UNIVERSITY OF CALIFORNIA
RIVERSIDE

Learning Robust Models for Control: Tradeoffs, Fundamental Insights, and Benchmarking Control Design

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Abed AlRahman Al Makdah

December 2023

Dissertation Committee:

Prof. Fabio Pasqualetti, Chairperson
Prof. Amit K. Roy-Chowdhury
Prof. Samet Oymak

The Dissertation of Abed AlRahman Al Makdah is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

Throughout this thesis, this has been the most challenging part to write. I feared missing the chance to express gratitude to those whose support made this work achievable, if not impossible.

My first and deepest gratitudes go to my advisor, Fabio Pasqualetti, for his invaluable guidance, unwavering support, and endless patience throughout the entirety of my PhD journey. His mentorship, insightful feedback, and encouragement have been instrumental in shaping the direction of this thesis and contributed immensely to my academic growth. Being one of his students has been an absolute privilege that I deeply cherish.

I extend my heartfelt appreciation to the members of my dissertation committee, Amit Roy-Chowdhury and Samet Oymak, for their valuable time, expertise, and constructive criticism that significantly improved the quality of this thesis.

I would like to express my sincere appreciation to my colleagues and collaborators, Vaibhav Katewa and Vishaal Krishnan, whose contributions – whether through thought-provoking discussions, mathematical analysis, or critical feedback – significantly enriched the outcomes of this thesis. Thank you Vaibhav for your invaluable insights and contributions in exploring the performance-robustness tradeoff in adversarial classification and perception-based control. Thank you Vishaal for your significant insights and contributions to the analysis of learning robust and safe controllers. Collaborating with both of you has been an enriching experience, from which I have gained crucial skills in critical thinking, problem-solving, and meticulous attention to detail. I am thankful for the opportunity to learn and develop alongside such exceptional colleagues.

I would like to extend my gratitude to my past and present lab mates. Thank you Giacomo, Rajasekhar, Gianluca, Tommaso, Karthik, and Shiqi for the insightful discussions, support, and for making my time in the lab enjoyable. Thank you Federico, Yuzhen, Darshan, Taosha, Manuel, Alberto, Silvia, Umberto, Sarbendu, and Shivanshu for the wonderful moments we shared outside the lab. Our hiking, surfing, fishing, and camping adventures were not only enjoyable but also moments of relaxation that provided refreshing breaks from our academic endeavors, fostering cherished memories.

During my stay at Riverside, I was fortunate to meet several people with whom I forged close friendships and strong connections. I want to extend my gratitude to Yahya, Rakib, Sudipta, Akbar, Awiz, Kaiqing, AbdulRahman, Tianchan, Abdullah, Eric, Sanjid, and Jonas for making my stay at UCR enjoyable.

I am deeply grateful to my master advisors at the American University of Beirut, Elie Shammas and Naseem Daher, for their unwavering guidance and encouragement, which paved the way for my academic pursuits. Their mentorship and support were instrumental in shaping my academic journey, and I am thankful for their invaluable contributions to my professional development.

Last but certainly not least, I want to express my deepest and most sincere gratitude to my family. Their unwavering support, love, and understanding have been the backbone of my journey. Their encouragement and sacrifices made my academic pursuit possible, and I am extremely grateful for their constant belief in me.

To my parents, Hala and Milad.

ABSTRACT OF THE DISSERTATION

Learning Robust Models for Control: Tradeoffs, Fundamental Insights, and
Benchmarking Control Design

by

Abed AlRahman Al Makdah

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2023
Prof. Fabio Pasqualetti, Chairperson

In the field of machine learning, the quest to optimize the *performance* of machine learning models while maintaining *robustness* against perturbations stands as a fundamental challenge. Performance of a machine learning model refers to its capacity to execute a desired task, such as classification, prediction, or generation. Conversely, robustness of a machine learning model refers to its capacity to maintain consistent and reliable performance when encountering perturbed data or data generated under unforeseen conditions.

This thesis investigates the inherent tradeoff between performance and robustness in both classification and control learning problems. Our contribution is threefold. First, we formally show that, in a quest to optimize their performance, machine learning models tend to exhibit reduced robustness against adversarial manipulation of the data. Our results suggest that this tradeoff, fundamental in nature, is deeply rooted in the way in which data is drawn and does not depend on the complexity of the learning model itself. Second, we leverage the insights acquired from this characterization of the tradeoff to establish a benchmark for learning controllers. In particular, we introduce a robust feed-back control policy learning framework based on Lipschitz-constrained loss minimization, where the feedback policies are learned directly from expert demonstrations. Our work integrates robust learning, optimal control and robust stability into a unified framework, enabling the learning of controllers that prioritize both performance and robustness. Finally, we revisit the linear quadratic Gaussian (LQG) optimal control problem through the perspective of input-output behaviors, where we derive direct data-driven expression for the optimal LQG controller using a dataset of input, state, and output trajectories. We show that our data-driven expression is consistent, since it converges as the number of experimental trajectories increases, we characterize its convergence rate, and we quantify its error as a function of the system and data properties. This analysis highlights the limitations and challenges posed by noisy data and unknown system dynamics in learning control problems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the realm of machine learning, the *performance* of a machine learning model refers to its capacity to execute a desired task, such as classification, prediction, or generation. For instance, an image classifier demonstrating high performance can accurately classify images of various objects. On the other hand, the *robustness* of a machine learning model refers to its capacity to maintain consistent and reliable performance when encountering perturbed data or data generated under unforeseen conditions. For instance, an image classifier exhibiting high robustness can maintain classification accuracy despite changes in the lightning conditions or the angle at which the image is captured.

Machine learning models are meticulously crafted and trained to maximize performance. However, this relentless pursuit of maximizing a model's performance often encounters an inevitable tradeoff with its robustness against data perturbations. This tradeoff, fundamental in nature, is deeply rooted in the way data is drawn and does not depend on the complexity of the learning model itself. For instance, as learning models strive to optimize performance based on data patterns drawn from specific conditions (e.g., images of street signs captured on clear days), they become increasingly specialized and tailored to those particular conditions represented in the data. Consequently, they may fail to adapt and generalize to novel scenarios (e.g., accurately classifying images of street signs in rainy weather) or to adversarially perturbed data samples (e.g., accurately classifying images of street signs covered with stickers).

Understanding and characterizing this fundamental tradeoff holds profound implications not only for improving the capabilities of machine learning models but also for fostering trust and dependability in their real-world deployment. Striking a delicate balance between optimizing performance and robustness is pivotal to harnessing the true potential of machine learning across diverse domains ranging from healthcare and finance to autonomous systems and natural language processing.

In this thesis, we delve into a formal examination and characterization of the fundamental tradeoffs between performance and robustness in both classification learning problems and learning control problems. Furthermore, we leverage the insights acquired from this characterization of the tradeoff to establish a benchmark for designing learning models. Where these models are engineered to attain specific performance levels while also ensuring robustness and reliability when deployed in environments that may contain perturbations or variations of the data.

1.1 Literature Synopsis

In this section, we examine the existing literature related to robust adversarial classification and robust learning for control, along with their fundamental limitations. Additionally, we provide a literature review focusing on behavioral representation for linear control systems and learning the linear quadratic Gaussian regulator from data. This comprehensive review will allow a more precise articulation of the contributions made by this thesis.

1.1.1 Fundamental limits in robust adversarial classification

Recent work has shown that classification based on neural networks is vulnerable to adversarial perturbations [35,90], and that these perturbations are universal and affect a large number of classification algorithms. While heuristic explanations of this phenomenon have been proposed, including adversarial learning [35,54,76], black-box [61], and gradient-based [35, 76], a fundamental analytical understanding of the limitations of classification algorithms under adversarial perturbations is critically lacking. We identify these limitations for a binary classification problem in a Bayesian setting. While in a simple setting, our analysis formally shows that a fundamental tradeoff exists between accuracy and sensitivity of any classification algorithm, independently of the complexity of the algorithm. The papers [73,79] are also related to this study, which derive methods to measure robustness of different classifiers against adversarial perturbations and obtain guarantees against bounded perturbations, as well as [54], which shows how adversarial training improves the classifier's performance against adversarial perturbations while deteriorating its performance under nominal conditions. Distributionally robust optimization has also been used to develop robust classifiers [86]. Yet, this theory does not formally explain the tradeoff highlighted in this work. Our approach provides rigorous support to the empirical evidence obtained in these works.

1.1.2 Fundamental limits in robust adversarial classification with abstaining

The literature on classification with an abstain option (also referred to as reject option or selective classification) mainly discusses methods on how to abstain on uncertain inputs. [9, 42] augmented the output class set with a reject class in a binary classification problem, where inputs with probability below a certain threshold are abstained on. Further, [32] used abstaining in multi-class classification problems, where abstaining was used in deep neural networks. In [106], abstaining was used in a regression learning problem. While little work has been done on using abstaining in the context of adversarial robustness, recent work has developed algorithms that guarantee robustness against adversarial attacks via abstaining [8,55], where a tradeoff between nominal performance and adversarial robustness has been observed upon tuning their algorithms. In this work, we formally prove the existence of such a tradeoff between performance and adversarial robustness, where we show that this tradeoff exist regardless of what algorithm is used to select the abstain region.

1.1.3 Fundamental limits in learning controllers for closed-loop systems

Machine learning and, more generally, data-driven algorithms have shown remarkable performance under nominal and well-modeled conditions in a variety of applications. Yet, the same algorithms have proven extremely fragile when subject to small, yet targeted, perturbations of the data [35, 90]. A detailed understanding of this unreliable behavior is still lacking, with recent theoretical results proving robustness and generalization guarantees for learning algorithms subject to adversarial disturbances, e.g., see [3, 40, 103], and showing that, in certain contexts, robustness to perturbations and performance under nominal conditions are inversely related [**?**, 25, 95, 107]. Compared to these works, we prove that a fundamental trade-off between performance and robustness arises in linear estimation algorithms, which may lead to a critical degradation of the closed loop performance [60]. Related to this work is the literature on robust control and estimation [62, 112]. However, the primary focus of this work is not on designing a robust estimator or controller, but rather on proving the existence of a fundamental trade-off between accuracy and robustness, which plays a critical role in the deployment of learning and data-driven methods in control applications.

1.1.4 Learning robust models for closed-loop systems

Robustness of data-driven models to adversarial perturbations has attracted much attention in recent years. One of the approaches to robust learning seeks to modulate the Lipschitz constant of the data-driven model [6, 20, 30], either via a regularization [15, 104] of the learning loss function or by imposing a Lipschitz constraint [37, 52]. Since the

Lipschitz constant determines the (worst-case) sensitivity of a model to perturbations of the input, data-driven models trained with Lipschitz constraints/regularizers are expected to be robust to bounded (adversarial) perturbations [90]. Prior works have primarily explored this approach for static input-output models [52, 90, 101]. However, in a feedback control setting, a static input-output robustness guarantee for a data-driven controller may not result in robust closed-loop performance. When a data-driven controller is integrated into the feedback loop, a static input-output robustness guarantee for the data-driven controller must be combined with appropriate robust control notions to yield a robustness certificate for the closed-loop system [10, 48]. Obtaining safety and robustness certificates for data-driven controllers in closed-loop systems remains an active area of research in general. In this work, we propose a learning framework to learn Lipschitz feedback policies with provable guarantees on closed-loop performance and robustness against bounded adversarial perturbation, where these policies are learned directly from expert demonstrations without any prior knowledge of the task and the system model.

