - \alpha\Gamma\hat{\Sigma}^T(t)\hat{\Sigma}(t)\Gamma, \tag{202}$$

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

The update law in (201) is motivated by the fact that the dynamics for the weight estimation error can be described by

$$\dot{\tilde{W}} = -\alpha\Gamma(t)\hat{\Sigma}^T(t)\left(\hat{\Sigma}(t)\tilde{W} - \Delta(t)\right), \tag{203}$$

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

6.6 Analysis of Inverse Reinforcement Learning

Using arguments similar to [60, Corollary 4.3.2], it can be shown that if $\lambda_{\min}\left\{\Gamma^{-1}(0)\right\} > 0$, and if $\mathcal{H}^{IRL}$ is full rank, uniformly in t, then the least squares gain matrix satisfies

$$\underline{\Gamma}\mathrm{I}_{L+P+m-1} \leq \Gamma(t) \leq \overline{\Gamma}\mathrm{I}_{L+P+m-1}, \forall\, t \geq 0, \tag{204}$$

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

The stability result is summarized in the following theorem.

Theorem 8 *If there exists a state and parameter estimator that satisfies Assumption 17, the signal $(\hat{x}, u)$ and sequence $(x_i, \hat{u}_i)$ are FI, the time instances $t_1^u, \ldots, t_M^u$ and $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$ are full rank, uniformly in t, and $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$ are refreshed using a time-based purging algorithm, then $t \mapsto \tilde{W}(t)$ is ultimately bounded.*

Proof. Consider the positive definite candidate Lyapunov function

$$V(\tilde{W}, t) = \frac{1}{2} \tilde{W}^T \Gamma^{-1}(t) \tilde{W}. \tag{205}$$

Using the bounds in (204), the candidate Lyapunov function satisfies

$$\frac{1}{2\overline{\Gamma}} \left\| \tilde{W} \right\|^2 \leq V\left(\tilde{W}, t\right) \leq \frac{1}{2\underline{\Gamma}} \left\| \tilde{W} \right\|^2. \tag{206}$$

Using (95), (202), (204) and (203), along with the identity $\dot{\Gamma}^{-1} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$, and using the Cauchy-Schwartz inequality, the time-derivative can be expressed as

$$\dot{V}(\tilde{W}, t) \leq -\frac{1}{2} \left(\alpha\underline{\sigma} + \frac{1}{\overline{\overline{\Gamma}}} \beta \right) \left\| \tilde{W} \right\|^2 + \alpha \|\tilde{W}\| \|\hat{\Sigma}(t)\| \|\Delta(t)\|. \tag{207}$$

Using (200), $\dot{V}$ can be bounded as

$$\dot{V}(\tilde{W}, t) \leq -\frac{1}{4} \left(\alpha\underline{\sigma} + \frac{1}{\overline{\overline{\Gamma}}} \beta \right) \left\| \tilde{W} \right\|^2, \forall \|\tilde{W}\| \geq \rho\left(\|\mu\|\right), \tag{208}$$

where $\mu = \left[\sqrt{\overline{\Delta}_\epsilon}, \sqrt{\overline{\tilde{u}}}, \sqrt{\overline{\tilde{\theta}}} \right]^T$, $\rho\left(\|\mu\|\right) = \left(\frac{4\alpha\overline{\Sigma}\max\{1, \overline{\Delta}_u, \overline{\Delta}_\theta\}}{\alpha\underline{\sigma} + \frac{1}{\overline{\overline{\Gamma}}}\beta} \right) \|\mu\|^2$, and $\overline{\Sigma}$ satisfies $\|\hat{\Sigma}(t)\| \leq \overline{\Sigma}$, $\forall t \geq 0$. Since $t \mapsto x(t), t \mapsto \hat{u}(t), t \mapsto u(t)$ and $t \mapsto \hat{\theta}(t)$ are bounded by Assumption 17, the bound $\overline{\Sigma}$ can be selected independent of t_i and the specific trajectories of x, u, and $\hat{u}$ currently stored in the history. Using (206) and (208), [80, Theorem 4.19] can be invoked to conclude that (203) is input-to-state stable with state $\tilde{W}$ and input μ.

If a time-based purging algorithm is implemented and if the signal $(\hat{x}, u)$ and sequence $(x_i, \hat{u}_i)$ are FI, there exists a time instance T_s, such that for all $t \geq T_s$, the history stacks $\mathcal{H}^u(t)$ and $\mathcal{H}^{IRL}(t)$ remain unchanged. As a result, using Exercise 4.58 from [80], it can be seen that the ultimate bound on $\tilde{W}$ can be expressed as

$$\limsup_{t\to\infty}\|\tilde{W}(t)\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_u,\overline{\Delta}_\theta}{\alpha\underline{\sigma}+\frac{1}{\overline{\Gamma}}\beta}\right)\overline{\Delta}_\epsilon \\ + \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_u,\overline{\Delta}_\theta}{\alpha\underline{\sigma}+\frac{1}{\overline{\Gamma}}\beta}\right)\left(\overline{\tilde{u}}(T_s)+\overline{\tilde{\theta}}(T_s)\right), \quad (209)$$

where $\overline{\tilde{u}}(T_s)$ denotes the bound on the control estimation error in the history stack $\mathcal{H}^{IRL}(t)$ for all $t \geq T_s$.

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

$$\limsup_{t\to\infty}\|\tilde{W}(t)\| \leq \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_u,\overline{\Delta}_\theta}{\alpha\underline{\sigma}+\frac{1}{\overline{\Gamma}}\beta}\right)\overline{\Delta}_\epsilon \\ + \sqrt{\frac{\overline{\Gamma}}{\underline{\Gamma}}}\left(\frac{4\alpha\overline{\Sigma}\max\{1,\overline{\Delta}_u,\overline{\Delta}_\theta}{\alpha\underline{\sigma}+\frac{1}{\overline{\Gamma}}\beta}\right)\left(\overline{\gamma}_u+\overline{\Theta}\right). \quad (210)$$

■

The ultimate bound for the estimation error, $\tilde{W}$, has a direct relationship to the approximation errors for both the reward function and the value function, along with the ultimate bounds for the state and parameter estimates. As such, the ultimate bound can be reduced by reducing those errors. This observation motivates the following corollary.

Corollary 4 *If $\overline{\Theta}$, $\overline{X}$, ϵ_Q, ϵ_V, and ϵ_u are zero, the signal $(\hat{x}, u)$ and sequence $(x_i, \hat{u}_i)$ are PI, the time instances $t_1^u, \ldots, t_M^u$ and $t_1, \ldots, t_N$ are selected using minimum singular value maximization so that $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$ are full rank, uniformly in t, and $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$ are refreshed using a time-based purging algorithm, then as $t \to \infty$, $\|\tilde{W}(t)\| \to 0$.*

Proof. Immediate from Theorem 8. ■

6.7 Simulation

This section presents simulations for the IRL method developed in Section 6.5. The simulation demonstrates the feedback-driven IRL method detailed in Section 6.5 to estimate the reward function when the trajectories of the system are not exciting enough to directly estimate the reward function.

6.7.1 Feedback-Driven MBIRL

In the simulation, the unknown reward and value function weights are estimated using feedback-driven MBIRL in the case where direct MBIRL using the measured data results in large reward function estimation errors. To demonstrate the performance of feedback-driven IRL, a linear optimal trajectory tracking problem with a known value function is designed using the method developed in [69, 70]. The state and parameter estimator developed in Chapter III is used to satisfy the conditions of Assumption 17.

Consider an agent with the linear dynamics

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \theta_1 & \theta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \theta_3 \end{bmatrix} u, \tag{211}$$

where the ideal values of the unknown parameters are $\theta_1 = -0.5, \theta_2 = -0.5$, and $\theta_3 = 1$.

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

$$\dot{x}_d = \begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix} x_d. \tag{212}$$

Since the agent under observation is attempting to follow a desired trajectory, the optimal control signal will likely be non-zero almost everywhere, resulting in an infinite cost. Following [70], to avoid infinite costs, it is assumed that the agent under observation solves an optimal control problem formulation to penalize an auxiliary controller, $\mu = u - u_d$, which converges to zero as the agent's controller u converges to the desired steady state control controller u_d.

The error dynamics are given by

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ -0.5 & -0.5 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu, \tag{213}$$

with the optimal control problem

$$J(e_0, \mu(\cdot)) = \int_0^\infty e(t)^T \begin{bmatrix} 1.1 & 0 \\ 0 & 3 \end{bmatrix} e(t) + 50\mu(t)^2 \, \mathrm{d}t, \tag{214}$$

where $t \mapsto e(t)$ denotes the solution of the error system in (213) under the controller $\mu(\cdot)$. The ideal reward function weights to be estimated corresponding to the optimal control problem in (214) are $Q = \operatorname{diag}([W_{Q_1},\ W_{Q_2}]) = \operatorname{diag}([1.1,\ 3])$ and $R = 50$. The steady state controller needed to track the desired trajectory is $u_d = [W_{d_1}\ W_{d_2}]\, x_d = [1.5,\ -0.5]\ x_d$. Since the objective is to estimate the reward function using measurements of x, x_d, and u, the steady-state policy u_d needs to be estimated along with the agent's dynamics.

The optimal value function to be estimated is

$$V^* = W_{V_1} e_1^2 + W_{V_2} e_2^2 + W_{V_3} e_1 e_2, \tag{215}$$

where the ideal weights are $W_{V_1} = 3.00, W_{V_2} = 4.71$, and $W_{V_3} = 2.15$. The optimal controller to be estimated is $\mu = -\left[W_{p_1},\ W_{p_2}\right] e = -[0.0215,\ 0.0942]e$. To generate an estimate of the optimal controller μ, the update law in (183) is used with the estimated state $\hat{x}$ and the known desired state x_d, found from (212) at current time t, concatenated into $\hat{\Sigma}_\sigma$. The estimated controller is then queried with random error values e_i in the set $[-5, 5]$, which produce estimates of the optimal control signal, $\hat{\mu}_i$. The pairs $(e_i, \hat{\mu}_i)$ are then iteratively collected in $\mathcal{H}^{IRL}$, and utilized to implement the feedback-driven MBIRL method in Section 6.5.

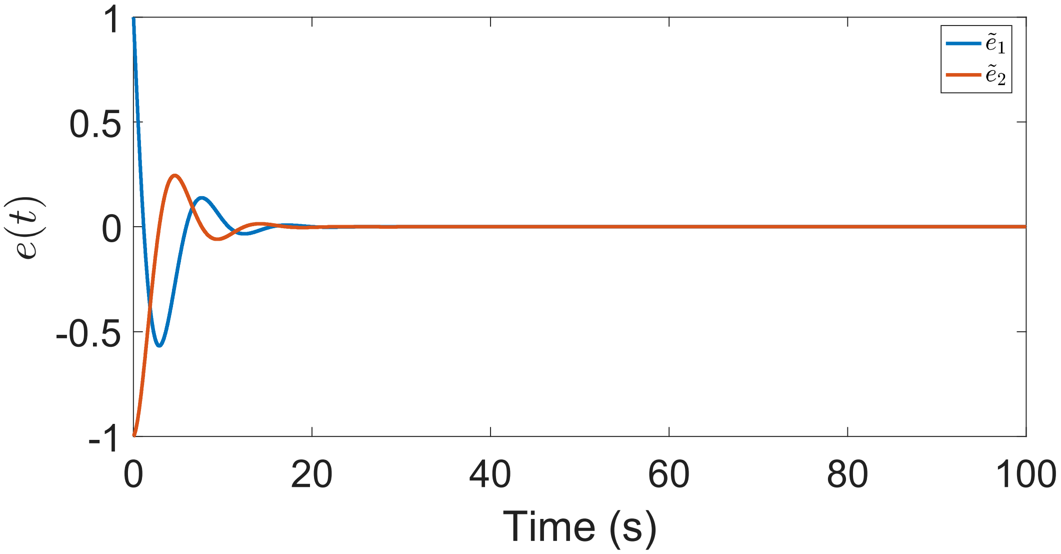


Figure 23: Trajectory tracking error corresponding to the optimal control problem in (214).

The history stacks, $\mathcal{H}^u$ and $\mathcal{H}^{IRL}$, are initialized so that all the elements in the history stacks are zero[4]. Data is added to the history stacks using a minimum singular value maximization algorithm. A time-based purging technique is utilized with $\tau = 1$. The parameters used for the two simulations are: $\beta = 0.1, \alpha = {}^{0.01}/_{M}, \beta_u = 10, \alpha_u = 4, N = 100$, $M = 10$, and a step size of $0.005s$.

First, the method developed in Chapter IV is utilized to estimate the reward function using only the trajectory. As demonstrated by Fig. 26, since the tracking errors and corresponding auxiliary controller converge to the origin within 20 sec. (see Fig. 23), the trajectories do not contain sufficient information to accurately estimate the unknown reward function using the direct MBIRL technique in Chapter IV.

As seen in Fig. 28, even though the tracking error has converged, the feedback-driven MBIRL in Section 6.5 estimates the ideal values of the reward and value functions online utilizing the synthesized estimates $\hat{\mu}_i$ (which, according to Fig. 27,

[4] It is clear from the simulation results that full rank initialization of the history stacks is a sufficient, but not a necessary condition for the analysis in Sections 6.4 and 6.6.

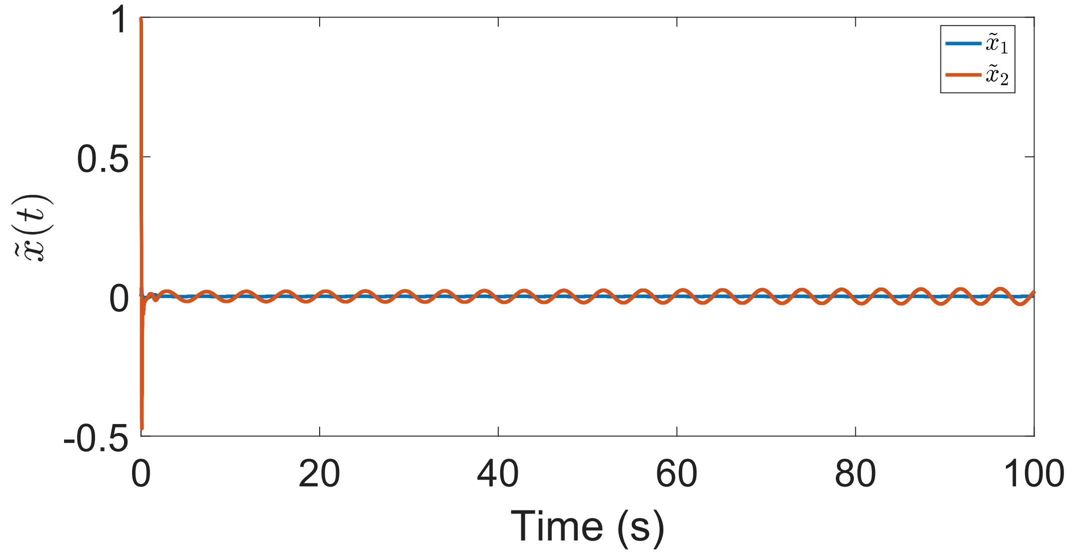


Figure 24: State estimation errors for the system in (211).