Learning controllers from expert demonstrations (imitation learning)

Generalization: The key obstacle to widespread adoption of imitation learning is that it is difficult to guarantee performance in unseen scenarios. One approach to overcome this obstacle is inverse reinforcement learning (also referred to as apprenticeship learning in the literature), where the learner infers the unknown cost function from expert demonstrations, then learn an optimal policy that optimizes the learned cost using reinforcement learning [1,2,44,58,89]. Since the learned cost represents the task of the expert, inverse reinforcement learning algorithms are able to generalize to unseen scenarios that are not covered by the expert demonstrations. However, one drawback of inverse reinforcement learning is that there can exist multiple cost functions that can be optimized under the expert's policy, which adds ambiguity in learning the cost function [114]. Another approach that overcomes the obstacle of performing in unseen scenarios is direct policy learning via interactive expert [82, 83]. In this approach, the learner can query an interactive expert at each iteration, then, the learner uses the expert's feedback to correct its mistakes and improve its policy. Since this approach keeps expanding the expert's data, it will eventually cover all possible scenarios in the long run. However, one drawback of this approach is that it requires the expert to be always available for feedback. In [56], noise is injected into the expert's policy in order to provide demonstrations on how to recover from errors. In [91], the authors develop a framework for learning a generative model for planning trajectories from demonstrations, which allows it to capture uncertainty about previously unseen scenarios.

Closed-loop performance and robustness: Several approaches to adversarial imitation learning have been proposed in [51, 115], where inverse reinforcement learning is used. In [99], the authors proposed an adversarially robust imitation learning framework, where an agent is trained in an adversarially perturbed environment with the expert being available for queries at any time step. In [59], the authors learn robust control barrier functions from safe expert demonstrations. In all these works, robustness of imitation learning algorithms is considered to be the ability of the learned policy to recover from errors, which is similar to the notion of generalization.

In contrast to many of the works referenced above, we seek a principled feedback policy learning framework with strong theoretical guarantees. In particular, we seek explicit bounds on the performance, stability and robustness of the closed-loop system under the learned feedback policy. The broader problem of obtaining closed-loop performance and robustness guarantees for learned feedback policies and understanding the underlying tradeoffs has attracted attention recently [43, 65, 97]; yet it remains an active area of research. This requires the integration of theoretical tools from several areas: (i) the underlying control task is typically specified as an optimal control problem with performance measured in terms of the cost incurred, (ii) the feedback control policy is learned from finite offline data which involves considerations of generalization and robustness to distributional shifts, and (iii) closed-loop performance guarantees typically rely on an underlying robust stability guarantee for the learned policy. Prior works have addressed this problem within various frameworks, such as the H_∞-control framework for linear systems [24]. However, the problem of obtaining guarantees on closed-loop generalization and robustness to distributional shifts of learned policies for general nonlinear systems still remains a challenge. In this work, we address this problem within a Lipschitz feedback policy learning framework. The Lipschitz property is a fairly mild requirement in nonlinear control, and through our analysis we will see that it can be exploited to provide closed-loop bounds on generalization and robustness to distributional shifts for learned policies, highlighting the effectiveness of this approach.

1.1.5 Linear quadratic Gaussian regulator and behavioral representation

Behavioral representation for data-driven control

In the context of data-driven control, the behavioral approach has garnered much attention in recent years [22, 31, 53, 102], as it circumvents the need for state space representation. Owing it this fact, it belongs in the same category as the difference operator representation and ARMAX models [36, Sec. 2.3 and Sec. 7.4], and shares several connections with these classes of models. We refer the reader to [?] for a comprehensive overview of the behavioral approach.

Data-driven Linear quadratic Gaussian regulator

The LQG control problem has been studied extensively in the literature [13, 113], where fundamental properties have been characterized, such as the existence of optimal solution, how to obtain it using separation principle [113], and its lack of stability margin guarantees in closed-loop [28]. In [111], the authors characterize the optimization landscape for the LQG problem, showing that the set of stabilizing dynamic controllers can be disconnected. Different from the literature, our work seeks to represent the optimal LQG problem in the space of input-output behaviors, characterize the optimal behavioral feedback controllers, and to demonstrate their suitability for data-driven control and gradient-based methods for controller design. More specifically, we show that the optimal LQG controller can be expressed as a static behavioral-feedback gain, which underlies its advantages for developing data-driven methods to learn LQG controllers.

Data-driven methods for system analysis and control have flourished in the last years and are revolutionizing the field [80]. The methods developed in this thesis fall in the category of direct data-driven methods [7], where controls are obtained directly from data bypassing the classic system identification step [33]. In line with earlier work and differently from optimization-based approaches [26, 45], we pursue here closed-form data-driven expressions, which are typically computationally advantageous [16], are transparent, and can reveal novel insights into the problems [18].

This work focuses on data-driven LQG control, while most of the literature on data-driven control has focused on the LQR problem with noiseless data [21, 71, 84]. Recent work [19] has studied the design of data-driven controllers from noisy data [38, 78], the design of data-driven Kalman filters [108], imitation-based LQG control design [39], and some versions of the output-weighted LQG control problem [29, 87]. Compared to [87], in particular, in our work, we do not assume perfect knowledge of the Markov parameters or any part of the system dynamics and noise, and it does not estimate them to solve the state-weighted LQG problem. To the best of our knowledge, this thesis contains the first direct, closed-form data-driven solution to the state-weighted LQG problem, with finite-sample performance guarantees.

The recent literature on the analysis of the sample complexity of estimation and control problems is also relevant to this work. In particular, [23, 109] follow an indirect approach, where sample complexity bounds are derived for the identification of the system dynamics and such errors are propagated towards the design of LQR and LQG controllers. Differently from our work, this analysis is valid only for stable systems and output-weighted LQG costs. Bounds on the performance of the learned LQG controller are also derived in [70] assuming a sufficiently small error in the system identification step [75,93,96,110], and in [72] where the optimal LQR is learned in a model-free setting using gradient methods. Although our work makes use of similar technical tools, the approach pursued here is direct and does not rely on the identification of the system matrices, nor on optimization algorithms to design or tune robust controllers. Further, this work considers the canonical LQG setting, rather than the noisy LQR problem or the output-weighted LQG problem with noisy controls, and it provides closed-form expressions for the optimal controllers rather than their performance.

1.2 Contributions of this Thesis

The main contributions of each chapter are as follows.

Chapter 2. This chapter features three main contributions. First, we propose metrics to quantify the accuracy of a classification algorithm and its sensitivity to arbitrary manipulation of the data. We prove that, under a set of mild technical assumptions, the accuracy of a classification algorithm can only be maximized at the expenses of its sensitivity. Thus, a fundamental tradeoff exists between the performance of a classification algorithm in nominal and adversarial settings. While our results formally apply to binary classification problems, we conjecture that this fundamental tradeoff in fact applies to more general classification problems. Second, we show that a tradeoff between accuracy and sensitivity exists for different classes of classification algorithms, and that simpler algorithms can sometimes outperform more complex one in adversarial settings. Third, we numerically show that the accuracy versus sensitivity tradeoff depends solely on the statistics of the data, and cannot be arbitrarily improved by tuning the classification algorithm (varying classification boundaries) or increasing its complexity (number of boundaries), including using sophisticated adversarial learning techniques. Taken together, our results suggest that performance and robustness of data-driven algorithms are dictated by the properties of the data, and not by the sophistication or intelligence of the algorithm, a key insight that has critical implications for the deployment of provably-robust data-driven and learning-based control algorithms.

Chapter 3. This chapter features three main contributions. First, we propose metrics to quantify the performance of a classifier with an abstain option and its adversarial robustness. Second, we show that for a binary classification problem with an abstain option, a tradeoff between performance and adversarial robustness always exist regardless of which region of the input space is abstained on. Thus, the robustness of a classifier with an abstain option can only be improved at the expense of its nominal performance. Further, we numerically show that such a tradeoff exist for the general multi-class classification problems. The type of the tradeoff we present in this chapter is different than the one studied in the literature [64,95,107], degrading the nominal performance implies that the classifier abstains more often on nominal inputs, and it does not imply an increase in the misclassification rate. Third, we provide necessary conditions to optimally design the abstain region for a given classifier for the 1-dimensional binary classification problem.

Chapter 4. This chapter features two main contributions. First, we study a perception-based control problem, where the state of a dynamical system is reconstructed using a high-dimensional sensor. We prove the existence of a fundamental trade-off between the accuracy of the estimation algorithm, as measured by its minimum mean squared error, and its robustness to variations and inaccuracies of the data statistics. Thus, (i) estimation algorithms that are optimal for the nominal data tend to perform poorly in practice, where the operating conditions may differ from the nominal data, and, conversely, (ii) estimation algorithms that are robust to data variations exhibit suboptimal performance in nominal conditions. Second, we characterize estimators that lie on the Pareto frontier between accuracy and robustness, that is, estimators that are maximally robust for a desired performance level, and estimators that are maximally accurate for a given bound on the data variations and inaccuracies. We also show, numerically, that the trade-off for estimation algorithms also affects the performance of the closed-loop system, and even when the measurement error is not normally distributed, as we assume for the derivation of our analytical results.

In a broader context, the results of this chapter further characterize a fundamental limitation of machine learning and data-driven algorithms, as described for different settings in [**?**, 25, 95, 107], and clarify its implications for control applications.

Chapter 5. Our primary contribution in this chapter is a robust feedback control policy learning framework based on Lipschitz-constrained loss minimization, where the feedback policies are learned directly from expert demonstrations. We then undertake a systematic study of the performance of feedback policies learned within our framework using meaningful metrics to measure closed-loop stability, performance and robustness. Our work integrates robust learning, optimal control and robust stability into a unified framework for robust feedback policy learning. More specifically, our work features three main technical contributions. First, we derive a robust stability bound for the closed-loop system under the learned feedback policies, which guarantees that the closed-loop trajectory under the learned policy stays within a bounded region around the expert trajectory and converges asymptotically to a bounded region around the origin. Second, we derive a bound on the regret incurred by learned feedback policies with respect to the expert policy in terms of the generalization error (learning error), and a bound on the deterioration of closed-loop performance in the presence of (adversarial) disturbances to state measurements. These bounds provide certificates for closed-loop performance and adversarial robustness for learned policies. Third, we provide an analysis of the Lipschitz-constrained policy learning problem, which results in a (probabilistic) bound on generalization error for the learned policies. This sheds light on the dependence of closed-loop control performance and robustness on learning. Conversely, our results specify target bounds on policy generalization error (learning error) for desired closed-loop performance. We then demonstrate our robust feedback policy learning framework via numerical experiments on (i) the standard LQR benchmark, and (ii) a non-holonomic differential drive mobile robot model. Finally, our analysis support the existence of a potential tradeoff between nominal performance of the learned policies and their robustness to adversarial disturbances of the feedback, which is borne out in numerical experiments where we observe that improvements to adversarial robustness can only be made at the expense of nominal performance.

Chapter 6. This chapter features three main contributions. First, we introduce equivalent representations for stochastic discrete-time, linear, time-invariant systems and the LQG optimal control problem in the behavioral space. Second, we show that, in the behavioral space, the optimal LQG controller can be expressed as a static behavioral-feedback gain, which can be solved for directly from the LQG problem represented in the behavioral space. Third, we highlight the advantages of having a static feedback LQG gain over a dynamic LQG controller in the context of data-driven control and gradient-based algorithms.

Chapter 7. The main contributions of this chapter is the characterization of direct data-driven formulas for the LQR gain, Kalman filter, and LQG gain using a dataset of trajectories of the input, state, and output of a discrete-time linear time-invariant system. Importantly, since the experimental data is noisy and the system dynamics and noise statistics are unknown, we show that our formulas are consistent, as they converge to the true expressions when the amount of experimental data increases. Additionally, we characterize the convergence rate of our expressions, as well as their error when the data is of finite size. Finally, we provide illustrative examples and remarks to highlight how the properties of the system and of the experimental data affect the accuracy of our formulas.

Part I

Unveiling Fundamental limits: Performance vs Robustness Tradeoff in Data-Driven Models

Chapter 2

Fundamental Performance - Robustness Tradeoff for Adversarial Classification

In this chapter, we formally study a fundamental tradeoff between the accuracy of a binary classification algorithm and its sensitivity to arbitrary manipulation of the data. In particular, we cast a binary classification problem into a hypothesis testing framework, parametrize classification algorithms – including those based on machine learning techniques – using their decision boundaries, and show that the accuracy of the algorithm can be maximized only at the expenses of its sensitivity. This tradeoff, which applies to general classification algorithms, depends on the statistics of the data, and cannot be improved by simply tuning the algorithm. Our theory explains how simple algorithms can outperform more complex ones when operating in adversarial environments. The results of this chapter are reported in our published paper [64].

2.1 Problem setup and preliminary notions

To reveal a fundamental tradeoff between the accuracy of a classification algorithm and its robustness against malicious data manipulation, we consider a binary classification problem where the objective is to decide whether a scalar observation $x \in \mathbb{R}$ belongs to one of the classes $\mathcal{H}_0$ and $\mathcal{H}_1$. We assume that the distribution of the observations satisfy

$$\mathcal{H}_0 : x \sim f_0(x;\theta_0), \text{ and } \mathcal{H}_1 : x \sim f_1(x;\theta_1), \tag{2.1}$$

where $f_0(x;\theta_0)$ and $f_1(x;\theta_1)$ are arbitrary, yet known, probability density functions with parameters $\theta_0 \in \mathbb{R}^{m_0}$ and $\theta_1 \in \mathbb{R}^{m_1}$, respectively. We assume that the partial derivatives of f_k with respect to x and θ_k exist and are continuous over the domain of the distributions, for $k = 0, 1$. Let p_0 and p_1 denote the prior probabilities of the observations belonging to the classes $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. Different (machine learning) algorithms can be used to solve the above binary classification problem. Yet, because of the binary nature of the problem, any classification algorithm can be represented by a suitable partition of the real line, and it can be written as

$$\mathfrak{C}(x;y) = \begin{cases} \mathcal{H}_0, & x \in \mathcal{R}_0, \\ \mathcal{H}_1, & x \in \mathcal{R}_1, \end{cases} \tag{2.2}$$

where[1] $y = [y_i]$ denotes a set of boundary points, with $y_0 \leq \cdots \leq y_{n+1}$, $y_0 = -\infty$, $y_{n+1} = \infty$, and

$$\mathcal{R}_0 = \{z \ : \ y_i < z < y_{i+1}, \text{ with } i = 0, 2, \dots, n\},$$
$$\mathcal{R}_1 = \{z \ : \ y_i \leq z \leq y_{i+1}, \text{ with } i = 1, 3, \dots, n-1\}.$$

We refer to (2.2) as general classifier. We measure the performance of a classification algorithm through its *accuracy*, that is, its probability of making a correct classification.

Definition 1 ***(Accuracy of a classifier)*** *The accuracy of the classification algorithm* $\mathfrak{C}(x; y)$ *is*

$$\mathcal{A}(y; \theta) = p_0 \mathbf{P}\left[x \in \mathcal{R}_0 | \mathcal{H}_0\right] + p_1 \mathbf{P}\left[x \in \mathcal{R}_1 | \mathcal{H}_1\right], \tag{2.3}$$

where $\theta = [\theta_0^\mathsf{T} \ \theta_1^\mathsf{T}]^\mathsf{T}$ *contains the distribution parameters.* □

Using Equation (2.3) and the distributions in (2.1), we obtain

$$\begin{aligned}\mathcal{A}(y; \theta) = p_0 &\left(\sum_{l=1}^{n} (-1)^{l+1} \int_{-\infty}^{y_l} f_0(x; \theta_0) dx + 1 \right) \\ &+ p_1 \left(\sum_{l=1}^{n} (-1)^{l} \int_{-\infty}^{y_l} f_1(x; \theta_1) dx \right).\end{aligned} \tag{2.4}$$

Clearly, the accuracy of a classification algorithm depends on the position of its boundaries, which can be selected to maximize the accuracy of the classification algorithm. To this aim, let $L(x)$ denote the Likelihood Ratio defined as

$$L(x) = \frac{p_1 f_1(x; \theta_1)}{p_0 f_0(x; \theta_0)}.$$

[1] For simplicity and without affecting generality, we assume that n is even. Further, an alternative configuration of the classifier (2.2) assigns $\mathcal{H}_0$ and $\mathcal{H}_1$ to $\mathcal{R}_1$ and $\mathcal{R}_0$, respectively. However, because accuracy and sensitivity of the two configurations can be obtained from each other, we consider only the configuration in (2.2) without affecting the generality of our analysis.

The Maximum Likelihood (ML) classifier is

$$\mathfrak{C}_{\mathrm{ML}}(x;\eta) = \begin{cases} \mathcal{H}_0, & L(x) < \eta, \\ \mathcal{H}_1, & L(x) \geq \eta, \end{cases} \tag{2.5}$$

where the threshold $\eta > 0$ is a design parameter that determines the boundary points and, thus, the accuracy of the classifier. As a known result in statistical hypothesis testing [85], the accuracy of the ML classifier with $\eta = 1$ is the largest among all possible classifiers. The value and the number of boundary points of the ML classifier depend on the distributions $f_0(x;\theta_0)$ and $f_1(x;\theta_1)$, the threshold η, and the prior probabilities through the equation

$$p_1 f_1(x;\theta_1) - \eta p_0 f_0(x;\theta_0) = 0. \tag{2.6}$$

Another important class of classifiers is the class of linear classifiers, which are less complex and often achieve a competitive performance compared to nonlinear classifiers (see [105] for more details). In our setting, a linear classifier consists of one decision boundary $y \in \mathbb{R}$, and is given by

$$\mathfrak{C}_{\mathrm{L}}(x;y) = \begin{cases} \mathcal{H}_0, & x < y, \\ \mathcal{H}_1, & x \geq y. \end{cases} \tag{2.7}$$

Following Definition 1, the accuracy of $\mathfrak{C}_{\mathrm{L}}$ is

$$\mathcal{A}(y;\theta) = p_0 \int_{-\infty}^{y} f_0(x;\theta_0)dx - p_1 \int_{-\infty}^{y} f_1(x;\theta_1)dx + p_1. \tag{2.8}$$

The optimal boundary y_{L}^* that maximizes $\mathcal{A}(y;\theta)$ is

$$\begin{aligned} y_{\mathrm{L}}^* = \arg\max_{y_i} \quad & \mathcal{A}(y_i;\theta) \\ \text{s.t.} \quad & y_i \text{ is a solution of (2.6) with } \eta = 1. \end{aligned} \tag{2.9}$$

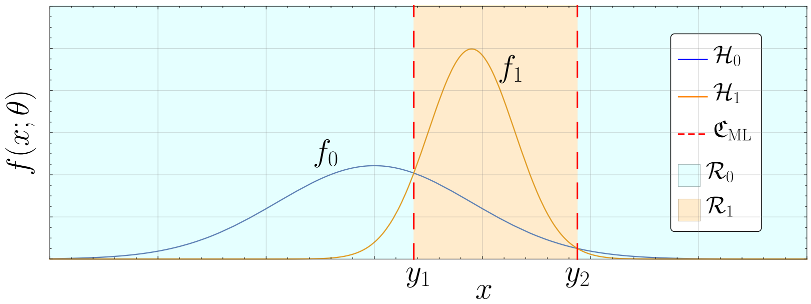


Figure 2.1: The distributions of x under Gaussian hypotheses with $\mu_0 = 0$, $\sigma_0 = 9$, $\mu_1 = 9$, $\sigma_1 = 4$, and $p_0 = p_1 = 0.5$. The dashed red lines represent the decision boundaries of the $\mathfrak{C}_{\mathrm{ML}}(x; \eta = 1)$, which divide the space into $\mathcal{R}_0$ (represented by the blue region) and $\mathcal{R}_1$ (orange region).

While the boundaries are difficult to compute for general distributions, they can be computed explicitly when the observations are Gaussian (see below). Let $\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ be the p.d.f. of a normal random variable with mean μ and variance σ, and $Q(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx$ the c.d.f. of the standard normal distribution.

Remark 2 ***(ML and linear classifiers for Gaussian distributions)*** *For the Gaussian distributions $f_i(x; \theta_i) = \mathcal{N}(x; \mu_i, \sigma_i)$, $i = 0, 1$, the boundaries of ML classifier satisfy*

$$\begin{aligned} & ax^2 + bx + c = 0 \qquad \textit{where,} \\ & a = \frac{1}{2}\left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}\right), b = \left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_0}{\sigma_0^2}\right), \; \textit{and} \\ & c = \log\left(\frac{\sigma_0}{\sigma_1}\right) + \log\left(\frac{p_1}{p_0}\right) + \frac{\mu_0^2}{2\sigma_0^2} - \frac{\mu_1^2}{2\sigma_1^2} - \log(\eta). \end{aligned} \tag{2.10}$$

Equation (2.10) has at most two real solutions, implying that the ML classifier has at most two decision boundaries (see Fig. 2.1). The ML classifier with boundaries corresponding to the solutions of (2.10) with $\eta = 1$ has maximum accuracy [85]. The solution of (2.10) which maximizes the accuracy in (2.8) is the boundary for the optimal linear classifier. □

In this paper, we consider adversarial manipulations of the observations in which an attacker is capable of adding deterministic or random perturbations to the observations in order to degrade the performance of the classifier. We model such manipulations as modification to the parameters of distributions in (1), i.e., the attacker can change the parameter θ. To characterize the robustness of a classifier to these adversarial manipulations of the observations, we define the following sensitivity metric, which captures the degradation of the classification accuracy following data manipulation.

Definition 3 ***(Sensitivity of a classifier)*** *The sensitivity of the classification algorithm*[2] $\mathfrak{C}(x; y)$ *is*

$$\mathcal{S}(y; \theta) = \left\| \frac{\partial \mathcal{A}(y; \theta)}{\partial \theta} \right\|_{\infty}, \tag{2.11}$$

where θ contains the parameters of the distributions in (2.1), and $\mathcal{A}(y; \theta)$ denotes the accuracy of $\mathfrak{C}(x; y)$. □

From Definition 3, a higher value of sensitivity implies that the adversary can affect the classifier's performance to a larger extent, whereas a lower sensitivity implies that the classifier is more robust to adversarial manipulation. Further, the $\infty-$norm captures the worst case in terms of the largest sensitivity with respect to the components of θ. Finally, the sensitivity vector $\frac{\partial \mathcal{A}(y;\theta)}{\partial \theta}$ can be used to determine a perturbation to θ that maximizes (locally) the degradation of the classifier.

Remark 4 ***(Comparison with adversarial classification)*** *In adversarial classification, the attacker designs a perturbation for a given observation (e.g., an image) to induce*

[2] Definition 3 is also valid for the ML and the linear classifier.

misclassification [90], [54]. Such observation can be viewed as a realization of a multi-dimensional distribution. In contrast, we consider perturbations of the distribution, which affect all the realizations, and focus on the average reduced performance of the classifier over all realizations. Despite this difference, our sensitivity vector and its norm capture the direction and the extent of the worst-case perturbation, similar to the worst-case smallest perturbation in adversarial classification, allow us to obtain formal guarantees, and provide additional insight into the performance limitations of adversarial classification. □

Remark 5 ***(Accuracy and sensitivity of the ML classifier for Gaussian distributions)*** *The accuracy and the sensitivity of the ML classifier are obtained by substituting the expression of the normal distributions $\mathcal{N}(x;\mu_i,\sigma_i)$ in (2.3) and (2.11):*

$$\begin{aligned}\mathcal{A}(y;\theta) &= p_0\Big(Q\Big(\frac{y_1-\mu_0}{\sigma_0}\Big) - Q\Big(\frac{y_2-\mu_0}{\sigma_0}\Big) + 1\Big)\\ &\quad + p_1\Big(-Q\Big(\frac{y_1-\mu_1}{\sigma_1}\Big) + Q\Big(\frac{y_2-\mu_1}{\sigma_1}\Big)\Big) \textit{ and,}\\ \mathcal{S}(y;\theta) &= \left\| \begin{bmatrix} p_0\Big(f_0(y_2;\theta_0) - f_0(y_1;\theta_0)\Big) \\ p_0\Big(\frac{\mu_0-y_1}{\sigma_0} f_0(y_1;\theta_0) - \frac{\mu_0-y_2}{\sigma_0} f_0(y_2;\theta_0)\Big) \\ p_1\Big(f_1(y_1;\theta_1) - f_1(y_2;\theta_1)\Big) \\ p_1\Big(\frac{\mu_1-y_2}{\sigma_1} f_1(y_2;\theta_1) - \frac{\mu_1-y_1}{\sigma_1} f_1(y_1;\theta_1)\Big) \end{bmatrix} \right\|_\infty ,\end{aligned}$$

where $\theta_i = [\mu_i \; \sigma_i]^\mathsf{T}$ and $i = 0, 1$. □

A classification algorithm should have high accuracy and low sensitivity, so as to exhibit robust performance against adversarial manipulation. Unfortunately, we show that accuracy and sensitivity are directly related, so that optimizing the accuracy of a classifier inevitably increases its sensitivity.