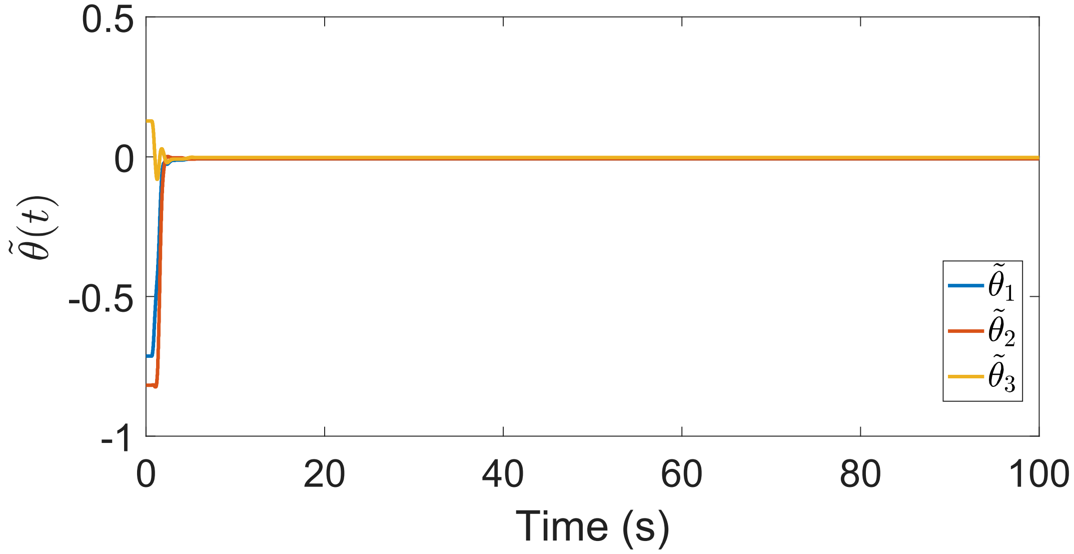


Figure 25: Parameter estimation errors for the uncertain dynamics in (211).

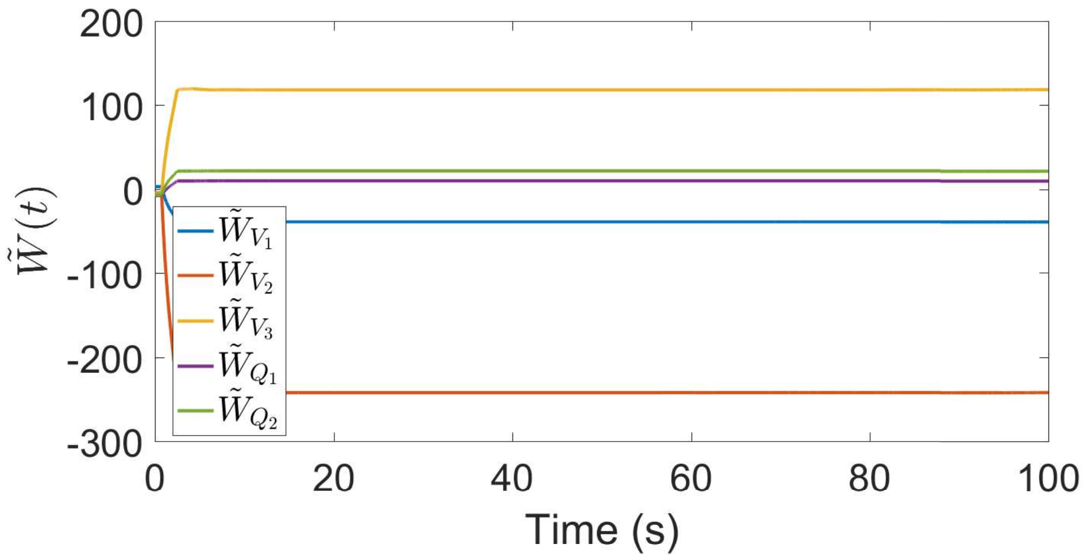


Figure 26: Reward and value function weight estimation errors using direct MBIRL in Chapter IV for the optimal control problem in (214).

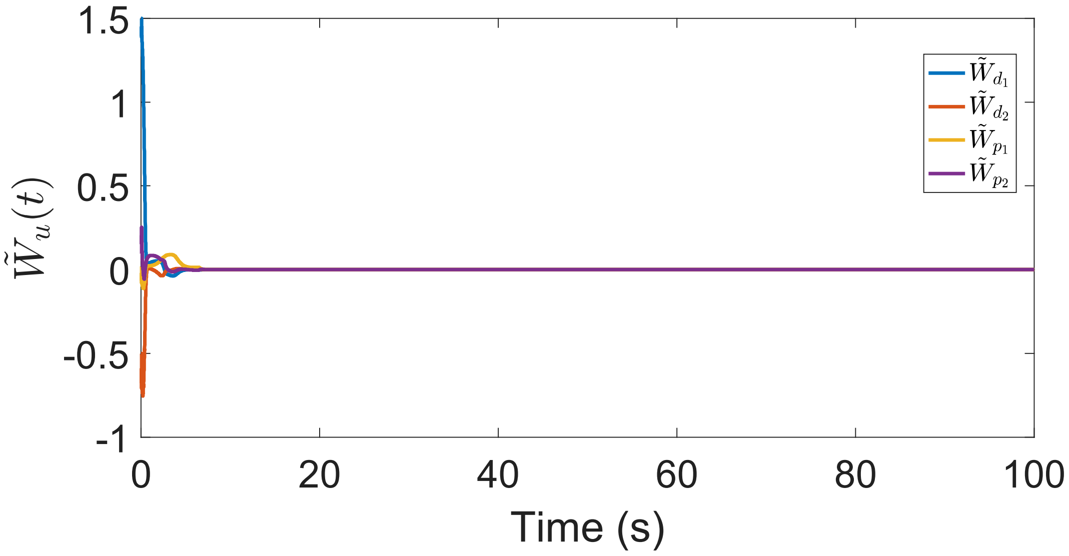


Figure 27: Control weight estimation errors for the auxiliary controller μ and the steady state desired controller u_d for the optimal control problem in (214).

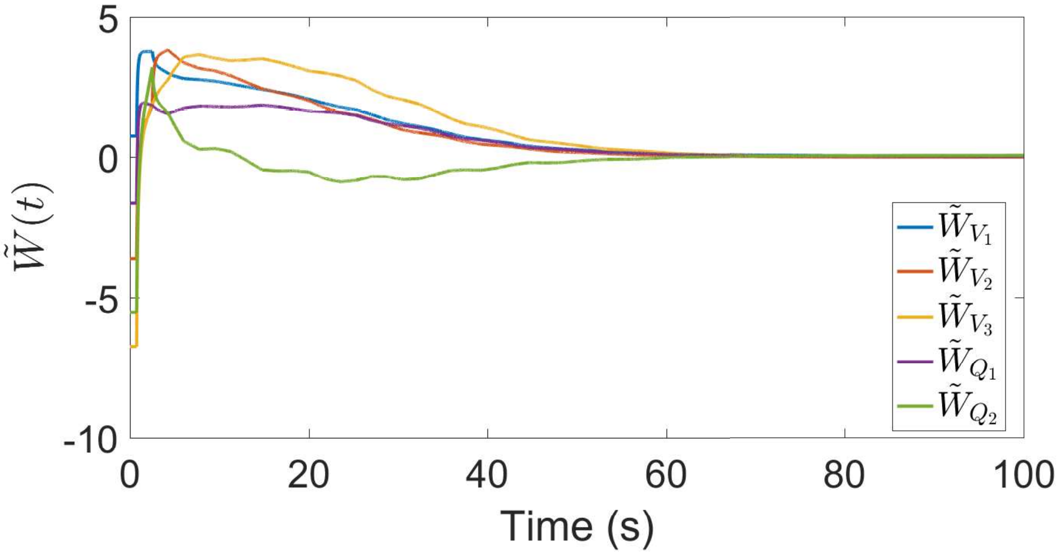


Figure 28: Reward and value function weight estimation errors using feedback-driven MBIRL in Section 6.5 for the optimal control problem in (214).

converge to the true policy, μ_i), while direct MBIRL in Chapter IV has large estimation errors (see Fig. 26).

6.8 Conclusion

In this chapter, an online model-based IRL method is developed that facilitate reward function estimation utilizing a single demonstration. Since a large majority of optimal control problems are aimed at driving a state to a set-point or an error signal to zero, single demonstrations may not provide sufficient excitation to directly estimate the reward function from only measured data. Therefore, the developed method in this chapter utilizes an estimated policy to synthetically create additional data that aims to represent the system under observation.

Chapter VII

OBSERVER BASED INVERSE REINFORCEMENT LEARNING

Real-time inverse reinforcement learning utilizing a single demonstration has been discussed in this dissertation. However, the previous chapters present work under the idea that the trajectory measurements are provided without noise, which is generally never the case in real-world applications. In the limited data context that is real-time inverse reinforcement learning, each data point becomes increasingly vital in order to uncover the unknown reward function. Yet, if the data is corrupt or noisy, extracting the best information is an unique challenge. This chapter aims to resolve the issue for real-time IRL utilizing noisy trajectory measurements.

7.1 Introduction

Inspired by recent results in online Reinforcement Learning methods [75,149,151], IRL has been extended to online implementations where the objective is to learn from a single demonstration or trajectory [68,110,138,139]. In [68,138], batch IRL techniques are developed to estimate reward functions in the presence of unmeasureable system states and/or uncertain dynamics for both linear and nonlinear systems. The case where the trajectories being monitored are suboptimal due to an external disturbance is addressed in [137] and Chapter V, and [139] and Chapter VI estimates a feedback policy and generates artificial data using the estimated policy to compensate for the sparsity of data in online implementations. However, results such as [68,110,137–139], either require full state feedback, or rely on state estimators that require dynamical systems in Brunovsky Canonical form. In addition, none of the aforementioned online

IRL methods address uncertainty in the state and control measurements.

This chapter builds on the authors' previous work in [137, 139] and Chapters IV-VI, where concurrent learning (CL) update laws are utilized to estimate reward functions online using output feedback. However, the dynamical systems in [137,139] are required to be in Brunovsky canonical form, and as such, only the output feedback case where the state is comprised of the output and its derivatives is addressed. In contrast, the IRL observer (IRL-O) technique in this chapter generalizes to any observable linear system, since the developed IRL-Os are in a standard observer form where the state estimates are modified based on the innovation (i.e., the error between the actual and the estimated output). As a result, in the case of noisy measurements, they can be implemented as Kalman filters by using the Kalman gain, instead of the developed Lyapunov-based gain design, to select the observer gain. While stability of the filters in the case where the measurements are noisy is not studied in this chapter, simulation results demonstrate that the IRL-Os utilizing both the Lyapunov-based gains and the Kalman filter gain are robust to measurement noise.

This chapter details two IRL-O formulations. The first method, called the IRL memoryless observer (MLO), is similar to a standard Luenberger observer with a modified observer gain, and guarantees parameter convergence under a persistence of excitation (PE) condition. The second observer implements a novel idea of re-using previous system state estimates and control measurements, along with the Hamilton-Jacobi-Bellman equation, to gain insights into the quality of the current estimate of the reward function. The key advantage of the IRL history stack observer (HSO) over MLO is that it provides an additional guarantee for boundedness of the estimation errors under *finite* (as opposed to *persistent*) excitation [131].

The chapter is organized as follows. Section 7.2 formulates the problem to be solved. Section 7.3 develops the innovation terms that are used in the developed IRL-O techniques using the theory of linear quadratic optimal control and details the formulation of the IRL problem as a state estimaton problem. The MLO and the HSO are designed in sections 7.4 and 7.5, respectively, along with the corresponding theoretical guarantees. Section 7.6 presents simulation results and Section 7.7 concludes the chapter.

7.2 Problem formulation

Consider an agent under observation with the following linear dynamics

$$\dot{x} = Ax + Bu, \quad y' = Cx, \tag{216}$$

where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is the state, $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ is the control, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are constant system matrices, $y' \in \mathbb{R}^L$ are the outputs, and $C \in \mathbb{R}^{L \times n}$ denotes the output matrix[1].

The agent under observation is using the policy which minimizes the following performance index

$$J(x_0, u(\cdot)) = \int_0^\infty \Big(x(t)^T Q x(t) + u(t)^T R u(t) \Big) \, \mathrm{d}t, \tag{217}$$

where $x(\cdot; x_0, u(\cdot))$ is the trajectory of the agent generated by the optimal control signal $u(\cdot)$ starting from the initial condition x_0. The objective of this chapter is to estimate the unknown matrices Q and R by utilizing noisy input-output pairs.

Remark 9 *Since Q and R can be selected to be symmetric without loss of generality, the developed IRL method only estimates the elements of Q and R that are on and above the main diagonal.*

7.3 Inverse Reinforcement Learning

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

[1] For $a \in \mathbb{R}$, the notation $\mathbb{R}_{\geq a}$ denotes the interval $[a, \infty)$ and the notation $\mathbb{R}_{>a}$ denotes the interval (a, ∞).

tion [95]

$$H\left(x\left(t\right),\nabla_{x}\left(V^{*}\left(x\left(t\right)\right)\right)^{T},u\left(t\right)\right)=0,\forall t\in\mathbb{R}_{\geq0},\tag{218}$$

and the optimal control equation

$$u(x(t))=-\frac{1}{2}R^{-1}B^{T}\nabla_{x}\left(V^{*}\left(x\left(t\right)\right)\right)^{T},\tag{219}$$

where $V^{*}:\mathbb{R}^{n}\to\mathbb{R}$ is the unknown optimal value function and $H:\mathbb{R}^{n}\times\mathbb{R}^{n}\times\mathbb{R}^{m}\to\mathbb{R}$ is the Hamiltonian, defined as $H(x,p,u):=p^{T}\left(Ax+Bu\right)+x^{T}Qx+u^{T}Ru$. Given a solution S of the Algebraic Riccati Equation, the optimal value function can be calculated as $V^{*}(x)=x^{T}Sx$.

To aid in the estimation of the reward function, note that V^{*}, $x^{T}Qx$, and $u^{T}Ru$ can be linearly parameterized as $V^{*}(x)=\left(W_{V}^{*}\right)^{T}\sigma_{V}(x)$, $x^{T}Qx=\left(W_{Q}^{*}\right)^{T}\sigma_{Q}(x)$, and $u^{T}Ru=\left(W_{R}^{*}\right)^{T}\sigma_{R1}(u)$, respectively, where $\sigma_{V}(x):\mathbb{R}^{n}\to\mathbb{R}^{P}$, $\sigma_{Q}(x):\mathbb{R}^{n}\to\mathbb{R}^{P}$, and $\sigma_{R1}(u):\mathbb{R}^{m}\to\mathbb{R}^{M}$, are the basis functions, selected as

$$\begin{aligned}\sigma_{V}(x)=\sigma_{Q}(x) & :=[x_{1}^{2},2x_{1}x_{2},2x_{1}x_{3},\ldots,2x_{1}x_{n},x_{2}^{2},\\ & \qquad 2x_{2}x_{3},2x_{2}x_{4},\ldots,x_{n-1}^{2},\ldots,2x_{n-1}x_{n},x_{n}^{2}]^{T},\\ \sigma_{R1}(u) & :=[u_{1}^{2},2u_{1}u_{2},2u_{1}u_{3},\ldots,2u_{1}u_{m},u_{2}^{2},\\ & \qquad 2u_{2}u_{3},2u_{2}u_{4},\ldots,u_{m-1}^{2},\ldots,2u_{m-1}u_{m},u_{m}^{2}]^{T},\end{aligned}$$

and $W_{V}^{*}\in\mathbb{R}^{P}$, $W_{Q}^{*}\in\mathbb{R}^{P}$, and $W_{R}^{*}\in\mathbb{R}^{M}$, are the ideal weights, given by

$$\begin{aligned}W_{V}^{*} & =\left[S_{11},2S_{1}^{(-1)},S_{22},2S_{2}^{(-2)},\ldots,2S_{n-1}^{-(n-1)},S_{nn}\right]^{T},\\ W_{Q}^{*} & =\left[Q_{11},2Q_{1}^{(-1)},Q_{22},2Q_{2}^{(-2)},\ldots,2Q_{n-1}^{-(n-1)},Q_{nn}\right]^{T},\\ W_{R}^{*} & =\left[R_{11},2R_{1}^{(-1)},R_{22},2R_{2}^{(-2)},\ldots,2R_{m-1}^{-(m-1)},R_{mm}\right]^{T},\end{aligned}$$

where, for a given matrix $E\in\mathbb{R}^{n\times n}$, E_{ij} denotes the corresponding element in the i-th row and the j-th column of the matrix E, and $E_{i}^{(-j)}$ denotes the i-th row of the matrix E with the first j elements removed, i.e., $E_{3}^{(-3)}:=\left[E_{34},E_{35},\ldots,E_{3(n-1)},E_{3n}\right]$.

Using $\hat{W}_V$, $\hat{W}_Q$, and $\hat{W}_R$, which are the estimates of W_V^*, W_Q^*, and W_R^*, respectively, in (218), the inverse Bellman error (IBE) $\delta' : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{2P+M} \to \mathbb{R}$ is obtained as

$$\delta'\left(x, u, \hat{W}'\right) = \hat{W}_V^T \nabla_x \sigma_V(x)\left(Ax + Bu\right) + \hat{W}_Q^T \sigma_Q(x) + \hat{W}_R^T \sigma_{R1}(u), \tag{220}$$

where $\hat{W}' := \begin{bmatrix} \hat{W}_V^T & \hat{W}_Q^T & \hat{W}_R^T \end{bmatrix}^T$.

Utilizing $2Ru = -B^T \nabla_x \left(V^*(x)\right)^T$, Ru can be linearly parameterized as $Ru = \sigma_{R2}(u) W_R^*$, where W_R^* is as previously defined in the IBE and $\sigma_{R2}(u) : \mathbb{R}^m \to \mathbb{R}^{m \times M}$, where the features $\sigma_{R2}(u)$ can be explicitly calculated as

$$\sigma_{R2}(u) = \begin{bmatrix} u^T & 0_{1\times m-1} & \dots & 0 \\ 0_{1\times m} & \left(u^{(-1)}\right)^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1\times m} & 0_{1\times m-1} & \dots & \left(u^{-(m-1)}\right)^T \end{bmatrix}, \tag{221}$$

where for a given vector $u \in \mathbb{R}^{1\times m}$, $u^{(-j)}$ denotes the vector u with the first j elements removed. Using $\hat{W}_R$ and $\hat{W}_V$ in the optimal controller equation for W_R^* and W_V^*, respectively, after rearranging, a control residual error $\Delta_u' : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{2P+M} \to \mathbb{R}^m$ is obtained as

$$\Delta_u'(x, u, \hat{W}') = B^T \left(\nabla_x \sigma_V(x)\right)^T \hat{W}_V + 2\sigma_{R2}(u)\hat{W}_R.$$

Augmenting the control residual error and the inverse Bellman error yields the error equation

$$\begin{bmatrix} \delta'\left(x, u, \hat{W}'\right) \\ \Delta_u'\left(x, u, \hat{W}'\right) \end{bmatrix} = \begin{bmatrix} \sigma_{\delta'}(x, u) \\ \sigma_{\Delta_u'}(x, u) \end{bmatrix} \begin{bmatrix} \hat{W}_V \\ \hat{W}_Q \\ \hat{W}_R \end{bmatrix}, \tag{222}$$

where

$$\sigma_{\delta'}(x, u) = \Big[(Ax + Bu)^T \left(\nabla_x \sigma_V(x)\right)^T, \sigma_Q(x)^T, \sigma_{R1}(u)^T \Big],$$

and

$$\sigma_{\Delta_u'}(x, u) = \Big[B^T \left(\nabla_x \sigma_V(x)\right)^T, 0_{m\times n}, 2\sigma_{R2}(u) \Big].$$

The IRL problem is then formulated as the need to estimate $\hat{W}_V, \hat{W}_Q$, and $\hat{W}_R$ by minimizing δ' and Δ_u'. However, the IRL problem, as formulated above, is ill-posed, because the minimization problem $\min_{\hat{W}'} |\delta'| + \|\Delta_u'\|$ admits an infinite number of solutions, including the trivial solution $\hat{W}_V = \hat{W}_Q = \hat{W}_R = 0$ and the scaled solutions $\hat{W}_V = \alpha W_V^*, \hat{W}_Q = \alpha W_Q^*$, and $\hat{W}_R = \alpha W_R^*$ $\forall \alpha \in \mathbb{R}_{>0}$. To address the scaling ambiguity and to remove the trivial solution, a single reward weight will be assumed to be known. Since the optimal solution corresponding to a cost function is invariant with respect to arbitrary scaling of the cost function, establishing the scale by assuming that one of the weights as known is without loss of generality. Selecting r_1 as the known weight and removing it from (222) yields

$$\begin{bmatrix} \delta\left(x, u, \hat{W}\right) \\ \Delta_u\left(x, u, \hat{W}\right) \end{bmatrix} = \begin{bmatrix} \sigma_\delta\left(x, u\right) \\ \sigma_{\Delta_u}\left(x, u\right) \end{bmatrix} \begin{bmatrix} \hat{W}_V \\ \hat{W}_Q \\ \hat{W}_R^- \end{bmatrix} + \begin{bmatrix} u_1^2 r_1 \\ 2u_1 r_1 \\ 0_{m-1\times 1} \end{bmatrix}, \tag{223}$$

where $\hat{W}_R^-$ denotes $\hat{W}_R$ with the first element removed, $\hat{W} := \begin{bmatrix} \hat{W}_V^T & \hat{W}_Q^T & \left(\hat{W}_R^-\right)^T \end{bmatrix}^T$,

$$\sigma_\delta\left(x, u\right) = \left[\, (Ax + Bu)^T \left(\nabla_x \sigma_V(x)\right)^T, \sigma_Q(x)^T, \left(\sigma_{R1}^-(u)\right)^T \,\right],$$

and

$$\sigma_{\Delta_u}\left(x, u\right) = \begin{bmatrix} B^T \left(\nabla_x \sigma_V(x)\right)^T & 0_{m\times n} & 2\sigma_{R2}^-(u) \end{bmatrix},$$

where $\left(\sigma_{R1}^-(u)\right)^T$ and $\sigma_{R2}^-(u)$ denote $\sigma_{R1}^T(u)$ and $\sigma_{R2}(u)$ with the first columns removed.

We can formulate the IRL problem as a state estimation problem by utilizing the IBE and the controller equation in an observer framework. Such a formulation allows us to address general output feedback linear systems and to leverage the use of Kalman gains under noisy conditions.

To cast the IRL problem in a state estimation form, the ideal weights are concatenated with the system state to yield the concatenated state vector $z = \begin{bmatrix} x^T & (W^*)^T \end{bmatrix}^T$, where $W^* := \begin{bmatrix}\left(W_V^*\right)^T & \left(W_Q^*\right)^T & \left(\left(W_R^*\right)^-\right)^T\end{bmatrix}^T$. Since the ideal weights are constant, the dynamics of the concatenated state is expressed as

$$\dot{z} = \begin{bmatrix} Ax + Bu \\ 0_{2P+M-1\times 1} \end{bmatrix},$$

and $y = h(z)$, where y denotes the measurement vector and $h(z)$ is the corresponding measurement model to be designed in the following.

7.4 A memoryless observer

The key idea behind MLO is to treat the measurements, y', and the measured/known quantities in (223) as the *output*, $y \in \mathbb{R}^{L+1+m}$, used for estimation of the concatenated state. The output is thus given by

$$y = \begin{bmatrix} (y')^T & -u_1^2 r_1 & -2u_1 r_1 & 0_{1\times m-1} \end{bmatrix}^T.$$

The corresponding measurement model is developed by using (223) to express the output as a function of the concatenated state as

$$h(z) = \begin{bmatrix} Cx \\ \begin{bmatrix} \sigma_\delta\left(x, u\right) \\ \sigma_{\Delta_u}\left(x, u\right) \end{bmatrix} \begin{bmatrix} W_V^* \\ W_Q^* \\ \left(W_R^*\right)^- \end{bmatrix} \end{bmatrix}.$$

Let $g(\hat{x}, u) := \begin{bmatrix} \sigma_\delta\left(\hat{x}, u\right) \\ \sigma_{\Delta_u}\left(\hat{x}, u\right) \end{bmatrix}$ and $\sigma_u(u_1) := \begin{bmatrix} -u_1^2 r_1 \\ -2u_1 r_1 \\ 0_{m-1\times 1} \end{bmatrix}$. The observer can then be designed as

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{W}} \end{bmatrix} = \begin{bmatrix} A\hat{x} + Bu \\ 0_{2P+M-1\times 1} \end{bmatrix} + K\left(\begin{bmatrix} Cx \\ \sigma_u(u_1) \end{bmatrix} - \begin{bmatrix} C\hat{x} \\ g(\hat{x}, u)\hat{W} \end{bmatrix}\right), \tag{224}$$

where $K \in \mathbb{R}^{n+2P+M-1 \times L+m+1}$ is the observer gain matrix, designed in the following section.

7.4.1 Observer Gain Design and Stability Analysis

In the following analysis, the gain matrix K will be designed in a block diagonal form. In particular, we choose

$$K_{MLO} := \begin{bmatrix} K_1 & 0_{n \times 1+m} \\ 0_{2P+M-1 \times L} & \gamma g(\hat{x}, u)^T K_2 \end{bmatrix}$$

and $\gamma := 1/\left(\nu \| g(\hat{x},u)^T g(\hat{x},u) \| + 1\right)$ where $\nu \in \mathbb{R}_{\geq 0}$ is a tunable constant.

The following theorem analyzes the stability properties of the resulting MLO using persistence of excitation.

Definition 1 *A signal $t \mapsto A(t)$ is called persistently excited, if for all $t \geq 0$ there exists $\alpha_1, \alpha_2, \delta \in \mathbb{R}_{>0}$ such that*[2] $\alpha_2 I \geq \int_{t_0}^{t_0+\delta} A(\tau)\, \mathrm{d}\tau \geq \alpha_1 I$.

Theorem 9 *Provided the gain K_1 is selected such that $(A - K_1 C)$ is Hurwitz, the gain K_2 is selected to be a symmetric positive definite matrix, and $g(\hat{x}, u)$ is PE, then* $\lim_{t \to \infty} \tilde{W}(t) = 0$.

Proof. The dynamics for the system state estimation errors can be described by $\dot{\tilde{x}} = Ax + Bu - A\hat{x} - Bu - K_1 C\tilde{x} = \dot{\tilde{x}} = (A - K_1 C)\tilde{x}$. If $A - K_1 C$ is Hurwitz, then $\tilde{x}$ converges exponentially to the origin.

The dynamics of the weight estimation error can be expressed as

$$\dot{\tilde{W}} = -\gamma g(\hat{x}, u)^T K_2 \sigma_u(u_1) + \gamma g(\hat{x}, u)^T K_2 g(\hat{x}, u)\hat{W}.$$

Adding $\pm\gamma g(\hat{x}, u)^T K_2 g(\hat{x}, u) W^*$ and using the fact that $\sigma_u(u_1) = g(x, u)W^*$, the weight estimation error dynamics can be expressed as a perturbed linear time-varying

[2] The notation I denotes an identity matrix.

system

$$\dot{\tilde{W}} = -A(t)\tilde{W} + B(t), \tag{225}$$

where

$$A(t) := \gamma(t) g(\hat{x}(t), u(t))^T K_2 g(\hat{x}(t), u(t)),$$

and

$$B(t) := \gamma(t) g(\hat{x}(t), u(t))^T K_2 (g(\hat{x}(t), u(t)) - g(x(t), u(t))) W^*.$$

Since $\hat{x}, x, u \in \mathcal{L}_\infty$, Theorem 2.5.1 from [134] implies that the nominal system $\dot{\tilde{W}} = -A(t)\tilde{W}$ is globally exponentially stable (GES) if K_2 is a symmetric positive definite matrix and the signal $(\hat{x}, u)$ is PE.

Lemma 4.6 from [80] can then be invoked with $B(t)$ as the input and $\tilde{W}$ as the state to conclude that (225) is input-to-state stable (ISS). Furthermore, as $t \to \infty$, $\tilde{x}(t) \to 0$, and as a result, $B(t) \to 0$. Exercise 4.58 in [80] can then be invoked to conclude that $\lim_{t\to\infty} \tilde{W}(t) = 0$. ■

7.5 Inclusion of memory

The observer designed in the previous section relies on *persistent* excitation for stability and convergence. As a result, it suffers from the well-known lack of robustness of PE-based adaptive control methods under loss of excitation. This section develops an observer (called the HSO) that relies on re-use of previously recorded data (henceforth referred to as the history stack) for robustness. If the system trajectories are PE, then the HSO results in convergence of the estimation errors to the origin, similar to the MLO. However, as opposed to the MLO, through the use of a history stack, the HSO guarantees boundedness of the state estimation errors even under loss of excitation.

The output for the HSO is

$$y(t) = \left[\left(y'(t)\right)^T, -u_1^2(t_1) r_1, -2u_1(t_1) r_1, 0_{1\times m-1}, \ldots, -u_1^2(t_N) r_1, -2u_1(t_N) r_1, 0_{1\times m-1} \right]^T,$$

with the corresponding measurement model, obtained by using past control values and past state estimates in (223), given by

$$h(z) = \begin{bmatrix} Cx \\ \begin{bmatrix} \sigma_\delta\left(x(t_1), u(t_1)\right) \\ \sigma_{\Delta_u}\left(x(t_1), u(t_1)\right) \\ \vdots \\ \sigma_\delta\left(x(t_N), u(t_N)\right) \\ \sigma_{\Delta_u}\left(x(t_N), u(t_N)\right) \end{bmatrix} \begin{bmatrix} W_V^* \\ W_Q^* \\ \left(W_R^*\right)^- \end{bmatrix} \end{bmatrix}, \tag{226}$$

where $\sigma_\delta\left(x(t_i), u(t_i)\right)$ and $\sigma_{\Delta_u}\left(x(t_i), u(t_i)\right)$ denotes $\sigma_\delta\left(x(t), u(t)\right)$ and $\sigma_{\Delta_u}\left(x(t), u(t)\right)$ evaluated at time t_i, respectively.

It is assumed that at every time instance t, the observer has access to a history stack $\mathcal{H} := \left\{\hat{\Sigma}, \Sigma_u\right\}$, defined as

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

where time instances $t_1, \ldots, t_N$ are selected to ensure that the resulting history stack is full rank, as subsequently defined in Def. 2. Denoting the observer gain matrix by $K \in \mathbb{R}^{n+2P+M-1\times L+N(1+m)}$, the HSO is designed as

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{W}} \end{bmatrix} = \begin{bmatrix} A\hat{x} + Bu \\ 0_{2P+M-1} \end{bmatrix} + K\left(\begin{bmatrix} Cx \\ \Sigma_u \end{bmatrix} - \begin{bmatrix} C\hat{x} \\ \hat{\Sigma}\hat{W} \end{bmatrix}\right). \tag{227}$$

The error in equation (222) implies that the innovation $\Sigma_u - \hat{\Sigma}\hat{W}$ in (227) corresponds to the weight estimation error $\tilde{W}$ only if $\hat{\Sigma} = \Sigma$. Since $\hat{\Sigma}$ depends continuously on $\hat{x}$ and because $\hat{x}$ exponentially converges to x, $\hat{\Sigma}$ exponentially converges to Σ. As a result, newer and better estimates of x can be leveraged to improve the estimates of W^* by purging and refreshing the history stack $\mathcal{H}$. Due to purging, the time instances $\{t_1, \cdots t_N\}$ and the matrices $\hat{\Sigma}$ and Σ_u are piecewise constant functions of time.

Definition 2 *The history stack is called* full rank *if rank* $\left(\hat{\Sigma}\right) = 2P + M - 1$*.*

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

A history stack management algorithm similar to [68, Fig. 1] is used to ensure the existence of a time instance t_M such that, if the signal $(\hat{x}, u)$ is FI, then the history stack is full rank for all $t \geq t_M$, and in addition, if it is PI, then $\lim_{t\to\infty} \left\|\Sigma(t) - \hat{\Sigma}(t)\right\| = 0$.

7.5.1 Observer Gain Design and Stability Analysis

The HSO gain matrix is designed in the block diagonal form

$$K_{HSO} := \begin{bmatrix} K_3 & 0_{n\times N+Nm} \\ 0_{2P+M-1\times L} & K_4 \left(\hat{\Sigma}^T\hat{\Sigma}\right)^{-1} \hat{\Sigma}^T \end{bmatrix},$$

where $K_3 \in \mathbb{R}^{n\times L}$ is a constant gain matrix and $K_4 : \mathbb{R}_{\geq 0} \to \mathbb{R}^{2P+M-1\times 2P+M-1}$ is a potentially time-varying gain matrix. Provided the gain matrices are selected to satisfy the hypothesis of Theorem 10 below, the resulting observer in (227) can be shown to be convergent in the presence of PE and bounded under loss of excitation. Finite excitation is needed for the history stack to be full rank so that $(\hat{\Sigma}^T\hat{\Sigma})^{-1}$ is well-defined.