2.2 A fundamental tradeoff between accuracy and sensitivity of classification algorithms

In this section, we characterize a tradeoff between accuracy and sensitivity of a classification algorithm for the binary classification problem in (2.1). We prove that, under some mild conditions, there exist a classifier that is less accurate than $\mathfrak{C}_{\mathrm{ML}}(x;1)$, yet more robust to adversarial manipulation of the data. This shows that there exist a tradeoff between accuracy and sensitivity at the configuration of maximum accuracy.

Let $y^* = [y_1^* \ y_2^* \ \cdots \ y_n^*]^\mathsf{T}$ be the vector of the boundaries of $\mathfrak{C}_{\mathrm{ML}}(x;1)$, which maximizes $\mathcal{A}(y;\theta)$. Let $\theta^{(i)}$ be the i^{th} component of θ. We make the following assumptions:

A1: The vector $\frac{\partial \mathcal{A}(y;\theta)}{\partial \theta}\Big|_{y^*}$ has a unique largest absolute element, located at index j.

A2: There exist at least one boundary y_i^* such that

$$\left(p_0 \frac{\partial}{\partial y_i} f_0(y_i;\theta_0)\Big|_{y_i^*} - p_1 \frac{\partial}{\partial y_i} f_1(y_i;\theta_1)\Big|_{y_i^*} \right) \frac{\partial y_i^*}{\partial \theta^{(j)}} \neq 0.$$

Assumptions A1 is specific to our definition of sensitivity in (2.11), and is not required if 2−norm is used (see Remark 10). Further, A2 is mild and typically satisfied in most problems.

Theorem 6 ***(Accuracy-sensitivity tradeoff for general classifier*** (2.2)***)*** *Let y^* contain the boundaries of the classifier $\mathfrak{C}_{ML}(x;1)$. Then, under Assumptions A1 and A2, it holds*

$$\frac{\partial \mathcal{S}(y;\theta)}{\partial y}\Big|_{y^*} \neq 0. \tag{2.12}$$

Proof. Assumption A1 guarantees that $\mathcal{S}(y;\theta)$ is differentiable with respect to y at y^*. Let $g\big(y;\theta\big) \triangleq \frac{\partial \mathcal{A}(y;\theta)}{\partial y}$. Since y^* maximizes $\mathcal{A}(y;\theta)$, $g\big(y^*;\theta\big) = 0$. Differentiating $g\big(y^*;\theta\big)$ with respect to $\theta^{(j)}$, and noting that y^* depends on θ, we get:

$$\begin{aligned} \frac{\mathsf{d}g\big(y^*;\theta\big)}{\mathsf{d}\theta^{(j)}} &= \frac{\partial g\big(y;\theta\big)}{\partial \theta^{(j)}}\bigg|_{y^*} + \frac{\partial g\big(y;\theta\big)}{\partial y}\bigg|_{y^*}\frac{\partial y^*}{\partial \theta^{(j)}} = 0, \\ \Rightarrow \frac{\partial}{\partial y}\frac{\partial \mathcal{A}(y;\theta)}{\partial \theta^{(j)}}\bigg|_{y^*} &= -\frac{\partial^2 \mathcal{A}(y;\theta)}{\partial y^2}\bigg|_{y^*}\frac{\partial y^*}{\partial \theta^{(j)}}, \end{aligned} \tag{2.13}$$

where the last equation follows by substituting $g\big(y;\theta\big) = \frac{\partial \mathcal{A}(y;\theta)}{\partial y}$ and switching the order of partial differentiation. Using (2.11), it can be easily observed that the left side of (2.13) equals $\pm\frac{\partial \mathcal{S}(y;\theta)}{\partial y}\Big|_{y^*}$. Further, differentiating (2.4) twice, we get $\frac{\partial^2}{\partial y^2}\mathcal{A}(y;\theta) = \mathrm{diag}(w_1(y_1), \cdots, w_n(y_n))$, where

$$w_i(y_i) = p_0(-1)^{i+1}\frac{\partial}{\partial y_i}f_0(y_i;\theta_0) + p_1(-1)^i\frac{\partial}{\partial y_i}f_1(y_i;\theta_1).$$

Assumption A2 guarantees that there exist a boundary y_i^* such that $w_i(y_i^*)\frac{\partial y_i^*}{\partial \theta^{(j)}} \neq 0$. The result follows from (2.13). ■

Theorem 6 implies that the sensitivity of the classifier $\mathfrak{C}(x;y)$ can be decreased by modifying the boundaries y^*. Yet, because $\mathfrak{C}(x;y^*)$ exhibits the largest classification accuracy among all classifiers, the reduction of sensitivity inevitably decreases the accuracy of classification. In other words, for any classification problem (2.1) satisfying Assumption A1 and A2 and for any classification algorithm (2.2), there exists an arbitrarily small δ such that[3]

$$\mathcal{S}(y^*+\delta;\theta) < \mathcal{S}(y^*;\theta) \text{ and } \mathcal{A}(y^*+\delta;\theta) \leq \mathcal{A}(y^*;\theta).$$

[3] The inequality for accuracy is strict for most distributions.

Thus, a fundamental tradeoff exists between the accuracy of a classifier and its robustness to adversarial manipulation. Note that the result of Theorem 6 holds for all distributions that satisfy Assumptions A1 and A2. Further, we show next that such tradeoff also exists for linear and ML classifiers, and for multi-dimensional digit classifier based on a neural network (Section 4.3). This tradeoff is observed for a large class of problems, thereby highlighting its fundamental nature.

Corollary 7 ***(Accuracy-sensitivity tradeoff for the linear classifier*** (2.7)***)*** *Let y_L^* be the boundary given in* (2.9) *that maximizes the accuracy (in* (2.8)*) of the linear classifier $\mathfrak{C}_L(x; y)$. Then, under Assumptions A1 and A2, it holds*

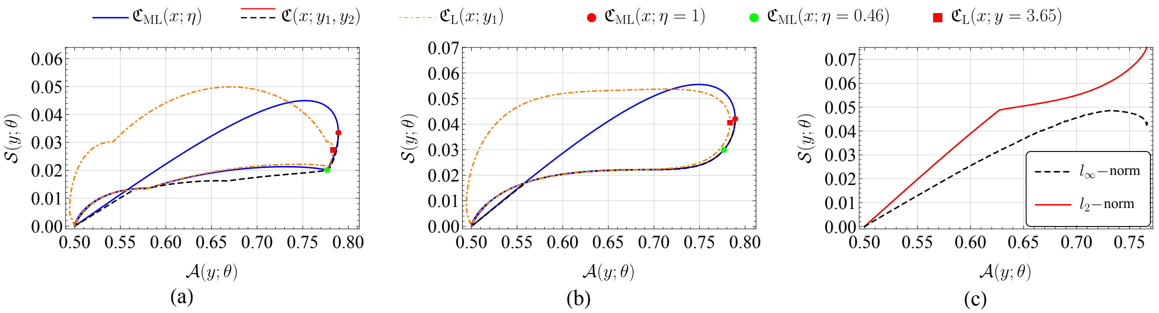
$$\left.\frac{\partial \mathcal{S}(y;\theta)}{\partial y}\right|_{y_L^*} \neq 0. \tag{2.14}$$

Proof. Since y_{L}^* corresponds to one of the boundaries contained in y^*, the proof follows from Theorem 6. ■

Next, we show that this tradeoff also exists for the Maximum Likelihood classifier. This fact does not follow trivially from Theorem 6, because the general classifier in theorem has independent boundaries, while the boundaries of the ML classifier are dependent on one another via (2.6). We make the following mild technical assumption.

A3: The vectors $\left.\frac{\partial y(\eta,\theta)}{\partial \eta}\right|_{\eta=1}$ and $\left.\frac{\partial \mathcal{S}(y;\theta)}{\partial y}\right|_{y^*}$ are not orthogonal, where $y(\eta, \theta)$ contains the boundaries of $\mathfrak{C}_{\mathrm{ML}}(x; \eta)$.

Lemma 8 ***(Accuracy-sensitivity tradeoff for the ML classifier*** (2.5)***)*** *Let $y(\eta, \theta)$ contain the boundaries of the classifier $\mathfrak{C}_{ML}(x; \eta)$. Then, under Assumptions A1, A2 and*



(a) (b) (c)

Figure 2.2: Accuracy-sensitivity tradeoff curves for a general classifier with 2 boundaries (black dashed line), the ML classifier (blue line), and a linear classifier (orange dash-dotted line) corresponding to the Gaussian hypothesis testing problem. The parameters of the two distributions for Fig. 2.2(a)-(b) are $\mu_0 = 0$, $\sigma_0 = 9$, $\mu_1 = 9$, and $\sigma_1 = 4$, and for Fig. 2.2(c) are $\mu_0 = 0$, $\sigma_0 = 4$, $\mu_1 = 5$, and $\sigma_1 = 3$. The red dot represents $\mathfrak{C}_{\mathrm{ML}}(x;1)$ (maximum accuracy point) and the green dot represents $\mathfrak{C}_{\mathrm{ML}}(x;0.46)$. The red square represents $\mathfrak{C}_{\mathrm{L}}(x;y=3.65)$, which is the linear classifier with maximum accuracy. The sensitivity in 2.2(a) and for the black dashed line in 2.2(c) is computed using Definition 3, while the sensitivity in Fig. 2.2(b) and for the red line in 2.2(c) is computed using (2.16).

A3, it holds

$$\left.\frac{\partial \mathcal{S}\left(y(\eta,\theta);\theta\right)}{\partial \eta}\right|_{\eta=1} \neq 0.$$

Proof. Let y^* contain the boundaries of the classifier $\mathfrak{C}_{\mathrm{ML}}(x;\eta=1)$. The derivative of $\mathcal{S}\big(y(\eta,\theta);\theta\big)$ with respect to η can be written as:

$$\left.\frac{\partial \mathcal{S}\big(y(\eta,\theta);\theta\big)}{\partial \eta}\right|_{\eta=1} = \left.\frac{\partial \mathcal{S}\big(y;\theta\big)}{\partial y^{\mathsf{T}}}\right|_{y^*} \left.\frac{\partial y(\eta,\theta)}{\partial \eta}\right|_{\eta=1}.$$

We conclude following Theorem 6 and Assumption A3. ■

In what follows we numerically show that a tradeoff between accuracy and sensitivity also exists when the classification boundaries are not selected to maximize the accuracy of the classifier. To this aim, first we compute the accuracy and sensitivity of the ML classifier $\mathfrak{C}_{\mathrm{ML}}(x;\eta)$, for different values of η. Notice that, by varying $0 < \eta < \infty$, Equation (2.6) returns different classification boundaries and, thus, different classification algorithms. Similarly, we compute the accuracy and sensitivity of linear classifier $\mathfrak{C}_{\mathrm{L}}(x; y)$ by varying the single boundary y. Second, we numerically solve

$$\begin{aligned} \min_{y} \quad & \mathcal{S}(y; \theta) \\ \text{s.t.} \quad & \mathcal{A}(y; \theta) = \zeta, \end{aligned} \tag{2.15}$$

for different values of ζ ranging from 0.5 to $\mathcal{A}(y^*; \theta)$. Notice that the minimization problem (2.15) returns the classifier with lowest sensitivity and accuracy equal to ζ, and that the boundaries solving the minimization problem (2.15) may not satisfy (2.6). Further, for a given number of classification boundaries, the minimization problem (2.15) returns a fundamental tradeoff curve relating accuracy and sensitivity over the range of ζ, which is independent of the choice of classification algorithm. Finally, the minimization problem (2.15) is not convex, because of its nonlinear equality constraint.