Theorem 10 *Provided K_3 is selected such that $(A - K_3 C)$ is Hurwitz, $K_4(t)$ is selected such that for $t < t_M$, $K_4(t) = 0$ and for $t \geq t_M$, $K_4(t)$ is symmetric positive definite, $0 < \underline{k} \leq \inf_{t \geq t_M} \left\{\lambda_{\min} K_4(t)\right\}$ and $\sup_{t \geq t_M} \left\{\left\|K_4(t)\right\|\right\} \leq \overline{k} < \infty$, then $\tilde{W}$ is ultimately bounded (UB) if the signal $(\hat{x}, u)$ is FI and $\lim_{t \to \infty} \tilde{W}(t) = 0$ if it is PI.*

Proof. Using Theorem 9, if $(A - K_3 C)$ is Hurwitz, $\tilde{x}(t) \to 0$ exponentially as $t \to \infty$. Using (227), the dynamics of the weight estimation error can be expressed as

$$\dot{\tilde{W}} = K_4(t)\hat{W} - K_4(t)\left(\hat{\Sigma}^T(t)\hat{\Sigma}(t)\right)^{-1}\hat{\Sigma}^T(t)\Sigma_u(t).$$

Since K_4 is set to 0, the weight estimates are constant over $[0, t_M)$. For $t \geq t_M$, adding $\pm K_4(t)W^*$ to $\dot{\tilde{W}}$, and using the fact that $\Sigma W^* = \Sigma_u$, the weight estimation error dynamics can be treated as the controlled system

$$\dot{\tilde{W}} = -K_4(t)\tilde{W} + K_4(t)w, \tag{228}$$

where $w(t) := \left(I - \left(\hat{\Sigma}^T(t)\hat{\Sigma}(t)\right)^{-1}\hat{\Sigma}^T(t)\Sigma(t)\right)W^*$ is treated as the control input. Using the Cauchy-Schwartz Inequality and the Rayleigh-Ritz Theorem [57], the orbital derivative of the positive definite candidate Lyapunov function $V\left(\tilde{W}\right) := \frac{1}{2}\tilde{W}^T\tilde{W}$ along the trajectories of (228) can be bounded as

$$\dot{V}\left(t, \tilde{W}\right) \leq -\underline{k}\left\|\tilde{W}\right\|^2 + \overline{k}\left\|\tilde{W}\right\|\|w\|, \quad \forall t \geq t_M, \tag{229}$$

and $\tilde{W} \in \mathbb{R}^{2P+M-1}$.

In the domain $\|\tilde{W}\| > \frac{2\overline{k}\|W^*\|}{\underline{k}}\|w\|$, the orbital derivative satisfies the bound $\dot{V}\left(t, \tilde{W}\right) \leq -\frac{k}{2}\|\tilde{W}\|^2$. Using Theorem 4.19 from [80], it can be concluded that the controlled system in (228) is input-to-state stable (ISS).

If the signal $(\hat{x}, u)$ is PI, then the history stack can be purged and refreshed infinitely many times such that $w(t) \to 0$ as $t \to \infty$. Utilizing Exercise 4.58 from [80], it can then be concluded that $\tilde{W}(t) \to 0$ as $t \to \infty$.

If the signal $(\hat{x}, u)$ is FI but not PI, then there exists a time instance T such that the history stack remains unchanged for all $t \geq T$. As a result, there exists a constant $\overline{w}$ such that for all $t \geq T$, $\left\|w(t)\right\| \leq \overline{w}$. By the definition of ISS, it can then be concluded that $\tilde{W}$ is UB. ■

Remark 10 *The UB result in the absence of PE is a distinct advantage of HSO over MLO, which provides no such guarantee. Once the system states are no longer exciting, the MLO could potentially become unstable.*

Remark 11 *The IRL-O formulation is not restricted to the choices of K in Theorems 9 and 10. Different stabilizing or heuristic gain selection methods can be incorporated in the developed framework. For example, motivated by robustness to measurement noise, the use of a Kalman filter for gain selection is explored in Section 7.6.*

7.6 Simulations

A key motivation for casting the IRL problem into the observer framework is that the observer can be extended to a Kalman filter in a straightforward fashion to address measurement noise. To implement the developed observers as Kalman filters, all that is needed is to select the gains K_3 and K_4 using the Kalman gain update equations. The following simulation study demonstrates the validity, the robustness, and the performance of the designed observers and their Kalman filter implementation.

While the developed observer IRL methods are applicable to general output feedback linear systems, the concurrent learning (CL) method used for comparison is only applicable to a restricted set of systems (the state estimator in [66] is modified slightly for the non-Brunovsky form of (230)). In the following, to make comparisons feasible, a system that both methods are applicable to is selected.

The agent under observation has linear dynamics

$$\dot{x} = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} x + \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} u, \; y = \begin{bmatrix} 1 & 0 \end{bmatrix} x. \tag{230}$$

The optimal controller, $u^*(x) = -\begin{bmatrix} 4.14 & 5.53 \end{bmatrix} x$, minimizes an LQR problem, $Q = \operatorname{diag}([2, 11])$ and $R = 1.5$, with an optimal value function

$$V^*(x) = 2.54x_1^2 + 7.59x_2^2 + 4.50x_1x_2.$$

The ideal weights that are to be estimated are $W_{V1}^* = 2.54$, $W_{V2}^* = 7.59$, $W_{V3}^* = 4.50$, $W_{Q1}^* = 2$, $W_{Q2}^* = 11$, and $R = 1.5$ is selected as the known value to remove the scaling ambiguity.

Since the system state estimates converge exponentially to the true system states, a time based purging technique similar to [68, Fig. 1] is utilized to reduce the estimation error associated with the system state estimates stored in the history stack. Furthermore, to improve numerical stability of gain computation, the history stack management algorithm also attempts to minimize the condition number of $\hat{\Sigma}^T\hat{\Sigma}$. In the presented simulation studies, the history stacks contain data for five previous time instances and are purged every 0.5 seconds if they can be repopulated.

Three simulation studies are performed. The first shows the performance of the designed observers in a noise-free setting for a system with two states and one control. The second simulations shows the designed observers in a noise-free setting for a lager dimensional system. The last simulation incorporates noise in order to investigate the observers/filter robustness.

The error metric used to compare all of the observers/filters is the summation of the five relative weight estimation errors, defined as

$$\sum \frac{\tilde{W}_i}{W_i^*} := \frac{\|\tilde{W}_{V_1}\|}{W_{V_1}^*} + \frac{\|\tilde{W}_{V_2}\|}{W_{V_2}^*} + \frac{\|\tilde{W}_{V_3}\|}{W_{V_3}^*} + \frac{\|\tilde{W}_{Q_1}\|}{W_{Q_1}^*} + \frac{\|\tilde{W}_{Q_2}\|}{W_{Q_2}^*}.$$

7.6.1 Persistently Excited Signal without Noise - Two State System

The first simulation study concerns a noise-free environment. The controller that the agent under observation implements is a combination of the optimal controller, u^*, and a known additive excitation signal, i.e., the feedback controller of the agent is $u(t, x(t)) = u^*(x(t)) + u_{exc}(t)$, where

$$u_{exc}(t) := 5\sin(t) + 18\cos(0.4t) + 36\sin(2t) + 0.5\cos(3t),$$

induces excitation in the signal $\hat{x}$.

The HSO in (227), is implemented using three different K_{HSO} matrices, comprised of the same K_3 matrices, computed using the "place" command in MATLAB for poles $p_1 = -2$ and $p_2 = -4$, and three different K_4 matrices. The first two K_4 matrices are computed using gains $K_4 = -I$ and $K_4 - 0.5I$ (denoted in Fig. 29 as HSO - P $= -1$ and HSO - P $= -0.5$, respectively). The third K_4 matrix is selected to be an exponentially varying gain matrix, $K_4 = (1 - 0.9\exp^{-t})0.5I$ (denoted as HSO - Exp in Fig. 29). The MLO in (224) is implemented using a single K_{MLO} matrix, with K_1 computed using the "place" command for poles $p_1 = -2$ and $p_2 = -4$, and $K_2 = 10000I$.

As seen in Fig. 29, all of the weight estimation errors for the designed observers converge to the origin as expected. Even though there is a larger initial estimation error for the HSO, with constant gains, compared to the MLO, the history stack based observers converge much quicker than the MLO. The initial estimation error can be reduced for the HSO either by moving the poles closer to the origin, or implementing an exponentially varying gain matrix, as in the HSO-Exp case. The exponentially varying gain matrix combines the benefits of initial small gains, when the state estimates are inaccurate, with those of progressively larger gains, leading to fast convergence.

7.6.2 Persistently Excited Signal without Noise - Four State System

The second simulation shows a four state system with the exponentially varying HSO and the Kalman filter implementation of the HSO observer in a noise-free setting.

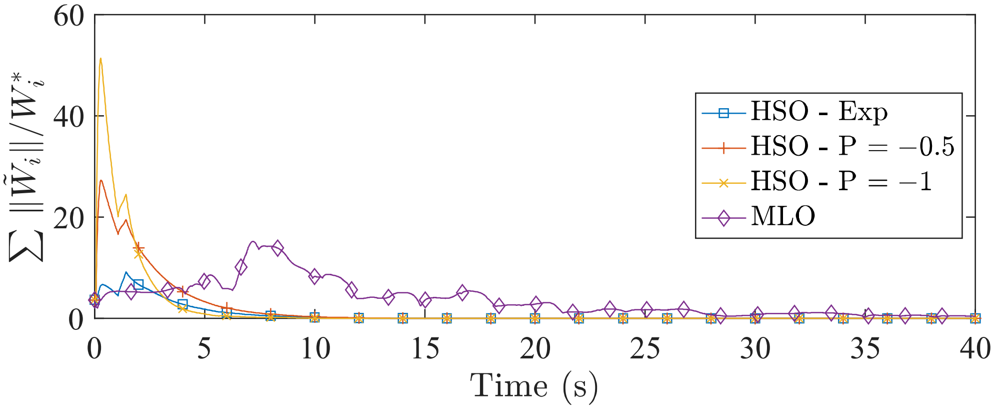


Figure 29: Weight estimation errors for the developed observers with no noise and PE signal.

Similarly to Section 7.6.1, the agent under observation implements a combination of the optimal controller and a known exciting controller. In this simulation, the exciting controller is randomly selected from a uniform distribution in the set $[0, 10]$. The dynamical system of the agent under observation is

$$\dot{x} = \begin{bmatrix} 2 & 4 & 1 & 0 \\ 0 & 3 & 6 & 2 \\ 3 & 2 & 2 & 6 \\ 3 & 5 & 6 & 2 \end{bmatrix} x + \begin{bmatrix} 7 & 2 \\ 4 & 5 \\ 3 & 3 \\ 2 & 6 \end{bmatrix} u, \; y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x. \tag{231}$$

The optimal controller,

$$u^*(x) = \begin{bmatrix} 1.79 & -0.235 & 1.022 & 0.487 \\ -0.345 & 1.75 & 3.96 & 4.35 \end{bmatrix} x,$$

minimizes an LQR problem, $Q = \text{diag}([2, 5, 8, 11])$, $R = \text{diag}([1.5, 0.5])$, and $R(1, 1) = 1.5$ is selected as the known weight.

As seen in Figure 30, both the exponential and Kalman filter implementations of the HSO converge to the origin. In Figure 31, the MLO eventually estimates the unknown weights in the reward and value function, however, the MLO takes

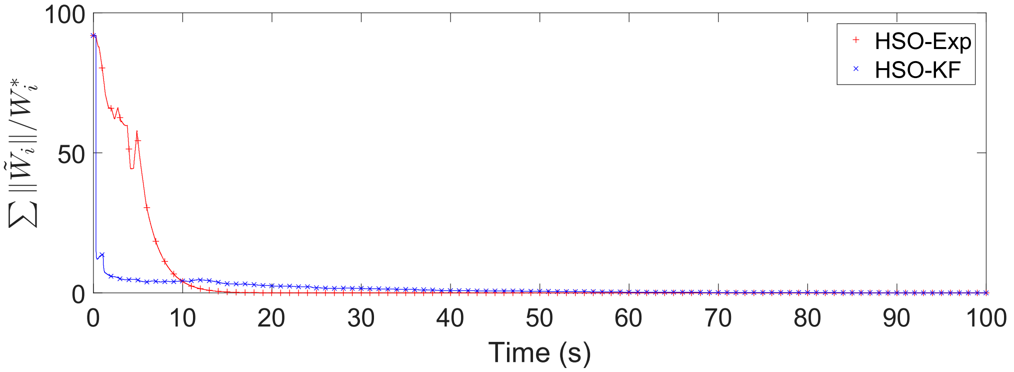


Figure 30: Weight estimation errors for the developed HSO observers with no noise and PE signal with larger dimensional system.

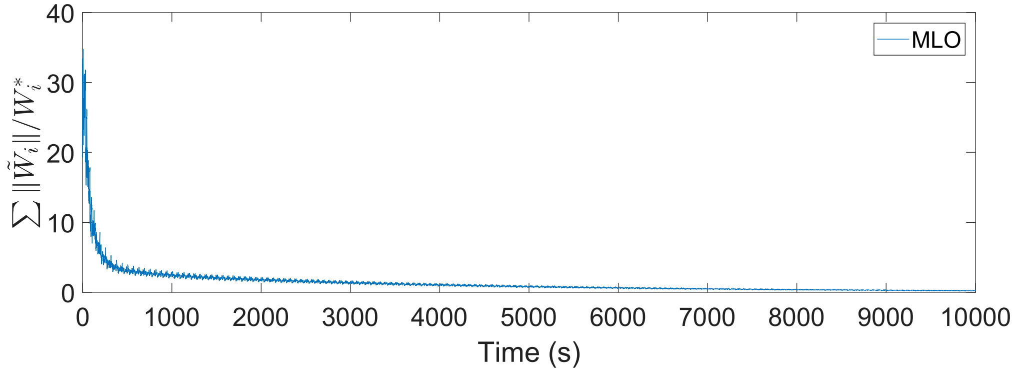


Figure 31: Weight estimation errors for the developed MLO observer with no noise and PE signal with larger dimensional system.

significantly longer than the HSO implementations. This further validates the design for using previously recorded data to help update the weight estimates.

7.6.3 Persistently Excited Signal with Noise

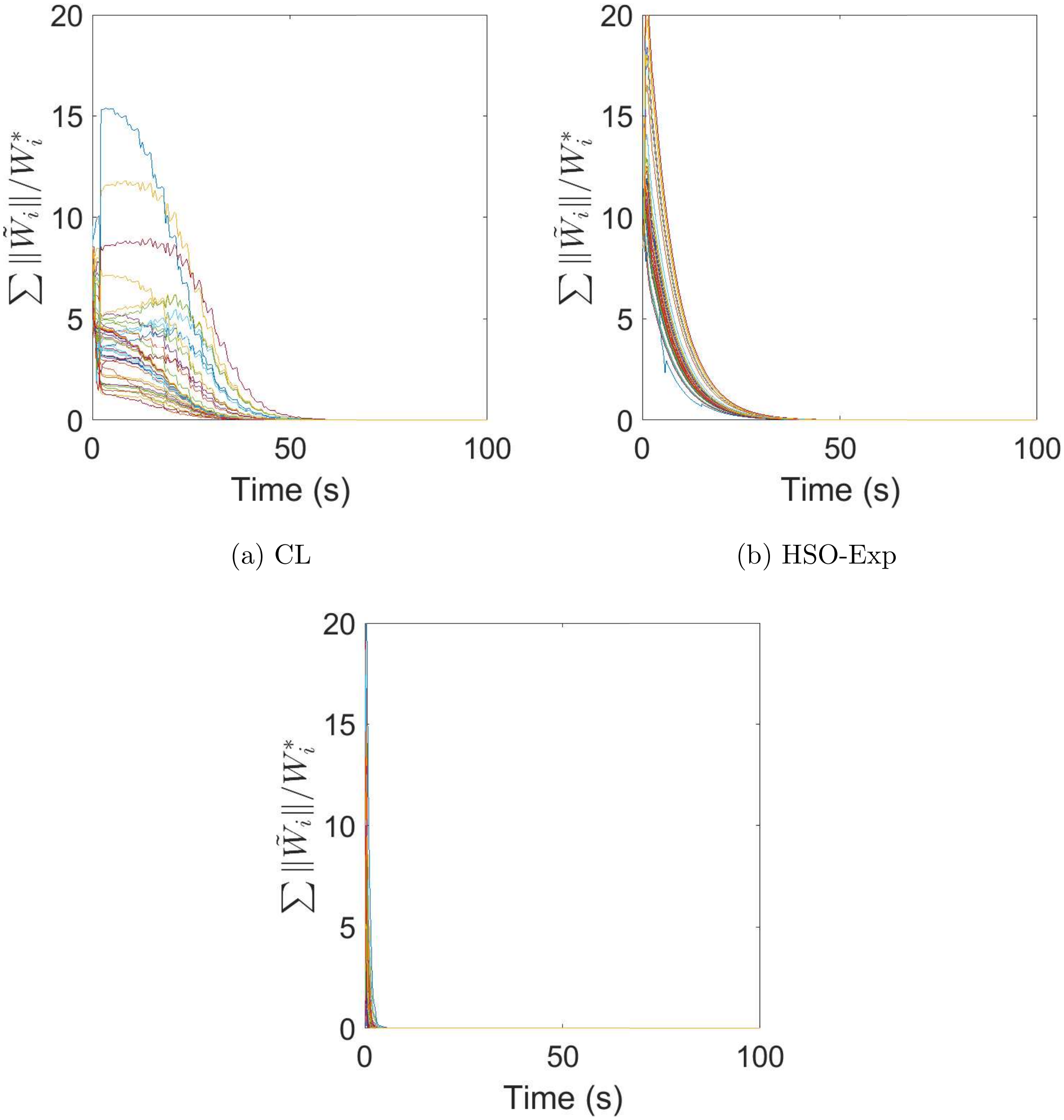
The last simulation is an investigation into noise robustness of the HSO and the Kalman filter implementation of the HSO (called HSO-KF) compared to the CL update law in [137, 139]. The simulation comparison is the same two state system as in Section 7.6.1. The state estimator used for the CL method is developed in [66] (with a slight modification to address the non-Brunovsky form of the dynamics).

Zero-mean Gaussian noise is added to y' and u, with three noise variances used to simulate low-noise ($R_1 = \text{diag}([0.1^2, 0.1^2])$), medium noise ($R_2 = \text{diag}(0.5^2, 0.5^2])$ and high noise ($R_3 = \text{diag}([1^2, 1^2])$) scenarios. Fifty Monte-Carlo simulations for each noise level are conducted and compared with the no-noise case. The Monte-Carlo simulations are shown for each noise level and each method in Figs. 32 - 35. We do not study the behavior of the MLO under noisy measurements due to the added robustness of the HSO due to the use of past data.

Table 3.: Comparison between concurrent learning (CL), KF based implementation of HSO (HSO-KF), and exponential pole selection implementation of HSO (HSO-Exp), with different noise variances. Simulations were ran for 100 seconds over 50 trials with step size $Ts = 0.005$. The standard deviations (SD) simulated are $0.0, 0.1, 0.5$, and 1.0. The metric used for comparison is the average of the average on the trajectories $\sum \tilde{W}_i / W_i^*$, where TT denotes the average over the entire trajectory, and SS denotes the average over the last 30 seconds of the trajectory. The exponential HSO gains are selected similar to Section 7.6.1, except $K_4 = (1 - 0.9 \exp^{-t}) 0.15 I$. The Kalman filter gain is selected using the gain matrix $K_{HSO} = \text{diag}([K_3, K_4])$ where K_3 and K_4 are independent Kalman gains.

	CL		HSO-Exp		HSO-KF	
SD	TT	SS	TT	SS	TT	SS
0.0	0.9855	7.43e-05	0.9101	8.41e-05	0.0446	1.86e-14
0.1	0.8977	0.2647	0.8463	0.1652	0.2591	0.2279
0.5	2.1766	2.0336	1.3064	0.6894	0.7291	0.7277
1.0	5.5415	5.5223	1.9055	1.4667	1.5055	1.4111

The results of the simulation study are shown in Table 3.. As seen from the data, all three methods perform well in the noise free case, and the performance of all three

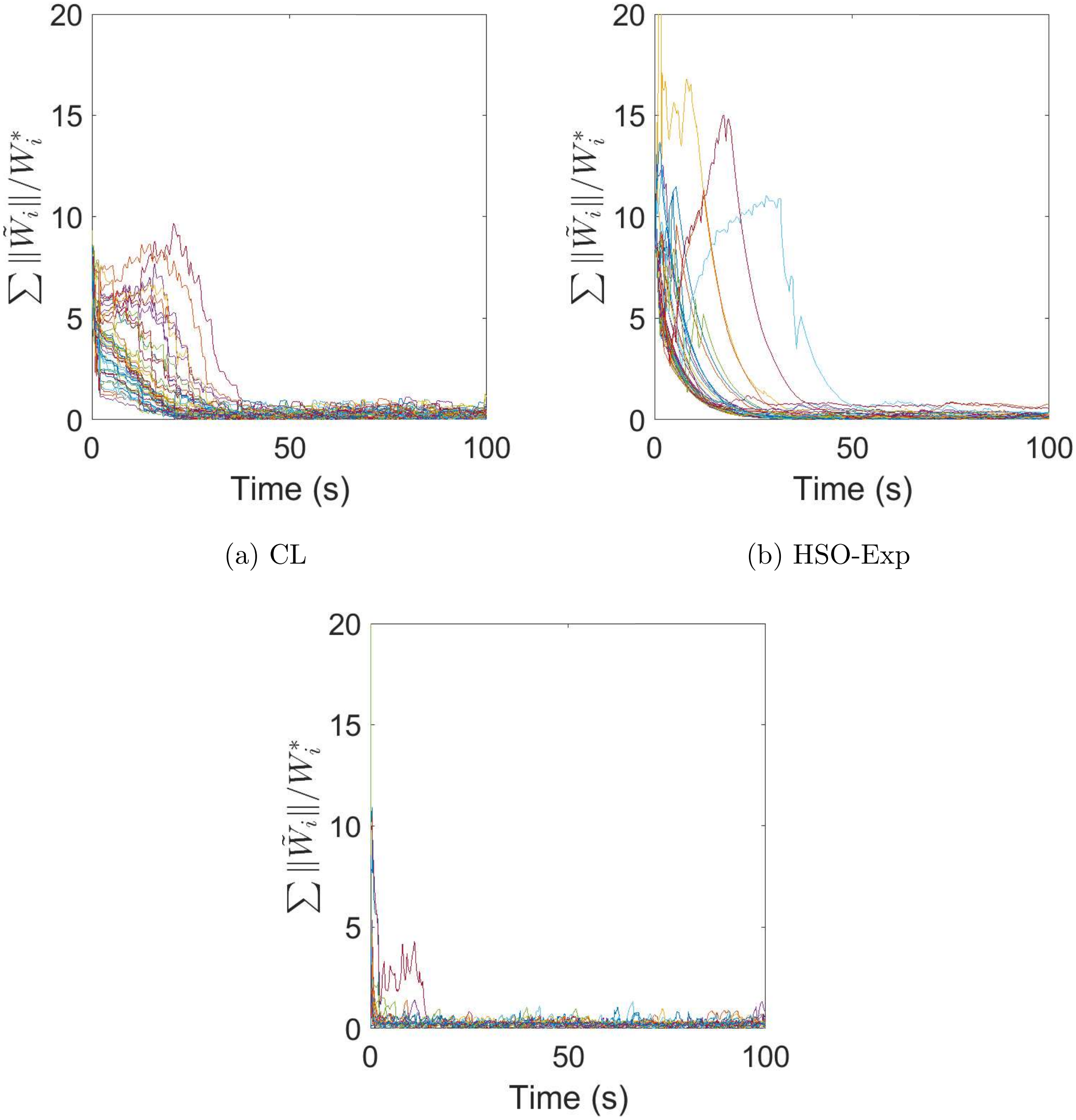


(a) CL

(b) HSO-Exp

(c) HSO-KF

Figure 32: No Noise

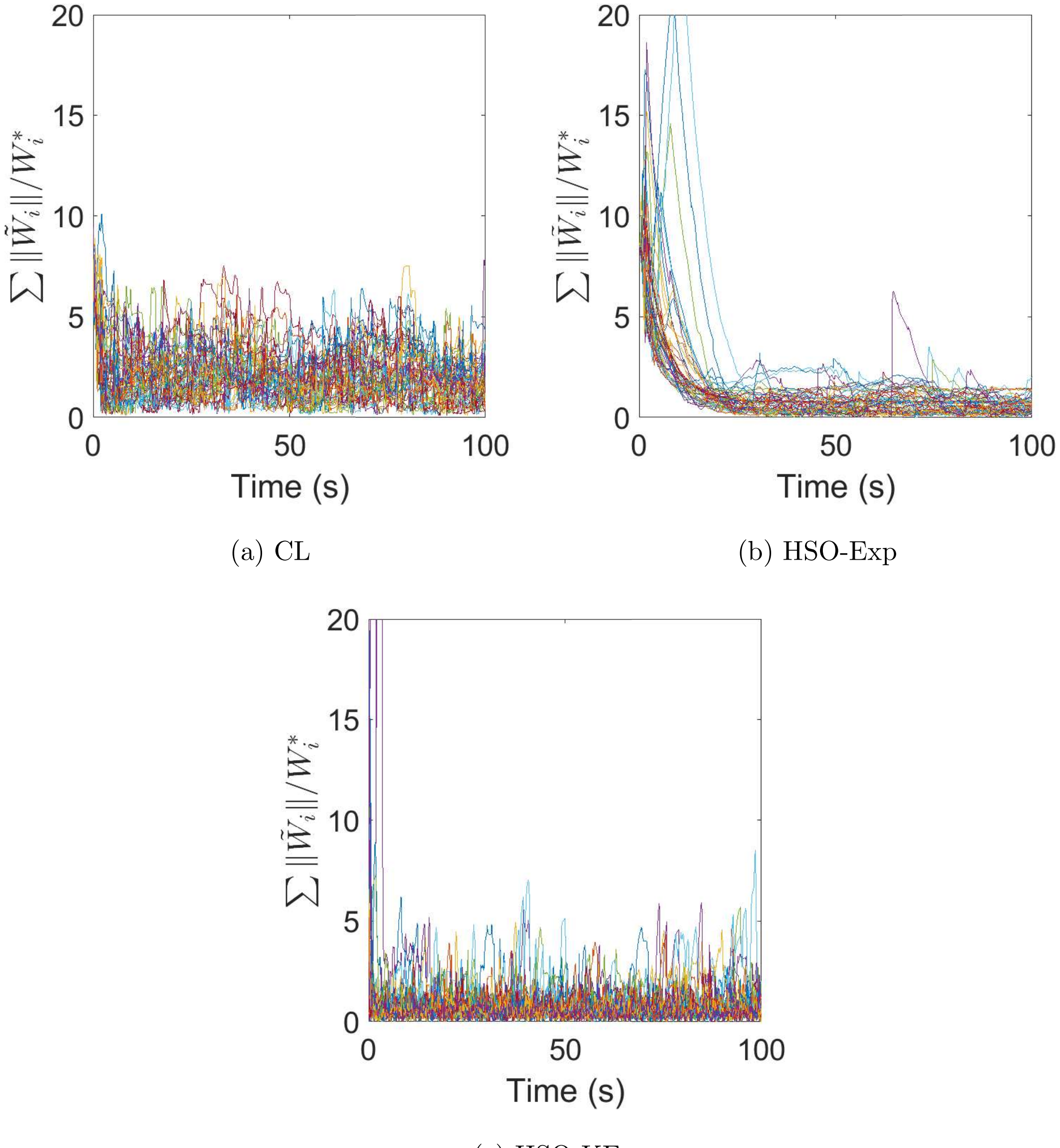


(a) CL

(b) HSO-Exp

(c) HSO-KF

Figure 33: 0.1 Noise Standard Deviation

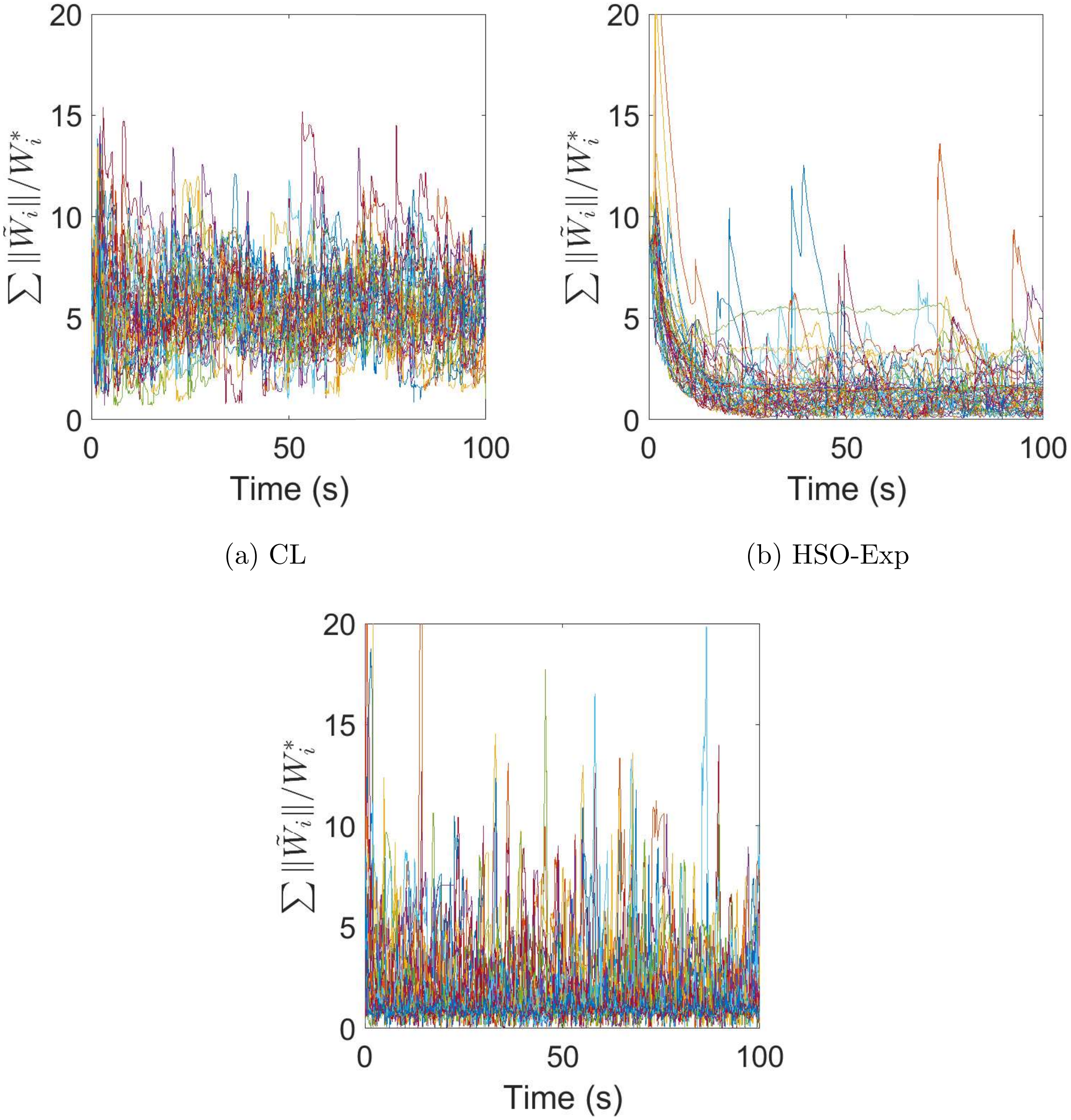


(a) CL

(b) HSO-Exp

(c) HSO-KF

Figure 34: 0.5 Noise Standard Deviation



(a) CL

(b) HSO-Exp

(c) HSO-KF

Figure 35: 1.0 Noise Standard Deviation

methods is comparable in the low noise scenario. The advantages of the two HSO methods over the CL method are evident in the medium and high noise scenarios. Both the HSO-Exp and HSO-KF show better robustness to noise when compared to the CL method, especially in high the noise situation (CL steady state (SS) error is almost four times higher than both HSO methods). Comparing the results of HSO-Exp to HSO-KF, HSO-Exp has lower SS errors for the low and medium noise cases, while, HSO-KF has lower SS errors for the no noise and high noise cases. In addition, HSO-KF converges quicker in every case compared to both CL and HSO-Exp, as evidenced by the average over the whole time interval (TT).