Fig. 2.2(a) shows the accuracy-sensitivity tradeoff for the Gaussian hypothesis testing problem discussed in Remark 5. In this case, since the ML classifier has 2 boundaries, we also consider general classifiers with 2 boundaries. We observe that the general classifier exhibits the tradeoff at the maximum accuracy point (identified by the red dot) in accordance with Theorem 6. Several comments are in order. First, the ML and linear classifiers also exhibit tradeoff at their respective maximum accuracy points in accordance with Lemmas 8 and 7. Second, the tradeoff for the ML classifier is not strict and there exist points where reducing accuracy increases sensitivity (green dot in the figure). On the other hand, the tradeoff for the general classifier is strict. This might be because the decision boundaries of the general classifier can be varied independently, whereas the boundaries of the ML classifier are related to each other since they are the solutions of (2.6). Thus, the general classifier provides more flexibility in choosing the boundaries, which induces lower sensitivity as compared to the ML classifier, and ultimately, results in a strict tradeoff. Similarly, the tradeoff for the linear classifier is not strict. Third, the tradeoff curve for the general classifier is below the tradeoff curves for the ML and linear classifiers, again, due to the aforementioned reason.[4] Fourth, the maximum accuracy of the linear classifier (corresponding to red square) is smaller than that of the ML classifier (corresponding to the red dot), but its sensitivity at the maximum accuracy configuration is also smaller than that of the ML classifier. This explains the observed phenomena that in some cases, linear models are more robust to adversarial attacks than nonlinear models (for example, neural networks) [34]. Finally, the curves are not smooth because of the ∞-norm in Definition 3.

Next, we present two remarks on using the 2-norm to define sensitivity and on the necessity of Assumption A1.

Remark 9 ***(Classification sensitivity using the*** $2-$***norm)*** *In Definition 3, the* ∞*-norm captures the largest change in accuracy with respect to a change in a single component of parameters vector* θ*. Instead, using the* 2*-norm to define the sensitivity of a classification algorithm leads to*

$$\mathcal{S}(y;\theta) = \left\| \frac{\partial \mathcal{A}(y;\theta)}{\partial \theta} \right\|_2, \tag{2.16}$$

which captures the change in accuracy with respect to changes in all the components of θ*. Fig. 2.2(b) shows the sensitivity versus accuracy tradeoff when sensitivity is defined using* (2.16) *instead of* (2.11)*. For this case, a strict tradeoff is observed for all classifiers,*

[4] ML and linear classifiers are particular instances of the general classifier.

although this may not be the case in general. Further, the tradeoff curves are smooth. □

Remark 10 ***(Necessity of Assumption A1)*** *Assumption A1 is required to ensure differentiability of the sensitivity in (2.11), and thus, it is required for Theorem 6. In contrast, the sensitivity defined in (2.16) is always differentiable, and A1 is not required in this case. We illustrate this in Fig. 2.2(c), where the vector* $\frac{\partial \mathcal{A}(y^*;\theta)}{\partial \theta} = [0.043,\ 0.024,\ -0.043,\ 0.040]^{\mathsf{T}}$ *has two elements with maximum absolute value, violating Assumption A1. We observe that a tradeoff at the maximum accuracy point (denoted by the red dot) does not exist in this case using (2.11), while it still exists using (2.16).* □

Next, we numerically analyze the effect of the complexity (determined by the number of boundaries) of the general classifier on the tradeoff. Fig. 3 shows the tradeoff curves corresponding to $\infty-$norm sensitivity for different number of boundaries. Ideally, the tradeoff should improve as the number of boundaries increase. Interestingly, we observe that, for high values of accuracy (> 0.72), increasing the number of boundaries does not improve the tradeoff, and all curves for $9 \geq n \geq 4$ coincide. For low values of accuracy, we face numerical difficulties in obtaining the global minimum of (15), and therefore, we do not observe smooth and ordered points on the curve. However, we still observe that the curves are close to each other, and the tradeoff does not seem to improve beyond a certain number of boundaries. Based on this, we conjecture that there exists a fundamental tradeoff curve which cannot be improved by increasing the number of boundaries arbitrarily.

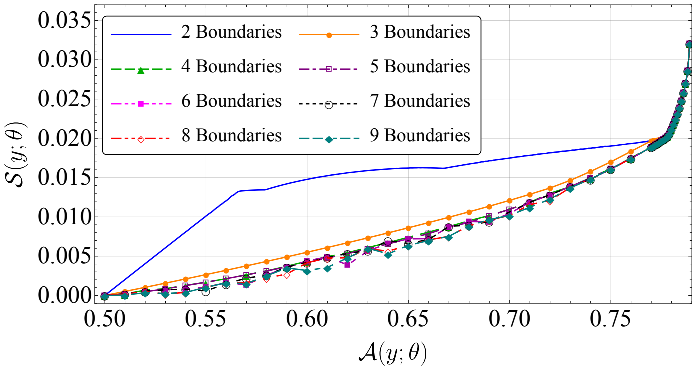


Figure 2.3: Accuracy-sensitivity tradeoff curves for general classifiers with different number of boundaries for the Gaussian hypothesis testing problem. The parameters of the distributions are $\mu_0 = 0$, $\sigma_0 = 9$, $\mu_1 = 9$, $\sigma_1 = 4$.

2.3 Illustrative examples

In this section we illustrate numerically the implications of Theorem 6. In particular, we consider two classification algorithms with different accuracy and sensitivity, and show how their performance degrades differently when the observations are corrupted by an adversary. This implies that, when robustness to adversarial manipulation of the observations is a concern, classification algorithms should be designed to simultaneously optimize accuracy and sensitivity, and should not operate at their point of maximum accuracy.

Consider the classification problem (2.1), and let

$$f_0(x, \theta_0) = \mathcal{N}(x; \mu_0, \sigma_0), f_1(x, \theta_1) = \mathcal{N}(x; \mu_1, \sigma_1). \tag{2.17}$$

Let $\mathfrak{C}^1 = \mathfrak{C}_{\mathrm{ML}}(x; 1)$ and $\mathfrak{C}^2 = \mathfrak{C}_{\mathrm{ML}}(x; 0.46)$ be the classification algorithms identified by the red and green points in Fig. 2.2(a), respectively. Notice that, when the observations are not manipulated and follow the distributions (2.17), $\mathfrak{C}^1$ achieves higher accuracy and sensitivity than $\mathfrak{C}^2$. This is also the case when using definition (2.16), as illustrated in Fig. 2.2(b). While the nominal distributions (2.17) are used to design the classifiers $\mathfrak{C}^1$ and $\mathfrak{C}^2$, we

Table 2.1: Numerical Results for Binary Classification

Classifier	y_1	y_2	$\mathcal{S}(y;\theta)$	$\mathcal{A}(y;\theta)$	$\mathcal{A}_{\text{adv1}}$	$\mathcal{A}_{\text{adv2}}$
$\mathfrak{C}^1$	3.65	18.78	0.0334	0.7891	0.6857	0.6808
$\mathfrak{C}^2$	1.83	20.60	0.0201	0.7766	0.6947	0.6939

consider an adversary that manipulates the observations so that their true distributions are

$$\begin{aligned} &f_0(x,\theta_0) = \mathcal{N}(x;\mu_0+\bar{\mu}_0,\sigma_0+\bar{\sigma}_0),\ \text{and} \\ &f_1(x,\theta_1) = \mathcal{N}(x;\mu_1+\bar{\mu}_1,\sigma_1+\bar{\sigma}_1), \end{aligned} \tag{2.18}$$

where $\bar{\mu}_0$, $\bar{\mu}_1$, $\bar{\sigma}_0$, and $\bar{\sigma}_1$ are unknown parameters selected by the adversary to deteriorate the accuracy of the classifiers.

To evaluate the accuracy of $\mathfrak{C}^1$ and $\mathfrak{C}^2$, we generate 10000 observations obeying the modified distributions (2.18), and compute the accuracy of the classifiers as the ratio of the number of correct predictions to the total number of observations. We repeat this experiment 100 times, and then compute the average accuracy of the classifiers over all trials.

Table 2.1 summarizes the results of the classification problems with $\mathfrak{C}^1$ and $\mathfrak{C}^2$ on the altered observations. In particular, y_1 and y_2 are the decision boundaries of the classifiers, while $\mathcal{S}(y;\theta)$ and $\mathcal{A}(y;\theta)$ denote their nominal sensitivity and accuracy. Instead, $\mathcal{A}_{\text{adv1}}$ and $\mathcal{A}_{\text{adv2}}$ denote the average accuracy of the classifiers when, respectively, the adversarial parameters are $\bar{\mu}_1 = \bar{\mu}_0 = \bar{\sigma}_0 = 0$, $\bar{\sigma}_1 = 3$, and $\bar{\mu}_0 = 1$, $\bar{\sigma}_0 = 2$, $\bar{\mu}_1 = -2$, $\bar{\sigma}_1 = 1.5$. The results show that, although $\mathfrak{C}^1$ exhibits higher accuracy than $\mathfrak{C}^2$ when the observations follow the nominal distributions (2.17), $\mathfrak{C}^2$ outperforms $\mathfrak{C}^1$ in both adversarial scenarios, as

Table 2.2: Numerical Results for Digit Classification

Neural Networks	NN1	NN2	NN3	NN4
$\mathcal{A}_{\text{nom}}$	0.9828	0.9641	0.9170	0.8665
$\mathcal{A}_{\text{adv}}$	0.2462	0.2734	0.3189	0.3204

supported by our analysis.

Next, we illustrate that the results of Theorem 3.1 can be observed for more complex and multidimensional classification problems. We consider the classification of hand-written digits (0-9) using a neural network (NN). We consider a NN with 6 layers, which uses cross entropy loss function, and we use the MNIST dataset [57] for its training. We add a regularization term to the loss function to increase the robustness of the NN against adversarial perturbations. We train 4 NNs using unperturbed images - NN1 without any regularization term, and NN2, NN3 and NN4 with increasing regularization weight coefficients. The adversarial images are computed using the framework of [90]. The results are reported in Table 2.2, where $\mathcal{A}_{\text{nom}}$ and $\mathcal{A}_{\text{adv}}$ denote the accuracy of a NN under clean and adversarial images, respectively. We observe that a NN with larger robustness ($\mathcal{A}_{\text{adv}}$) exhibits lower accuracy ($\mathcal{A}_{\text{nom}}$), indicating the existence of an accuracy-sensitivity tradeoff.