7.7 Conclusion

This chapter presents a novel observer-like formulation for performing online estimation of reward functions using input-output observations. Two observers are proposed and their convergence guarantees are established. The Monte-Carlo simulations demonstrate that the developed observer based IRL techniques, utilizing exponentially varying gains and Kalman gains, demonstrate better noise robustness than existing CL based IRL techniques [137, 139].

Chapter VIII

APPLICATIONS

The previous chapters investigate techniques for inverse reinforcement learning which facilitate reward function estimation in real-time. Specifically, Chapter IV develops a model-based real-time inverse reinforcement learning technique, Chapter V deals with the case where the observed trajectories of an agent are not consistently representative of its internal reward function due to external disturbances, Chapter VI addresses data sparsity issues by estimating the optimal controller and querying for additional data, and Chapter VII formulates the IRL problem in an observer framework to estimate the unknown reward function with noisy measurements.

This chapter aims to discuss potential real-world applications that align with the work developed in this dissertation.

8.1 Consistency Checking/Validation

As autonomy increases in the workforce, misbehavior and fault detection of autonomous entities becomes increasingly important. Traditional detection methods [77, 78, 145] generally classify actions into predefined categories and analyze the trajectories of agents without attention to the underlying intent that generates the trajectories. As a result, these methods lack the full understanding of what the agent is trying to achieve. If only the trajectories are used to detect misbehavior, then malicious, yet seemingly normal behavior might not be detected. Additionally, intent monitoring methods that lack the ability to adapt to alterations in task objectives online are liable to generate false alarms.

Adaptability can make the misbehavior detection problem significantly more difficult to solve. Adaptation necessitated by a fault could potentially be misdiagnosed as misbehavior, even though the agent is acting in accordance with it's intended design. Intent understanding and estimation could potentially result in monitoring systems that can accurately distinguish between faults and misbehavior.

The work developed in this dissertation may allow for improved monitoring of autonomous entities and help increase safety in the workforce. The intent of an entity will be interpreted as a reward function and estimation of this reward function is achieved utilizing real-time IRL techniques developed in this dissertation. This monitoring would be achieved by utilizing the estimated intent, or reward function, of the autonomous agent and comparing the estimate to the designed reward function that the agent is supposed to be acting with respect to. Area surveillance utilizing UAVs, assembly line robots, and autonomous taxis are some of the specific scenarios that consistency checking utilizing real-time IRL methods in this dissertation can help resolve.

8.2 Pilot Modeling

8.2.1 Introduction

Everyday use of unmanned aerial vehicles (UAVs) will soon be commonplace, and the need for safe navigation in urban areas is critical if the UAV market is going to be expanded to tasks such as package delivery, search and rescue, etc. The task to maintain a congested airfield in cities becomes even more difficult in the presence of disturbances. While steady wind fields can be predicted, unpredictable gusts can make urban air mobility challenging. Collisions with buildings or other UAVs can result in injury to civilians and property damage. Therefore, a framework is needed in order to facilitate human remote controlled UAVs and autonomous systems to safely navigate a common airspace. A pilot modeling project is part of a group effort at Oklahoma State University to develop an all encompassing method for UAVs, both human controlled and autonomous, where gust estimation is incorporated in a path recommendation/implementation platform to facilitate safe navigation and increase urban air density of vehicles.

The research task is to develop techniques for modeling pilot preferences and updating these preferences in real-time by observing the pilot's behavior. The proposed concept is to model pilot behavior as a reward, or cost, function using optimal control theory, with the goal to uncover a pilots preferences. The model will then be used to recommend pilot-specific paths to navigate through a wind field.

The proposed reward function based pilot modeling research could use a combination of offline IRL methods existing in literature, concurrently with the real-time IRL techniques from Chapters IV - VII, to estimate reward functions and to update them continuously using new data. The updated reward function will then be used for forward reinforcement learning or optimal control to recommend optimal paths to the pilot or the autonomous system based on current gust predictions and observed obstacles.

8.2.2 Literature Review

The vast majority of pilot modeling literature can be structured into two distinct categories: physiological modeling and psychological modeling.

Physiological pilot models are focused on how the body of the pilot is affected during flight and how to accurately model these affects, along with how pilots perceive the aircraft in flight. This category can be broken down further into categories such as sensory dynamics (perception) and bio-dynamics (human body) [96]. Sensory dynamics include systems such as visual and vestibular (spatial orientation) systems [102]. The bio-dynamic modeling generally models human body responses as classical mechanical systems, such as incorporating spring, mass, dampers systems, or flexible beams [81], for modeling human spines in flight. Since, the work in this dissertation will focus on unmanned aircraft, the physiological effects of flight on pilots are not important.

Psychological modeling is focused on the control component of pilot modeling and attempts to model a pilot's thought process during flight. This category can be broken down into a variety of methods, however, the majority of pilot modeling in the literature use classical control approaches [56, 104, 105]. These classical control approaches are based on the concept of the "crossover model" [104], where a pilot's control action, whether that constitutes lead or lag control, drives the open loop function approximately to

$$Y_p Y_c = \frac{\omega_c e^{-j\omega\tau_e}}{j\omega}, \tag{232}$$

where Y_c is the controlled element dynamics, Y_p is the pilot's control action, ω_c is the crossover frequency, and (232) includes a time-delay constant term, where τ_e is modeled as the pilot's response time resulting in quasi-linear [76] describing functions.

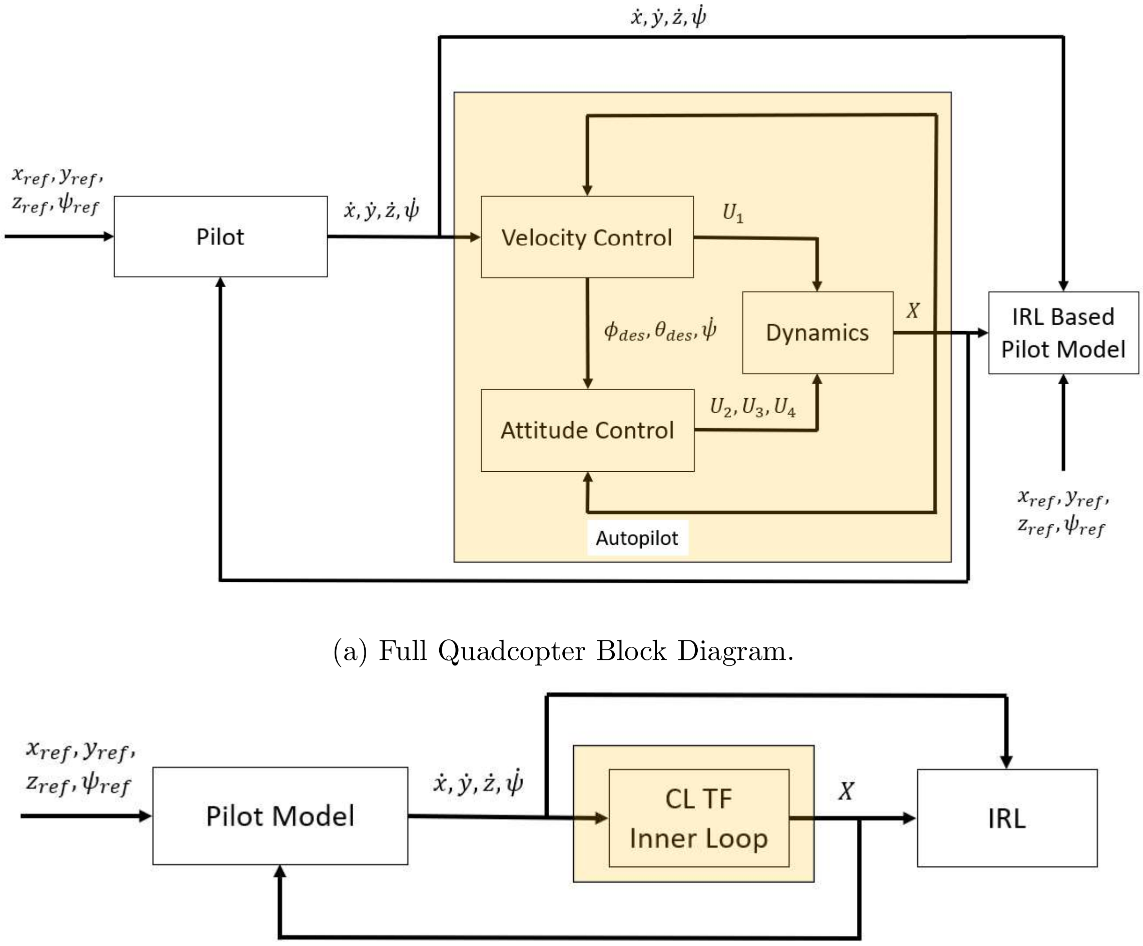
There has been some work focused on human in the loop modeling [55] using optimal control theory for linear systems [14, 82, 150, 152], and more recently, there has been some research focused on newer control techniques for reward function based pilot modeling utilizing machine learning approaches [6, 155]. Results such as [6, 155] categorize the pilots goals into a variety of categories such as minimizing effort, staying on assigned trajectory, maintaining proper distance among other aircraft, how many aircraft occupy the current space, etc., and use game theory and reinforcement learning approaches to predict each players strategy. The results in [91, 121] use game theory to predict pilot behavior during mid-air encounters, and utilize the assumption, as the authors state in [91], that pilots, in general, actively direct the aircraft to avoid collisions as opposed to relying on avoidance recommendation systems currently in place.

However, most of the aforementioned methods described in literature are focused on manned fixed wing aircraft, while the pilot modeling task in this dissertation is focused on unmanned multirotor aircraft. Of the methods developed for unmanned aircraft pilot modeling using reward function based pilot models, nearly all focus on specific scenarios, such as collision avoidance. However, since pilot models are likely to change drastically through the duration of the flight depending on the current task, different pilot models may be needed for a variety of scenarios that may develop during a full flight mission. Therefore, the work presented in this dissertation could help by developing a wider range of reward functions for various flight scenarios, and exploring viability of updating these reward functions in real-time. Utilizing the real-time IRL methods developed in this dissertation to update the pilot models in real-time would facilitate a more realistic modeling approach and allow for more broad flight conditions.

8.2.3 Preliminary Results

Preliminary results on the pilot modeling project were achieved by analyzing UAV control from a kinematic (as opposed to dynamic) point of view. Traditional control designs for quadcopter flight utilize inputs such thrust and torques. However, the hypothesis is that pilots instead interpret UAV flight utilizing inputs such as linear velocities and yaw rate. Therefore, the preliminary work is performed from this perspective.

The quadcopter kinematic model is developed by designing a velocity tracking controller for the quadcopter's dynamics. A block diagram of this is shown in Fig. 36a. In Fig. 36a, velocity and attitude controllers are designed and the closed-loop dynamics are then transformed into Fig. 36b. The states and controls for the



(a) Full Quadcopter Block Diagram.

(b) Closed Loop Quadcopter Block Diagram.

Figure 36: Kinematic Control Simulation Block Diagrams.

kinematic control problem are

$$X := \left[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\right]^T,$$
$$U := \left[\dot{x}_d, \dot{y}_d, \dot{z}_d, \dot{\psi}_d\right]^T, \tag{233}$$

where the subscript d on a variable denotes the desired value of that variable.

A quadcopter's translational dynamics can be described by [61]

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} - k_t \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{234}$$

where U_1 is the thrust, k_t is an aerodynamic thrust drag coefficient, m is the mass, g is the gravity, and the rotation matrix is given by

$$R = \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\phi\sin\theta - \sin\psi\cos\phi & \cos\phi\sin\theta\cos\psi + \sin\psi\sin\phi \\ \cos\theta\sin\psi & \sin\psi\sin\phi\sin\theta + \cos\theta\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}. \tag{235}$$

Using small angle approximation, the rotation matrix becomes

$$R = \begin{bmatrix} 1 & \phi\theta - \psi & \theta + \psi\phi \\ \psi & \psi\phi\theta + 1 & \theta\psi - \phi \\ -\theta & \phi & 1 \end{bmatrix}. \tag{236}$$

The thrust U_1 is designed as a proportional controller

$$U_1 = mg + mk_{p_{13}} \left(\dot{z}_d - \dot{z} \right). \tag{237}$$

where $k_{p_{13}}$ is the proportional gain.

The rotational motion of a quadcopter can be described by [24, 25]

$$\begin{aligned} \ddot{\phi} &= \dot{\theta}\dot{\psi}\frac{I_{yy} - I_{zz}}{I_{xx}} + \frac{l}{I_{xx}}U_2, \\ \ddot{\theta} &= \dot{\phi}\dot{\psi}\frac{I_{zz} - I_{xx}}{I_{yy}} + \frac{l}{I_{yy}}U_3, \\ \ddot{\psi} &= \dot{\theta}\dot{\phi}\frac{I_{xx} - I_{yy}}{I_{zz}} + \frac{1}{I_{zz}}U_4 \end{aligned} \tag{238}$$

where I_{xx}, I_{yy}, I_{zz} are the moments of inertia and U_2, U_3, U_4 are the torques.

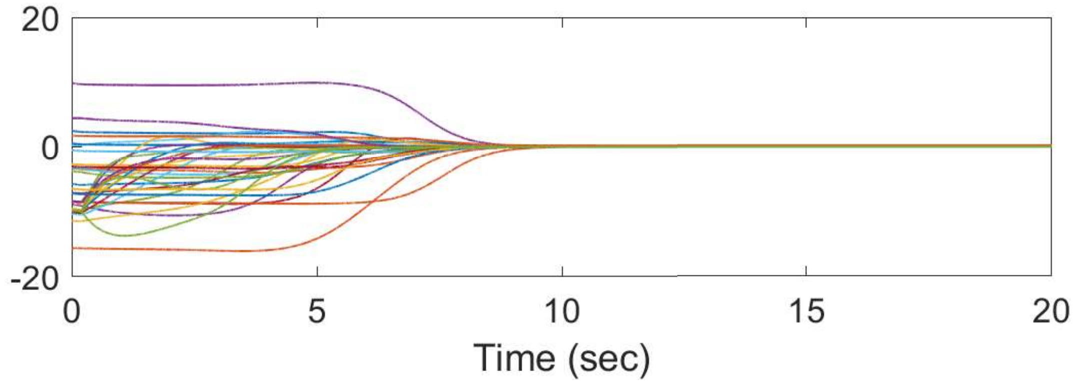


Figure 37: Quadcopter simulation using linear trajectories and linearized model inside IRL.