Chapter 3

Fundamental Performance - Robustness Tradeoff for Adversarial Classification with Abstaining

In this chapter, we take a different route than Chapter 2 for addressing adversarial robustness in classification problems. We consider an abstain option, where a classifier with fixed classification boundaries may abstain from giving an output over some region in the input space that the classifier is uncertain about. Mainly the inputs in such a region are the most prone to adversarial attacks. Thus, abstaining over such a region helps the classifier to avoid misclassifying perturbed inputs, and hence improve its adversarial robustness. In particular, under a perturbed input, instead of giving a wrong output (or possibly a correct

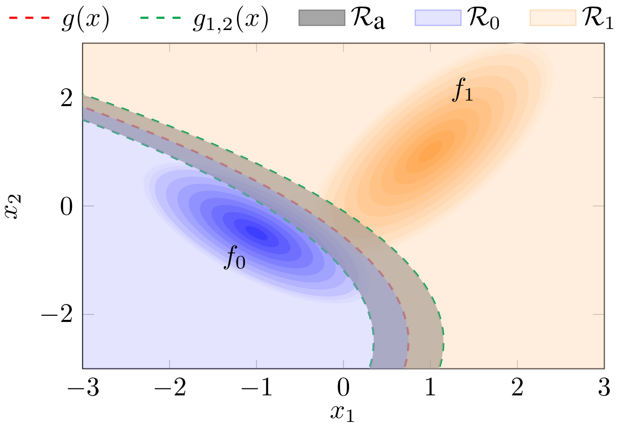


Figure 3.1: This figure shows the distribution of x under class $\mathcal{H}_0$ (blue ellipsoid) and class $\mathcal{H}_1$ (orange ellipsoid). $g(x)$, represented by the dashed red line, is the hyperspace decision boundary for the non-abstain case for the classifier in (3.2). It divides the observation space into $\mathcal{R}_0$ (blue region) and $\mathcal{R}_1$ (orange region). $g_1(x)$ and $g_2(x)$, represented by the green dashed lines, are the boundaries of the abstain region $\mathcal{R}_a$ (gray region).

output with low confidence), the model decides to abstain from giving one. For instance, if a self-driving car detects an object that it is uncertain about (it could be a shadow or maybe sensor measurements are perturbed by an adversary), it could abstain from giving an output that might lead to a car accident, and ask a human to take control. In safety critical applications, abstaining on low confidence output might be better than making a wrong decision. Motivated by this, we study the problem of classification with an abstain option by casting it into a binary hypothesis testing framework, where we add a third region in the observation space that corresponds to the observations on which the classifier abstains on (Fig. 3.1). Particularly, we study the relation between the accuracy and the adversarial robustness of a binary classifier upon varying the abstain region, where we show that improving the adversarial robustness of a classifier via abstaining comes at the expense of its accuracy. The results of this chapter are reported in our published paper [66].

3.1 Problem setup and preliminary notions

We consider a d-dimensional binary classification problem formulated as hypothesis testing problem as in [64]. The objective is to decide whether an observation $x \in \mathbb{R}^d$ belongs to class $\mathcal{H}_0$ or class $\mathcal{H}_1$. We assume that the distribution of the observations under class $\mathcal{H}_0$ and class $\mathcal{H}_1$ satisfy

$$\mathcal{H}_0 : x \sim f_0(x), \text{ and } \mathcal{H}_1 : x \sim f_1(x), \tag{3.1}$$

where $f_0(x)$ and $f_1(x)$ are known arbitrary probability density functions. For notational convenience, in the rest of this paper we denote $f_0(x)$ and $f_1(x)$ by f_0 and f_1, respectively. We denote the prior probabilities of the observations under f_0 and f_1 by p_0 and p_1, respectively. In this setup, any classifier can be represented by a partition of the $\mathbb{R}^d$ space by placing decision boundaries at suitable positions (see Fig. 3.1). We consider adversarial manipulations of the observations, where an attacker is capable of adding perturbations to the observations in order to degrade the performance of the classifier. We model[1] such manipulations as a change of the probability density functions in (3.1). We refer to the perturbed f_0 and f_1 in (3.1) as $\widetilde{f}_0$ and $\widetilde{f}_1$, respectively. In this work, we aim to improve the adversarial robustness of any classifier by abstaining from making a decision for low confidence outputs. A classifier with an abstain option can be written as

$$\mathfrak{C}(x; g(x), g_1(x), g_2(x)) = \begin{cases} \mathcal{H}_0, & x \in \mathcal{R}_0 \cap \overline{\mathcal{R}}_{\mathrm{a}}, \\ \mathcal{H}_1, & x \in \mathcal{R}_1 \cap \overline{\mathcal{R}}_{\mathrm{a}}, \\ \mathcal{H}_{\mathrm{a}}, & x \in \mathcal{R}_{\mathrm{a}}, \end{cases} \tag{3.2}$$

[1] In this work, we do not specify a model for the adversary, our analysis holds independently of the adversary model.

where $g(x)$[2] gives the hyperspace decision boundary for the non-abstain case, $g_1(x)$ and $g_2(x)$ give the hyperspace boundaries for the abstain region, specifically,

$$\begin{aligned}
\mathcal{R}_0 &= \{z \ : \ g(z) \leq 0, \forall z \in \mathbb{R}^d\}, \\
\mathcal{R}_1 &= \{z \ : \ g(z) > 0, \forall z \in \mathbb{R}^d\}, \qquad (3.3) \\
\mathcal{R}_\mathrm{a} &= \{z \ : \ (g_1(z) \geq 0) \cap (g_2(z) \leq 0), \forall z \in \mathbb{R}^d\},
\end{aligned}$$

and $\overline{\mathcal{R}}_\mathrm{a}$ is the complement set of $\mathcal{R}_\mathrm{a}$. We define two metrics to measure the performance and robustness of classifier (3.2).

Definition 11 ***(Nominal error)*** *The nominal error of a classifier with an abstain option is the proportion of the (unperturbed) observations that are misclassified or abstained on,*

$$\begin{aligned}
e_{nom}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_a) =& p_0\mathbf{P}\left[x \in \mathcal{R}_1 | \mathcal{H}_0\right] + p_1\mathbf{P}\left[x \in \mathcal{R}_0 | \mathcal{H}_1\right] \\
&+ p_0\mathbf{P}\left[x \in (\mathcal{R}_0 \cap \mathcal{R}_a) | \mathcal{H}_0\right] \\
&+ p_1\mathbf{P}\left[x \in (\mathcal{R}_1 \cap \mathcal{R}_a) | \mathcal{H}_1\right], \qquad (3.4)
\end{aligned}$$

where $\mathcal{R}_0$, $\mathcal{R}_1$, and $\mathcal{R}_a$ are as in (3.3). □

The first two terms in (3.4) correspond to the error without abstaining, therefore, they do not depend on the abstain region $\mathcal{R}_\mathrm{a}$. The last two terms correspond to the abstain error, thus, they depend on $\mathcal{R}_\mathrm{a}$. Using Definition 11 and the distributions in (3.1), the nominal error for classifier (3.2) is written as

$$\begin{aligned}
e_\mathrm{nom}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_\mathrm{a}) = & p_0 \int\limits_{\mathcal{R}_1} f_0 dx + p_1 \int\limits_{\mathcal{R}_0} f_1 dx \\
&+ p_0 \int\limits_{\mathcal{R}_0 \cap \mathcal{R}_\mathrm{a}} f_0 dx + p_1 \int\limits_{\mathcal{R}_1 \cap \mathcal{R}_\mathrm{a}} f_1 dx. \qquad (3.5)
\end{aligned}$$

[2] Technically, $g(x)$ is not the boundary, $g(x) = 0$ provides the boundary, but for the notational convenience we use $g(x)$ to refer to the boundary. Similarly, we use $g_1(x)$ and $g_2(x)$ instead of $g_1(x) = 0$ and $g_2(x) = 0$.

As can be seen in (3.5), the nominal classification error depends on $\mathcal{R}_0$, $\mathcal{R}_1$, and $\mathcal{R}_\mathrm{a}$, and thus on the position of the boundaries, $g(x)$, $g_1(x)$, and $g_2(x)$, as described in (3.3). Lower nominal error implies higher classification performance. Note that, if there is no abstain option ($\mathcal{R}_\mathrm{a} = \varnothing$), then the nominal error is equal to the error computed in the classic hypothesis testing framework [85].

Definition 12 ***(Adversarial error)*** *The adversarial error of a classifier with an abstain option is the proportion of the perturbed observations that are misclassified and not abstained on,*

$$\begin{aligned} e_{adv}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_a) = & p_0 \mathbf{P}\left[\widetilde{x} \in (\mathcal{R}_1 \cap \overline{\mathcal{R}}_a) | \mathcal{H}_0\right] \\ & + p_1 \mathbf{P}\left[\widetilde{x} \in (\mathcal{R}_0 \cap \overline{\mathcal{R}}_a) | \mathcal{H}_1\right], \end{aligned} \tag{3.6}$$

where $\widetilde{x} \in \mathbb{R}^d$ is a perturbed observation that follows distributions $\widetilde{f}_0$ and $\widetilde{f}_1$ under classes $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. □

Using Definition 12 and the distributions in (3.1), we can write the adversarial error for classifier (3.2) as

$$e_\mathrm{adv}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_\mathrm{a}) = p_0 \int\limits_{\mathcal{R}_1 \cap \overline{\mathcal{R}}_\mathrm{a}} \tilde{f}_0 dx + p_1 \int\limits_{\mathcal{R}_0 \cap \overline{\mathcal{R}}_\mathrm{a}} \tilde{f}_1 dx, \tag{3.7}$$

Similar to the nominal error, the adversarial error depends on $\mathcal{R}_0$, $\mathcal{R}_1$, and $\mathcal{R}_\mathrm{a}$ defined in (3.3). Further, the adversarial error depends on the perturbed distributions $\tilde{f}_0$ and $\tilde{f}_1$. The adversarial error is related to the classifier's robustness to adversarial attacks, where low adversarial error implies higher robustness. Note that, if a classifier abstains over the whole input space ($\mathcal{R}_\mathrm{a} = \mathbb{R}^d$), then the adversarial error converges to zero, and the classifier achieves maximum possible robustness. Yet, such classifier achieves maximum nominal error.

Remark 13 ***(Intuition behind Definition 11 and 12)*** *Abstaining from making a decision can be better than making a wrong one, yet worse than making a correct one. e_{nom} penalizes abstaining (along with misclassification) since the classifier is not performing the required task, which is to make a decision. On the other hand, e_{adv} does not penalize abstaining since by abstaining from making a decision on perturbed inputs, the classifier is avoiding an adversarial attack that can lead to misclassification. Each of these two definitions is a different performance metric, where e_{nom} measures the classifier's nominal performance, while e_{adv} measures the classifier's robustness against adversarial perturbations of the input. Further, these definitions guarantee that abstaining does not yield a unilateral advantage or disadvantage, where the classifier would abstain always or never. We remark that different definitions are possible.* □

3.2 Tradeoff between nominal and adversarial errors

Ideally, we would like both the nominal error and the adversarial error to be small. However, in this section we show that these errors cannot be minimized simultaneously.

Theorem 14 ***(Nominal-adversarial error tradeoff)*** *For classifier (3.2), let $\mathcal{R}_{a0} = \mathcal{R}_0 \cap \mathcal{R}_a$ and $\mathcal{R}_{a1} = \mathcal{R}_1 \cap \mathcal{R}_a$, and let $\widetilde{\mathcal{R}}_a \supset \mathcal{R}_a$ be another abstain region that is partitioned* *as* $\widetilde{\mathcal{R}}_a = \widetilde{\mathcal{R}}_{a0} \cup \widetilde{\mathcal{R}}_{a1}$*, with* $\widetilde{\mathcal{R}}_{a0} \supset \mathcal{R}_{a0}$ *and* $\widetilde{\mathcal{R}}_{a1} \supset \mathcal{R}_{a1}$*. Then,*

$$e_{nom}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_a) < e_{nom}(\mathcal{R}_0, \mathcal{R}_1, \widetilde{\mathcal{R}}_a), \tag{3.8}$$

$$e_{adv}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_a) > e_{adv}(\mathcal{R}_0, \mathcal{R}_1, \widetilde{\mathcal{R}}_a). \tag{3.9}$$

Proof. *For notational convenience, we denote* $e_{nom}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_a)$*,* $e_{adv}(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_a)$*,* $e_{nom}(\mathcal{R}_0, \mathcal{R}_1, \widetilde{\mathcal{R}}_a)$*, and* $e_{adv}(\mathcal{R}_0, \mathcal{R}_1, \widetilde{\mathcal{R}}_a)$ *by* e_{nom}*,* e_{adv}*,* $\widetilde{e}_{nom}$*, and* $\widetilde{e}_{adv}$*, respectively. For a classifier as in (3.2) with abstain region* $\widetilde{\mathcal{R}}_a$*, we can write*

$$\begin{aligned}\widetilde{e}_{nom} =& p_0\Big(\int\limits_{\mathcal{R}_1} f_0 dx + \int\limits_{\widetilde{\mathcal{R}}_{a0}} f_0 dx\Big) + p_1\Big(\int\limits_{\widetilde{\mathcal{R}}_{a1}} f_1 dx + \int\limits_{\mathcal{R}_0} f_1 dx\Big)\\ =& p_0 \int\limits_{\mathcal{R}_1} f_0 dx + p_0 \int\limits_{\mathcal{R}_{a0}} f_0 dx + p_0 \int\limits_{\widetilde{\mathcal{R}}_{a0} \setminus \mathcal{R}_{a0}} f_0 dx\\ &+ p_1 \int\limits_{\mathcal{R}_0} f_1 dx + p_1 \int\limits_{\mathcal{R}_{a1}} f_1 dx + p_1 \int\limits_{\widetilde{\mathcal{R}}_{a1} \setminus \mathcal{R}_{a1}} f_1 dx.\end{aligned}$$

Then, we can write

$$\widetilde{e}_{nom} - e_{nom} = p_0 \int\limits_{\widetilde{\mathcal{R}}_{a0} \setminus \mathcal{R}_{a0}} f_0 dx + p_1 \int\limits_{\widetilde{\mathcal{R}}_{a1} \setminus \mathcal{R}_{a1}} f_1 dx > 0.$$

Similarly, we can write

$$\widetilde{e}_{adv} - e_{adv} = -p_0 \int\limits_{\widetilde{\mathcal{R}}_{a0} \setminus \mathcal{R}_{a0}} \tilde{f}_0 dx - p_1 \int\limits_{\widetilde{\mathcal{R}}_{a1} \setminus \mathcal{R}_{a1}} \tilde{f}_1 dx < 0.$$

■

As we increase the abstain region from $\mathcal{R}_\text{a}$ to $\widetilde{\mathcal{R}}_\text{a}$, e_nom strictly increases, while e_adv strictly decreases, which indicates a tradeoff relation between both errors as we vary the abstain region. Theorem 14 implies that there exist a tradeoff between e_nom and e_adv. Therefore, the classifier's adversarial robustness can be improved only at the expense of its classification performance. In practice, the classifier's robustness can be improved by increasing $\mathcal{R}_{\mathrm{a}}$, while the nominal classification performance can be improved by decreasing $\mathcal{R}_{\mathrm{a}}$.

Remark 15 ***(Comparing our tradeoff with the literature)*** *[64, 95, 107] showed that a tradeoff relation exists between a classifier's nominal performance and its adversarial robustness. Despite using different frameworks, their performance-robustness tradeoff relation is obtained via tuning the classifier's boundaries in a way that improves its robustness. In our result, we fix the classifier's decision boundaries, and include an abstain region that can be tuned to obtain our performance-robustness tradeoff. It is possible that a classifier with an abstain option and a classifier without an abstain option but with different decision boundaries achieve the same e_{nom} and e_{adv}. Although both classifiers achieve same metrics, they are different, where the latter gives an output all the time, while the former abstains on some inputs.* □

Next we provide our analysis on how to select the abstain region for the 1-dimensional binary classification problem. Consider the same binary hypothesis testing problem introduced in Section 7.1, but with a scalar observation space where the observation $x \in \mathbb{R}$ is distributed under classes $\mathcal{H}_0$ and $\mathcal{H}_1$ as in (3.1). In this setup, any classifier can be represented by a partition of the real line by placing decision boundaries at suitable positions (see Fig. 3.2). Let[3] $-\infty = y_0 \leq \cdots \leq y_{n+1} = \infty$ denote n decision boundaries with $y = [y_i]$. Then, the

[3] For simplicity and without loss of generality, we assume that n is even. Further, an alternative configuration of the classifier (3.2) assigns $\mathcal{H}_0$ and $\mathcal{H}_1$ to $\mathcal{R}_1$ and $\mathcal{R}_0$, respectively. However, we consider only the configuration in (3.2) without affecting the generality of our analysis.

classifier regions are

$$\mathcal{R}_0 = \{z \ : \ y_i < z < y_{i+1}, \text{ for } i = 0, 2, \ldots, n\},$$

$$\mathcal{R}_1 = \{z \ : \ y_i \leq z \leq y_{i+1}, \text{ for } i = 1, 3, \ldots, n-1\},$$

$$\mathcal{R}_\text{a} = \{z \ : \ y_i - \gamma_{i1} \leq z \leq y_i + \gamma_{i2}, \text{ for } i = 1, 2, \ldots, n\},$$

where $\gamma_{ij} \in \mathbb{R}_{\geq 0}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2$. Let $\gamma = [\gamma_{11}, \gamma_{12}, \ldots, \gamma_{i1}, \gamma_{i2}, \ldots, \gamma_{n1}, \gamma_{n2}]^\mathsf{T}$, $y_{i1} = y_i - \gamma_{i1}$, and $y_{i2} = y_i + \gamma_{i2}$. Using (3.4) and (3.1), we have

$$\begin{aligned} e_\text{nom}(y, \gamma) =& p_0 \left(\sum_{l=1}^{n} (-1)^l \int\limits_{-\infty}^{y_{lj}} f_0 dx \right) \\ &+ p_1 \left(\sum_{l=1}^{n} (-1)^{l+1} \int\limits_{-\infty}^{y_{lk}} f_1 dx + 1 \right). \end{aligned} \tag{3.10}$$

where $j = \frac{(-1)^l+1}{2} + 1$ and $k = \frac{(-1)^{l+1}+1}{2} + 1$ for $l = 1, \ldots, n$. Using (3.6), the adversarial error becomes

$$\begin{aligned} e_\text{adv}(y, \gamma) =& p_0 \left(\sum_{l=1}^{n} (-1)^l \int\limits_{-\infty}^{y_{lk}} \tilde{f}_0 dx \right) \\ &+ p_1 \left(\sum_{l=1}^{n} (-1)^{l+1} \int\limits_{-\infty}^{y_{lj}} \tilde{f}_1 dx + 1 \right), \end{aligned} \tag{3.11}$$

where j and k are the same as above. Given a classifier as in (3.2) with known boundaries y, we are interested in how to select the abstain region, i.e., how to choose γ given y. To this aim, we cast the following optimization problem:

$$\begin{aligned} e^*_\text{adv}(\zeta) = \min_{\gamma} \quad & e_\text{adv}(y, \gamma) \\ \text{s.t.} \quad & e_\text{nom}(y, \gamma) \leq \zeta, \end{aligned} \tag{3.12}$$

where $\zeta \in [e_{\text{nom}}(y,0),1]$. In what follows, we characterize the solution γ^* to (5.3). We begin by writing the derivative of the errors in (3.10) and (3.11) with respect to γ:

$$\begin{aligned}\frac{\partial e_{\text{nom}}}{\partial \gamma_{i1}} &= p_q f_q(y_i - \gamma_{i1}), \quad \frac{\partial e_{\text{nom}}}{\partial \gamma_{i2}} = p_r f_r(y_i + \gamma_{i2}),\\ \frac{\partial e_{\text{adv}}}{\partial \gamma_{i1}} &= -p_r \tilde{f}_r(y_i - \gamma_{i1}), \quad \frac{\partial e_{\text{adv}}}{\partial \gamma_{i2}} = -p_q \tilde{f}_q(y_i + \gamma_{i2}),\end{aligned} \tag{3.13}$$

where $q = \frac{(-1)^i+1}{2}$ and $r = \frac{(-1)^{i+1}+1}{2}$ for $i = 1, \ldots n$. Note that the derivative of e_{nom} with respect to γ is strictly positive, while that of e_{adv} is strictly negative. Thus, e_{nom} increases while e_{adv} decreases as γ increases (i.e., as $\mathcal{R}_{\text{a}}$ increases), which agrees with the result of Theorem 14. Problem (5.3) is not convex and it might not exhibit a unique solution. The following theorem characterizes a solution γ^* to (5.3).

Theorem 16 ***(Characterizing the solution to the minimization problem*** (5.3)***)*** *Given classifier* (3.2) *with* 1*-dimensional input and known n boundaries y, the solution γ^* to problem* (5.3) *satisfies the following necessary conditions*

$$e_{nom}(y,\gamma) = \zeta, \tag{3.14}$$

$$\frac{\partial e_{adv}(y,\gamma)}{\partial \gamma_{iu}} . \frac{\partial e_{nom}(y,\gamma)}{\partial \gamma_{jv}} = \frac{\partial e_{adv}(y,\gamma)}{\partial \gamma_{jv}} . \frac{\partial e_{nom}(y,\gamma)}{\partial \gamma_{iu}}, \tag{3.15}$$

for $i, j = 1, \ldots, n$, $i \neq j$, and $u, v = 1, 2$, where the derivatives of e_{nom} and e_{adv} with respect to γ are as in (3.13).

Proof. *Defining the Lagrange function of* (5.3)

$$\mathcal{L}(\gamma, \lambda) = e_{adv}(y,\gamma) + \lambda(e_{nom}(y,\gamma) - \zeta), \tag{3.16}$$

where λ is the Karush-Kuhn-Tucker (KKT) multiplier. For notational convenience, we denote $e_{adv}(y,\gamma)$ and $e_{nom}(y,\gamma)$ by e_{adv} and e_{nom}, respectively. The stationarity KKT con-

dition implies $\frac{\partial}{\partial\gamma}\mathcal{L}(\gamma,\lambda)=0$*, which is written as*

$$\frac{\partial e_{adv}}{\partial\gamma}=-\lambda\frac{\partial e_{nom}}{\partial\gamma}. \tag{3.17}$$

Using (3.17) *we write*

$$-\lambda=\frac{\partial e_{adv}}{\partial\gamma_{iu}}\Big/\frac{\partial e_{nom}}{\partial\gamma_{iu}}=\frac{\partial e_{adv}}{\partial\gamma_{jv}}\Big/\frac{\partial e_{nom}}{\partial\gamma_{jv}}, \tag{3.18}$$

for $i,j=1,\dots,n$*,* $i\neq j$*, and* $u,v=1,2$*, which gives us* (3.15)*. The KKT condition for dual feasibility implies that* $\lambda\geq 0$*. However, since we have* $\frac{\partial e_{adv}}{\partial\gamma}\neq 0$ *and* $\frac{\partial e_{nom}}{\partial\gamma}\neq 0$ *from* (3.13)*, we get from* (3.17) *that* $\lambda>0$*. Further, the KKT condition for complementary slackness implies* $\lambda(e_{nom}-\zeta)=0$*. Since* $\lambda>0$*, then* $e_{nom}-\zeta=0$*, which gives us* (3.14)*.* ■

Remark 17 ***(Location of the abstain region in the observation space)*** *The abstain region in Theorem 14 can be located anywhere in the observation space. However, in Theorem 16, we assume that the abstain region is located around the decision boundaries. This assumption is fair since the observations near the classifier's boundaries tend to have low classification confidence and are prone to misclassification.* □

We conclude this section with an illustrative example.