The controls U_2, U_3, U_4 are designed as PD controllers

$$\begin{aligned} U_2 =& k_{p_{21}} \left(\phi_d - \phi\right) + k_{d_1} \left(\dot{\phi}_d - \dot{\phi}\right), \\ U_3 =& k_{p_{22}} \left(\theta_d - \theta\right) + k_{d_2} \left(\dot{\theta}_d - \dot{\theta}\right), \\ U_4 =& k_{p_{23}} \left(\psi_d - \psi\right) + k_{d_3} \left(\dot{\psi}_d - \dot{\psi}\right), \end{aligned} \tag{239}$$

where the desired angles ϕ_d, θ_d are given by

$$\begin{aligned} \theta_d &= \arctan\left(\frac{mk_{p_{12}}\left(\dot{y}_d - \dot{y}\right)\sin\psi_d + mk_{p_{11}}\left(\dot{x}_d - \dot{x}\right)\cos\psi_d}{mg + mk_{p_{13}}\left(\dot{z}_d - \dot{z}\right)}\right), \\ \phi_d &= \arctan\left(\cos\theta_d \frac{mk_{p_{11}}\left(\dot{x}_d - \dot{x}\right)\sin\psi_d - mk_{p_{12}}\left(\dot{y}_d - \dot{y}\right)\cos\psi_d}{mg + mk_{p_{13}}\left(\dot{z}_d - \dot{z}\right)}\right), \end{aligned} \tag{240}$$

and $k_{p_{11}}, k_{p_{12}}, k_{p_{21}}, k_{p_{22}}, k_{p_{23}}, k_{d_1}, k_{d_2}, k_{d_3}$ are control gains. Using small angle approximations and a linear approximation for arctan [123], (240) becomes

$$\begin{aligned} \theta_d &= \frac{\pi}{4}\left(\frac{mk_{p_{12}}\left(\dot{y}_d - \dot{y}\right)\psi_d + mk_{p_{11}}\left(\dot{x}_d - \dot{x}\right)}{mg + mk_{p_{13}}\left(\dot{z}_d - \dot{z}\right)}\right), \\ \phi_d &= \frac{\pi}{4}\left(\frac{mk_{p_{11}}\left(\dot{x}_d - \dot{x}\right)\psi_d - mk_{p_{12}}\left(\dot{y}_d - \dot{y}\right)}{mg + mk_{p_{13}}\left(\dot{z}_d - \dot{z}\right)}\right). \end{aligned} \tag{241}$$

Linearizing (234) and (238) about the origin, while using (237), (239), and (241), yields the linear system

$$\begin{aligned}
\ddot{x} &= g\theta - \frac{k_t}{m}\dot{x}, \\
\ddot{y} &= -g\phi - \frac{k_t}{m}\dot{y}, \\
\ddot{z} &= -\frac{k_t}{m}\dot{z} - k_{p_{13}}\dot{z}_d + k_{p_{13}}\dot{z}, \\
\ddot{\phi} &= \frac{b_1 \pi k_{p_{21}} k_{p_{12}} \left(\dot{y} - \dot{y}_d\right)}{4g} - b_1 k_{d_1}\dot{\phi} - b_1 k_{p_{21}}\phi, \\
\ddot{\theta} &= \frac{b_2 \pi k_{p_{22}} k_{p_{11}} \left(\dot{x}_d - \dot{x}\right)}{4g} - b_2 k_{p_{22}}\theta - b_2 k_{d_2}\dot{\theta}, \\
\ddot{\psi} &= -b_3 k_{p_{23}}\psi + b_3 k_{d_3}\dot{\psi}_d - b_3 k_{d_3}\dot{\psi}, \qquad (242)
\end{aligned}$$

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

The controller that the agent implements is a combination of the optimal controller and an exciting controller which is randomly selected from the set $u_i \in [0, 1]$.

Fig. 37 shows the preliminary results for the quadcopter simulation. As seen in the figure, the reward function is able to be estimated using trajectory data from the linear simulations. The values used in the simulation are: $l = 0.23$ m, $I_{xx} = I_{yy} = 7.5 \times 10^{-3}$ kg m^2, $I_{zz} = 1.3 \times 10^{-2}$ kg m^2, $k_t = 0.001, g = 9.81$ m/s^2, $m = 1$ kg, $k_{p_{11}} = 5.25, k_{p_{12}} = 6, k_{p_{13}} = 3, k_{p_{21}} = 2, k_{p_{22}} = 1, k_{p_{23}} = 0.35, k_{d_1} = 0.5, k_{d_2} = 0.4, k_{d_3} = 0.1$, and the matrices to be found are

$$Q = \text{diag}([9.5752, 6.9139, 2.8378, 0, 0, 0, 0, 0, 11.6834, 0, 0, 0]),$$

and

$$R = \text{diag}([9.572, 3.4773, 14.4034, 0.1707]).$$

The preliminary results that are established utilize a linearized dynamic model of a quadrotor since it has a known optimal value function. Future work of this section is to utilize the full nonlinear dynamics and estimate Q and R matrices. Utilizing the estimated Q and R matrices, an numerical optimization program, such as GPOPS [125], can be utilized to generate optimal trajectories for the nonlinear system. Once the trajectories are generated, a metric that may evaluate the quality of the reward function estimate would be to compare the trajectories from the measured data that generated the reward function to the trajectories that are generated from GPOPS.

8.3 Learning (IRL) and Control (RL) in Real-Time

The last application is a learning and control architecture where the work in this dissertation would be to perform RL and IRL in real-time. The situation under consideration could be a team of two agents working together on a task. Utilizing the assumption that one of the two agents is designed to properly complete the task, or controlled by a human, and the other agent is trying to learn the reward function to copy the task. The proposed application is to analyze the stability of using a combination of RL/IRL in real-time, in which the assisting agent is simultaneously attempting to learn the reward function and use the updated reward function for real-time control to complete the task. This would involve situations where an autonomous system and a human controlled robot are working together on a task, and the autonomous system is simultaneously learning what the task is, while learning how to complete the task.

Utilizing the methods discussed in this dissertation, and pursuing additional IRL formulations for double filter ideas which align with recent real-time RL results in [62, 63], could be explored.

Chapter IX

CONCLUSION AND FUTURE WORK

In this dissertation, model-based inverse reinforcement learning methods are developed. The work herein provides foundational methods for reward function estimation in real-time and under non-ideal situations. Chapter IV develops a data-driven model-based inverse reinforcement learning technique that is less data intensive than its model-free counterparts, which help facilitate reward function estimation in real-time. Chapter V addresses IRL for scenarios that the observed trajectories of an agent under observation are inconsistent with its internal reward function. Chapter VI attempts to further address the issue of sparsity of available data by formulating a method to artificially create additional data to help drive reward function estimation if trajectories are not sufficiently information rich. Chapter VII formulates the IRL problem in an observer framework to solve the IRL problem in the presence of noisy or imperfect measurements, and Chapter VIII discusses applications relevant to the methods developed in this dissertation.

This dissertation focuses on addressing the five key challenges in real-time IRL: (a) sparsity of available data, (b) nonuniqueness of solutions, (c) partial measurements, (d) noisy/imperfect measurements, and (e) inconsistent observations. In the subsequent chapters, this dissertation addressed: (a) sparsity of available data, (c) partial measurements, (d) noisy/imperfect measurements, and (e) inconsistent observations.

Future work of this dissertation will be revolved around developing methods to solve the non-uniqueness concern that is an important topic in the field of IRL. In addition, a large majority of the effort will be spent developing on the preliminary discussions on the applications relevant to the methods presented in this dissertation.

Bibliography

[1] Pieter Abbeel. *Apprenticeship learning and reinforcement learning with application to robotic control.* Stanford University, 2008.

[2] Pieter Abbeel, Adam Coates, and Andrew Ng. Autonomous helicopter aerobatics through apprenticeship learning. *Int. J. Robot. Res.*, 29(13):1608–1639, 2010.

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

[4] Pieter Abbeel and Andrew Y. Ng. Inverse reinforcement learning. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 554–558. Springer, Boston, MA, 2010.

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

[6] Berat Mert Albaba and Yildiray Yildiz. Modeling cyber-physical human systems via an interplay between reinforcement learning and game theory. *Annu. Rev. Control*, 2019.

[7] B. Anderson. Exponential stability of linear equations arising in adaptive identification. *IEEE Trans. Autom. Control*, 22(1):83–88, February 1977.

[8] Saurabh Arora, Prashant Doshi, and Bikramjit Banerjee. A framework and method for online inverse reinforcement learning. arXiv:1805.07871, 2018.

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

[10] Christopher G. Atkeson and Stefan Schaal. Robot learning from demonstration. In *Proc. Int. Conf. Mach. Learn.*, volume 97, pages 12–20, 1997.

[11] J. A. Bagnell, Joel Chestnutt, David M. Bradley, and Nathan D. Ratliff. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems*, pages 1153–1160, 2007.

[12] Paul Bakker and Yasuo Kuniyoshi. Robot see, robot do: an overview of robot imitation. In *AISB96 Workshop on Learning in Robots and Animals*, pages 3–11, 1996.

[13] Yaakov Bar-Shalom. Optimal simultaneous state estimation and parameter identification in linear discrete-time systems. *IEEE Trans. Autom. Control*, 17(3):308–319, 1972.

[14] Sheldon Baron, D. Kleinman, and W. Levison. An optimal control model of human response part II: Prediction of human performance in a complex task. *Automatica*, 6(3):371–383, 1970.

[15] G. Bastin and M. R. Gevers. Stable adaptive observers for nonlinear time-varying systems. *IEEE Trans. Autom. Control*, 33(7):650–658, 1988.

[16] G. Besançon, J. de León-Morales, and O. Huerta-Guevara. On adaptive observers for state affine systems. *Int. J. Control*, 79(06):581–591, 2006.

[17] Gildas Besançon. Remarks on nonlinear adaptive observer design. *Syst. Control Lett.*, 41(4):271–280, 2000.

[18] Gildas Besançon and A Ţiclea. On adaptive observers for systems with state and parameter nonlinearities, 2017.

[19] Tao Bian, Yu Jiang, and Zhong-Ping Jiang. Adaptive dynamic programming and optimal control of nonlinear nonaffine systems. *Automatica*, 50(10):2624–2632, 2014.

[20] Tao Bian and Zhong-Ping Jiang. Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design. *Automatica*, 71:348–360, 2016.

[21] Aude Billard, Sylvain Calinon, Rüdiger Dillmann, and Stefan Schaal. Robot programming by demonstration. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, chapter 59, pages 1371–1394. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[22] Kenneth Bogert and Prashant Doshi. Multi-robot inverse reinforcement learning under occlusion with interactions. In *Proc. Conf. Auton. Agents MultiAgent Syst.*, pages 173–180. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[23] Kenneth Bogert and Prashant Doshi. Multi-robot inverse reinforcement learning under occlusion with state transition estimation. In *Proc. Conf. Auton. Agents MultiAgent Syst.*, pages 1837–1838. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

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

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

[26] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proc. Int. Conf. Artif. Intell. Stat.*, volume 15, 2011.

[27] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. arXiv:1904.06387, 2019.

[28] Daniel Brown and Scott Niekum. Machine teaching for inverse reinforcement learning: algorithms and applications. arXiv:1805.07687, 2018.

[29] P. Brunovskỳ. A classification of linear controllable systems. *Kybernetika*, 6(3):173–188, 1970.

[30] Y. Uny Cao, Alex S. Fukunaga, and Andrew Kahng. Cooperative mobile robotics: antecedents and directions. *Auton. Robot.*, 4(1):7–27, 1997.

[31] Mou Chen, Shu-Yi Shao, and Bin Jiang. Adaptive neural control of uncertain nonlinear systems using disturbance observer. *IEEE Trans. Cybern.*, 47(10):3110–3123, 2017.

[32] Mou Chen and Jing Yu. Adaptive dynamic surface control of nsvs with input saturation using a disturbance observer. *Chin. J. Aeronaut.*, 28(3):853–864, 2015.

[33] Tao Chen, Julian Morris, and Elaine Martin. Particle filters for state and parameter estimation in batch processes. *J. Process Control*, 15(6):665–673, 2005.

[34] Wen-Hua Chen. Disturbance observer based control for nonlinear systems. *IEEE/ASME Trans. Mechatron.*, 9(4):706–710, 2004.

[35] Jaedeug Choi and Kee-Eung Kim. Map inference for Bayesian inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1989–1997, 2011.

[36] Jaedeug Choi and Kee-Eung Kim. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems*, pages 305–313, 2012.

[37] Jaedeug Choi and Kee-Eung Kim. Bayesian nonparametric feature construction for inverse reinforcement learning. In *IJCAI*, pages 1287–1293, 2013.

[38] Sungjoon Choi, Kyungjae Lee, and Songhwai Oh. Robust learning from demonstrations with mixed qualities using leveraged gaussian processes. *IEEE Trans. Robot*, 35(3):564–576, 2019.

[39] Michelle S. Chong, Dragan Nešić, Romain Postoyan, and Levin Kuhlmann. State and parameter estimation of nonlinear systems: a multi-observer approach. In *IEEE Conf. Decis. Control*, pages 1067–1072. IEEE, 2014.

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

[41] Girish Chowdhary. *Concurrent learning for convergence in adaptive control without persistency of excitation*. PhD thesis, Georgia Institute of Technology, December 2010.

[42] Girish Chowdhary, Hassan A. Kingravi, Jonathan P. How, and Patricio A. Vela. Bayesian nonparametric adaptive control using Gaussian processes. *IEEE Trans. Neural Netw. Learn. Syst.*, 26(3):537–550, 2015.

[43] Girish Chowdhary, Maximilian Mühlegg, JonathanP. How, and Florian Holzapfel. Concurrent learning adaptive model predictive control. In Qiping Chu, Bob Mulder, Daniel Choukroun, Erik-Jan van Kampen, Coen de Visser, and Gertjan Looye, editors, *Advances in Aerospace Guidance, Navigation and Control*, pages 29–47. Springer Berlin Heidelberg, 2013.

[44] Girish Chowdhary, Tansel Yucelen, Maximillian Mühlegg, and Eric N. Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *Int. J. Adapt. Control Signal Process.*, 27(4):280–301, 2013.

[45] Adam Coates, Pieter Abbeel, and Andrew Y. Ng. Apprenticeship learning for helicopter control. *Commun. ACM*, 52(7):97–105, 2009.

[46] Daniel R. Creveling, Philip E. Gill, and Henry D. I. Abarbanel. State and parameter estimation in nonlinear systems as an optimal tracking problem. *Phys. Lett. A*, 372(15):2640–2644, 2008.

[47] Huyen T. Dinh, Rushikesh Kamalapurkar, Shubhendu Bhasin, and Warren E. Dixon. Dynamic neural network-based robust observers for uncertain nonlinear systems. *Neural Netw.*, 60:44–52, December 2014.

[48] Manuel A. Duarte and K. S. Narendra. Combined direct and indirect approach to adaptive control. *IEEE Trans. Autom. Control*, 34(10):1071–1075, October 1989.

[49] Gregory Dudek, Michael R. M. Jenkin, Evangelos Milios, and David Wilkes. A taxonomy for multi-agent robotics. *Auton. Robot.*, 3(4):375–397, 1996.

[50] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pages 49–58, 2016.

[51] S. T. Glad and L. Ljung. Model structure identifiability and persistence of excitation. In *Proc. IEEE Conf. Decis. Control*, pages 3236–3240, 1990.

[52] Michael Green and John B Moore. Persistence of excitation in linear systems. *Syst. Control Lett.*, 7(5):351–360, 1986.

[53] Daniel Grollman and Aude Billard. Robot learning from failed demonstrations. *Int. J. Soc. Robot.*, 4(4):331–342, 2012.

[54] Jack K. Hale. *Ordinary differential equations.* Robert E. Krieger Publishing Company, Inc., second edition, 1980.

[55] Keita Hara, Masaki Inoue, and José María Maestre. Data-driven human modeling: Quantifying personal tendency toward laziness. *IEEE Control Syst. Lett.*, 5(4):1219–1224, 2020.

[56] Ronald A Hess. Simplified approach for modelling pilot pursuit control behaviour in multi-loop flight control tasks. *Proc. Inst. Mech. Eng., Part G: J. Aero. Eng.*, 220(2):85–102, 2006.

[57] Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press., Cambridge, 1993.

[58] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Netw.*, 4:251–257, 1991.

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

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

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

[62] Sumit Kumar Jha, Sayan Basu Roy, and Shubhendu Bhasin. Memory-efficient filter based novel policy iteration technique for adaptive lqr. In *Proc. Am. Control Conf.*, pages 4963–4968, 2018.

[63] Sumit Kumar Jha, Sayan Basu Roy, and Shubhendu Bhasin. Initial excitation-based iterative algorithm for approximate optimal control of completely unknown lti systems. *IEEE Trans. Autom. Control*, 64(12):5230–5237, 2019.

[64] Z. Jin, H. Qian, S. Chen, and M. Zhu. Convergence analysis of an incremental approach to online inverse reinforcement learning. *J. Zhejiang Univ. - Sci. C*, 12(1):17–24, 2011.

[65] R. E. Kalman. When is a linear control system optimal? *J. Basic Eng.*, 86(1):51–60, 1964.

[66] Rushikesh Kamalapurkar. Online output-feedback parameter and state estimation for second order linear systems. In *Proc. Am. Control Conf.*, pages 5672–5677, Seattle, WA, USA, May 2017.

[67] Rushikesh Kamalapurkar. Simultaneous state and parameter estimation for second-order nonlinear systems. In *Proc. IEEE Conf. Decis. Control*, pages 2164–2169, Melbourne, VIC, Australia, December 2017.

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

[69] Rushikesh Kamalapurkar, Lindsey Andrews, Patrick Walters, and Warren E. Dixon. Model-based reinforcement learning for infinite-horizon approximate optimal tracking. *IEEE Trans. Neural Netw. Learn. Syst.*, 28(3):753–758, March 2017.

[70] Rushikesh Kamalapurkar, Huyen T. Dinh, Shubhendu Bhasin, and Warren E. Dixon. Approximate optimal trajectory tracking for continuous-time nonlinear systems. *Automatica*, 51:40–48, January 2015.

[71] Rushikesh Kamalapurkar, Justin R. Klotz, and Warren E. Dixon. Concurrent learning-based online approximate feedback Nash equilibrium solution of N-player nonzero-sum differential games. *IEEE/CAA J. Autom. Sin.*, 1(3):239–247, July 2014. Special Issue on Extensions of Reinforcement Learning and Adaptive Control.

[72] Rushikesh Kamalapurkar, Ben Reish, Girish Chowdhary, and Warren E. Dixon. Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Trans. Autom. Control*, 62(7):3594–3601, July 2017.

[73] Rushikesh Kamalapurkar, Joel A. Rosenfeld, and Warren E. Dixon. Efficient model-based reinforcement learning for approximate online optimal control. *Automatica*, 74:247–258, December 2016.

[74] Rushikesh Kamalapurkar, Patrick Walters, and Warren E. Dixon. Model-based reinforcement learning for approximate optimal regulation. *Automatica*, 64:94–104, February 2016.

[75] Rushikesh Kamalapurkar, Patrick Walters, Joel A. Rosenfeld, and Warren E. Dixon. *Reinforcement learning for optimal feedback control: A Lyapunov-based approach*. Communications and Control Engineering. Springer International Publishing, 2018.

[76] Tosio Kato. *Quasi-linear equations of evolution, with applications to partial differential equations*. Springer, 1975.

[77] Richard Kelley, Christopher King, Alireza Tavakkoli, Mircea Nicolescu, Monica Nicolescu, and George Bebis. An architecture for understanding intent using

a novel hidden Markov formulation. *Int. J. Humanoid Robot.*, 5(02):203–224, 2008.

[78] Richard Kelley, Alireza Tavakkoli, Christopher King, Amol Ambardekar, Monica Nicolescu, and Mircea Nicolescu. Context-based Bayesian intent recognition. *IEEE Trans. Auton. Ment. Dev.*, 4(3):215–225, 2012.

[79] S. Kersting and M. Buss. Concurrent learning adaptive identification of piecewise affine systems. In *Proc. IEEE Conf. Decis. Control*, pages 3930–3935, December 2014.

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

[81] S. Kitazaki and M. Griffin. A modal analysis of whole-body vertical vibration, using a finite element model of the human body. *J. Sound Vib.*, 200(1):83–103, 1997.

[82] D. Kleinman, S. Baron, and W. Levison. An optimal control model of human response part I: Theory and validation. *Automatica*, 6:357–369, 1970.

[83] J. Zico Kolter, Pieter Abbeel, and Andrew Y. Ng. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems*, pages 769–776, 2008.

[84] Gerhard Kreisselmeier. Adaptive observers with exponential rate of convergence. *IEEE Trans. Autom. Control*, 22(1):2–8, 1977.

[85] Miroslav Krstic, Ioannis Kanellakopoulos, and Peter V. Kokotovic. *Nonlinear and adaptive control design.* John Wiley & Sons, New York, NY, USA, 1995.

[86] Miroslav Krstić, Petar V. Kokotović, and Ioannis Kanellakopoulos. Transient-performance improvement with a new class of adaptive controllers. *Syst. Control Lett.*, 21(6):451–461, 1993.

[87] Peter Kühl, Moritz Diehl, Tom Kraus, Johannes P. Schlöder, and Hans Georg Bock. A real-time algorithm for moving horizon state and parameter estimation. *Comput. Chem. Eng.*, 35(1):71–83, 2011.

[88] Adam Laud and Gerald DeJong. Reinforcement learning and shaping: encouraging intended behaviors. In *ICML*, pages 355–362, 2002.

[89] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. PhD thesis, University of Illinois at Urbana-Champaign, 2004.

[90] H. I. Lee, H. S. Shin, and A. Tsourdos. Concurrent learning adaptive control with directional forgetting. *IEEE Trans. Autom. Control*, 2019.

[91] Ritchie Lee and David Wolpert. *Game theoretic modeling of pilot behavior during mid-air encounters*. Springer, 2012.

[92] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Feature construction for inverse reinforcement learning. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1342–1350. Curran Associates, Inc., 2010.

[93] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 19–27. Curran Associates, Inc., 2011.

[94] Kun Li and Joel Burdick. Online inverse reinforcement learning via bellman gradient iteration. arXiv:1707.09393, 2017.

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

[96] Mudassir Lone and Alastair Cooke. Review of pilot models used in aircraft flight dynamics. *Aerospace Sci. and Tech.*, 34:55–74, 2014.

[97] A. Loria, E. Panteley, D. Popovic, and A. R. Teel. δ-persistency of excitation: a necessary and sufficient condition for uniform attractivity. In *Proc. IEEE Conf. Decis. Control*, volume 3, pages 3506–3511, 2002.

[98] A. Loria, E. Panteley, D. Popovic, and A. R. Teel. A nested Matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems. *IEEE Trans. Autom. Control*, 50(2):183–198, 2005.

[99] A. Loría, E. Panteley, and A. Zavala-Río. Adaptive observers with persistency of excitation for synchronization of chaotic systems. *IEEE Trans. Circuits Syst.*, 56(12):2703–2716, 2009.

[100] Biao Luo, Huai-Ning Wu, Tingwen Huang, and Derong Liu. Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design. *Automatica*, 2014.

[101] Sridhar Mahadevan. Average reward reinforcement learning: foundations, algorithms, and empirical results. *Mach. Learn.*, 22(1-3):159–195, 1996.

[102] Neil J Mansfield. *Human response to vibration*. CRC press, 2004.

[103] R. Marine, G. L. Santosuosso, and P. Tomei. Robust adaptive observers for nonlinear systems with bounded disturbances. *IEEE Trans. Autom. Control*, 46(6):967–972, 2001.

[104] Duane T. McRuer and Henry R. Jex. A review of quasi-linear pilot models. *IEEE Trans. Hum. Factors Electronics*, (3):231–249, 1967.

[105] Duane T. McRuer and Ezra S. Krendel. *Mathematical Models of Human Pilot Behavior.* 1974.

[106] Jorge Mendez, Shashank Shivkumar, and Eric Eaton. Lifelong inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4502–4513. 2018.

[107] Bernard Michini and Jonathan P. How. Bayesian nonparametric inverse reinforcement learning. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524 of *Lecture Notes in Computer Science*, pages 148–163. Springer Berlin Heidelberg, 2012.

[108] Hamidreza Modares and Frank L. Lewis. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica*, 50(7):1780–1792, 2014.

[109] Hamidreza Modares, Frank L. Lewis, and Mohammad-Bagher Naghibi-Sistani. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. *Automatica*, 50(1):193–202, 2014.

[110] T. Molloy, J. Ford, and T. Perez. Online inverse optimal control on infinite horizons. In *Proc. IEEE Conf. Decis. Control*, pages 1663–1668. IEEE, 2018.

[111] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion—an inverse optimal control approach. *Auton. Robot.*, 28(3):369–383, 2010.

[112] A. Morgan and K. S. Narendra. On the uniform asymptotic stability of certain linear nonautonomous differential equations. *SIAM J. Control Optim.*, 15(1):5–24, 1977.

[113] Katharina Muelling, Abdeslam Boularias, Betty Mohler, Bernhard Schölkopf, and Jan Peters. Inverse reinforcement learning for strategy extraction. In *ECML PKDD 2013 Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA 2013)*, pages 1–9, 2013.

[114] Sriraam Natarajan, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. Multi-agent inverse reinforcement learning. In *Int. Conf. Mach. Learn. Appl.*, pages 395–400, 2010.

[115] Lawrence Nelson and Edwin Stear. The simultaneous on-line estimation of parameters and states in linear systems. *IEEE Trans. Autom. Control*, 21(1):94–98, 1976.

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

[117] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

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

[119] E. Panteley, A. Loria, and A. Teel. Relaxed persistency of excitation for uniform asymptotic stability. *IEEE Trans. Autom. Control*, 46(12):1874–1886, 2001.

[120] Anup Parikh, Rushikesh Kamalapurkar, and Warren E. Dixon. Integral concurrent learning: adaptive control with parameter convergence using finite excitation. *Int. J. Adapt. Control Signal Process.*, 33(12):1775–1787, December 2019.

[121] Hyunju Park, Byung-Yoon Lee, Min-Jea Tahk, and Dong-Wan Yoo. Differential game based air combat maneuver generation using scoring function matrix. *J. Aeronautical & Space Sci.*, 17(2):204–213, 2016.

[122] Bilal Piot, Matthieu Geist, and Olivier Pietquin. Bridging the gap between imitation learning and inverse reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 28(8):1814–1826, 2017.

[123] Sreeraman Rajan, Sichun Wang, Robert Inkol, and Alain Joyal. Efficient approximations for the arctangent function. *IEEE Signal Process. Mag.*, 23(3):108–111, 2006.

[124] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *Proc. Int. Joint Conf. Artif. Intell.*, pages 2586–2591, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[125] A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, and G. T. Huntington. Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the Gauss pseudospectral method. *ACM Trans. Math. Softw.*, 37(2):1–39, 2010.

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

[127] N. Rhinehart and K. Kitani. First-person activity forecasting from video with online inverse reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):304–317, 2018.

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

[129] N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proc. IEEE Conf. Comput. Vis.*, pages 3696–3705, 2017.

[130] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

[131] Ghananeel Rotithor, Daniel Trombetta, Rushikesh Kamalapurkar, and Ashwin P. Dani. Reduced order observer for structure from motion using concurrent learning. In *Proc. IEEE Conf. Decis. Control*, pages 6815–6820, December 2019.

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

[133] Caude Sammut. Behavioral cloning. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 93–97. Springer, Boston, MA, 2010.

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

[135] Stefan Schaal. Learning from demonstration. In M. I. Jordan and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 1040–1046. MIT Press, Cambridge, MA, 1997.

[136] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.*, 3(6):233–242, 1999.

[137] Ryan V. Self, Moad Abudia, and Rushikesh Kamalapurkar. Online inverse reinforcement learning for systems with disturbances. In *Proc. Am. Control Conf.*, pages 1118–1123, July 2020.

[138] Ryan V. Self, Michael Harlan, and Rushikesh Kamalapurkar. Online inverse reinforcement learning for nonlinear systems. In *Proc. IEEE Conf. Control Technol. Appl.*, pages 296–301, Hong Kong, China, August 2019. IEEE.

[139] Ryan V. Self, S. M. Nahid Mahmud, Katrine Hareland, and Rushikesh Kamalapurkar. Online inverse reinforcement learning with limited data. In *Proc. IEEE Conf. Decis. Control*, to appear. See also, arXiv:2008.08972.

[140] Adrian Šošić, Wasiur R. KhudaBukhsh, Abdelhak M. Zoubir, and Heinz Koeppl. Inverse reinforcement learning in swarm systems. In *Proc. Conf. Auton. Agents MultiAgent Syst.*, pages 1413–1421. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[141] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction.* MIT Press, Cambridge, MA, USA, 1998.

[142] U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. In *Proc. Int. Conf. Mach. Learn.*, pages 1032–1039. ACM, 2008.

[143] Umar Syed and Robert E. Schapire. A game-theoretic approach to apprenticeship learning. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1449–1456. Curran Associates, Inc., 2008.

[144] István Szita. Reinforcement learning in games. In *Reinforcement Learning*, pages 539–577. Springer, 2012.

[145] Alireza Tavakkoli, Richard Kelley, Christopher King, Mircea Nicolescu, Monica Nicolescu, and George Bebis. A vision-based architecture for intent recognition. *Adv. Vis. Comput.*, pages 173–182, 2007.

[146] Roberto Togneri and Li Deng. Joint state and parameter estimation for a target-directed nonlinear dynamic system model. *IEEE Trans. Signal Process.*, 51(12):3061–3070, 2003.

[147] Gerardo De La Torre, Girish Chowdhary, and Eric N. Johnson. Concurrent learning adaptive control for linear switched systems. In *Proc. Am. Control Conf.*, pages 854–859, 2013.

[148] James P. Trevelyan, Sung-Chul Kang, and William R. Hamel. Robotics in hazardous applications. In *Springer handbook of robotics*, pages 1101–1126. Springer, 2008.

[149] K. G. Vamvoudakis and F. L. Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

[150] Chunguang Wang, Feng Liao, Junwei Han, and Guixian Li. A revised optimal control pilot model for computer simulation. In *Int. Conf. Bioinform. Biomed. Eng.*, pages 844–848. IEEE, 2008.

[151] Ding Wang, Derong Liu, Hongliang Li, Biao Luo, and Hongwen Ma. An approximate optimal control approach for robust stabilization of a class of discrete-time nonlinear systems with uncertainties. *IEEE Trans. Syst. Man Cybern. Syst.*, 46(5):713–717, 2016.

[152] Rodney D Wierenga. An evaluation of a pilot model based on kalman filtering and optimal control. *IEEE Trans. Man-Mach. Syst.*, 10(4):108–117, 1969.

[153] C. Wu, X. Huang, B. Niu, and X. Xie. Concurrent learning-based global exponential tracking control of uncertain switched systems with mode-dependent average dwell time. *IEEE Access*, 6:39086–39095, 2018.

[154] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. arXiv:1507.04888, 2015.

[155] Yildiray Yildiz, Adrian Agogino, and Guillaume Brat. Predicting pilot behavior in medium-scale scenarios using game theory and reinforcement learning. *J. Guid. Control Dynam.*, 37(4):1335–1343, 2014.

[156] Jiangchuan Zheng, Siyuan Liu, and Lionel Ni. Robust bayesian inverse reinforcement learning with sparse behavior noise. In *AAAI Conf. Artif. Intell.*, 2014.

[157] Zhengyuan Zhou, Michael Bloem, and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. *IEEE Trans. Autom. Control*, 63(9):2787–2802, 2018.

[158] Xiaojin Zhu. Machine teaching: an inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087, 2015.

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

[160] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Human behavior modeling with maximum entropy inverse optimal control. In *AAAI Hum. Behav. Model.*, page 92, 2009.

VITA

Ryan Voyd Self

Candidate for the Degree of

Doctor of Philosophy

Dissertation: ON MODEL-BASED ONLINE INVERSE REINFORCEMENT LEARNING

Major Field: Mechanical and Aerospace Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2020.

Completed the requirements for the Master of Science in Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2016.

Completed the requirements for the Bachelor of Science in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2014.