Example 18 ***(Classifier with an abstain option for exponential distributions)*** *Consider a 1-D binary hypothesis testing problem, where the observation* $x\in\mathbb{R}$ *under classes* $\mathcal{H}_0$ *and* $\mathcal{H}_1$ *follows exponential distributions, i.e., the probability density functions in* (3.1) *have the form* $f_i(x)=\rho_i\exp(-\rho_i x)$ *over the domain* $x\in\mathbb{R}_{\geq 0}$ *with parameter* $\rho_i>0$ *for* $i=0,1$*. We consider a single boundary classifier with an abstain option as in* (3.2)*, with boundary* y_1 *and abstain parameters* γ_{11} *and* γ_{12} *(see Fig. 3.2). For simplicity, we model*

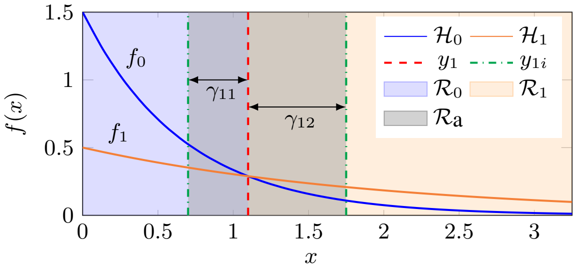


Figure 3.2: This figure shows the binary classification problem described in Example 18, where the observation x under hypotheses $\mathcal{H}_0$ (solid blue line) and $\mathcal{H}_1$ (solid orange line) follows exponential distribution with $\rho_0 = 1.5$ and $\rho_1 = 0.5$, respectively. The dashed red line is the decision boundary for the non-abstain case, which divides the space into $\mathcal{R}_0$ (blue region) and $\mathcal{R}_1$ (orange region). The dot-dashed green lines are the boundaries of the abstain region $\mathcal{R}_\text{a}$ (gray region), which is parametrized by γ_{11} and γ_{12}.

the adversarial manipulations of the observations as perturbation added to the distributions' parameters. We refer to the perturbed parameters as $\tilde{\rho}_0$ and $\tilde{\rho}_1$. Using Theorem 16:

$$\begin{aligned}
&p_0 \exp(-\rho_0(y_1 - \gamma_{11})) - p_1 \exp(-\rho_1(y_1 + \gamma_{12})) + p_1 = \zeta, \\
&p_1^2 \tilde{\rho}_1 \rho_1 \exp(-\tilde{\rho}_1(y_1 - \gamma_{11}) - \rho_1(y_1 + \gamma_{12})) \\
&\qquad = p_0^2 \tilde{\rho}_0 \rho_0 \exp(-\tilde{\rho}_0(y_1 + \gamma_{12}) - \rho_0(y_1 - \gamma_{11})).
\end{aligned} \tag{3.19}$$

For a given classifier with known boundary, y_1, and with desired nominal performance, ζ, along with the knowledge of the perturbed distribution parameters $\tilde{\rho}_0$ and $\tilde{\rho}_1$, we can choose the optimal abstain region by solving (3.19) for γ_{11} and γ_{12}. A solution of (3.19) corresponds to a local minima of (5.3). Note that the constraint (5.3) is active (see Theorem 16), hence we have $e_{nom}(y_1, \gamma^*) = \zeta$. Fig. 3.3 shows the values of e^*_{adv} obtained by solving (3.19) for γ^*_{11} and γ^*_{12} over the range $\zeta \in [e_{nom}(y_1, 0), 1]$ with $\rho_0 = 1.5$, $\rho_1 = 0.5$, $\tilde{\rho}_0 = 1.2$, $\tilde{\rho}_1 = 0.7$, and $p_0 = p_1 = 0.5$. Moreover, Fig. 3.3 shows the values of e_{adv} as a function of e_{nom} as γ_{11} and γ_{12} are varied arbitrarily. Both curves show a tradeoff between e_{nom} and e_{adv} as predicted by Theorem 14. Further, at each value of $e_{nom} \in (e_{nom}(y_1, 0), 1)$, we observe that

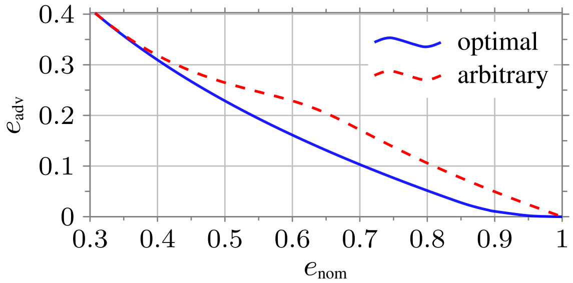


Figure 3.3: This figure shows the tradeoff between e_{nom} and e_{adv} as we vary the abstain region, γ, for the classifier described in Example 18. The solid blue line is obtained using Theorem 16 to solve for γ^* for each value of e_{nom}, while the dashed red line is obtained by varying γ arbitrarily. Both curves coincide at the extreme points at $e_{\text{nom}} = 0.31$ and $e_{\text{nom}} = 1$, which correspond to $\mathcal{R}_{\text{a}} = \varnothing$ (no abstaining) and $\mathcal{R}_{\text{a}} = \mathbb{R}$ (always abstaining), respectively. For $e_{\text{nom}} \in (0.31, 1)$, we observe that the optimal curve achieves lower e_{adv} than the curve obtained by arbitrary selection of γ.

$e^*_{adv}(\zeta) < e_{adv}(y_1, \gamma)$. □

3.3 Numerical experiment using MNIST dataset

In this section, we illustrate the implications of Theorem 14 using the classification of hand-written digits from the MNIST dataset [57]. First, we design and train a classifier with an abstain option. Then, we use Definition 11 and 12 to compute e_{nom} and e_{adv} for a classifier given the dataset. Finally, we present our numerical results on the MNIST dataset. Although our theoretical results are for binary classification, we show that a tradeoff between e_{nom} and e_{adv} exists for multi-class classification using the MNIST dataset.

3.3.1 Classifier design and training

We design a classifier $h : \mathbb{X} \to \mathbb{Y}$ using the Lipschitz-constrained loss minimization scheme introduced in [52][4]:

$$\begin{aligned} \min_{h \in \mathrm{Lip}(\mathbb{X};\mathbb{Y})} \quad & \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} L\left(h\left(x_i\right), y_i\right), \\ \text{s.t.} \quad & \mathrm{lip}(h) \le \alpha, \end{aligned} \tag{3.20}$$

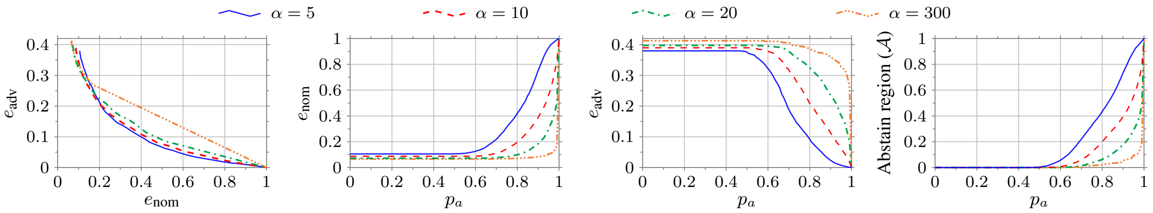
where $\mathbb{X} \subset \mathbb{R}^d$ and $\mathbb{Y} \subset \mathbb{R}^m$ are the respective input and output space, $\mathrm{Lip}(\mathbb{X};\mathbb{Y})$ denotes the space of the Lipschitz continuous maps from $\mathbb{X}$ to $\mathbb{Y}$, L is the loss function of the learning problem, the pair $\{x_i, y_i\}_{i=1}^{N_{\text{train}}}$ denotes the training dataset of size N_{train}, with input $x \in \mathbb{X}$ and output y[5] $\in \mathbb{Y}$, $\mathrm{lip}(h)$ is the Lipschitz constant of classifier h, and $\alpha \in \mathbb{R}_{\ge 0}$ is the upper bound constraint on the Lipschitz constant. The classifier takes an input image of d pixels and outputs a vector of probabilities of size m, which is the number of classes. The classifier chooses the class with the highest probability: higher probability implies higher decision confidence. We incorporate an abstain option, where the classifier abstains if the maximum probability is less than a threshold probability p_a. We consider adversarial examples, $\widetilde{x} = x + \delta$, computed as in [52], where $\delta \in \mathbb{R}^d$ is a bounded perturbation ($\|\delta\|_\infty \le \xi$) in the direction that induces misclassification.

3.3.2 Nominal and Adversarial error

Let $\mathcal{Z} = \{0, 1, \ldots, m-1\}$ and $\widehat{\mathcal{Z}} = \{0, 1, \ldots, m-1, a\}$ be the sets containing all possible true labels and all possible predicted labels by classifier h, respecticvely, where a

[4] Other classification algorithms, e.g. neural networks, can also be used.

[5] Label $y_i \in \mathbb{R}^m$ is a vector which contains 1 in the element that correspond to the true class and zero everywhere else.



(a) (b) (c) (d)

Figure 3.4: In the classification problem discussed in Section 3.3, 4 classifiers are trained on the MNIST dataset using the Lipschitz-constrained loss minimization scheme in (3.20), with $\alpha = 5, 10, 20, 300$, which are represented in all 4 panels by the solid blue line, the dashed red line, the dot-dashed green line, and the three-dot-dashed orange line, respectively. Panel (a) shows the tradeoff between e_{nom} and e_{adv}, panels (b) and (c) show e_{nom} and e_{adv} as a function of the threshold probability, p_a, respectively, and panel (d) shows the ratio of the abstain region to the input space, denoted by $\mathcal{A}$, as a function of p_a. As observed in (d), the abstain region is zero for $p_a \in [0, 0.5)$, it monotonically increases for $p_a \geq 0.5$ till it covers the whole input space when $p_a = 1$. When there is no abstaining (i.e., $p_a \in [0, 0.5)$), all classifiers achieve their lowest e_{nom} and their highest e_{adv} as observed in (b) and (c), respectively, where the classifier with $\alpha = 300$ achieves the lowest e_{nom} and the highest e_{adv} among all 4 classifiers, while the classifier with $\alpha = 5$ achieves the highest e_{nom} and the lowest e_{adv}, which agrees with the tradeoff result in [52]. When the abstain region covers the whole input space (i.e., $p_a = 1$), all classifiers achieve $e_{\text{nom}} = 1$ and $e_{\text{adv}} = 0$ as seen in (b) and (c), respectively. Also, it is observed in (b) and (c), respectively, that as the abstain region increases (i.e., p_a increases), e_{nom} increases while e_{adv} decreases for all classifiers, which leads to the tradeoff relation between the two as observed in (a).

corresponds to the abstain option. Let $z_i \in \mathcal{Z}$ and $\widehat{z}_i \in \widehat{\mathcal{Z}}$ be the true label and the label predicted by h for the input x_i, respectively $\big($i.e., $\widehat{z}_i$ is the label that corresponds to the maximum probability in the vector $h(x_i)$, or label a if the maximum probability is less than $p_a\big)$. Further, let $\widetilde{z}_i \in \widehat{\mathcal{Z}}$ be the label predicted by h for the perturbed input image $\widetilde{x}_i$. Using Definition 11 and 12 we compute e_{nom} and e_{adv} for h with threshold probability p_a on the testing dataset of size N_{test} as,

$$\begin{aligned} e_{\text{nom}}(h, p_a) &= \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{1}\{\widehat{z}_i \neq z_i\}, \\ e_{\text{adv}}(h, p_a) &= \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{1}\{\widetilde{z}_i \neq z_i \cap \widetilde{z}_i \neq a\}, \end{aligned} \tag{3.21}$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function.

3.3.3 Nominal-Adversarial error tradeoff

To show the implications of Theorem 14, we train four classifiers on the MNIST dataset using (3.20) with $\alpha = 5, 10, 20$, and 300, respectively (refer to [52] for details about the training scheme). Then, we compute e_{nom} and e_{adv} for each classifier using (3.21) with different values of p_a and a bound on the perturbation $\xi = 0.3$. Fig. 3.4 shows the numerical results on the testing dataset. Fig. 3.4(a) shows the tradeoff between e_{nom} and e_{adv} for all the classifiers, which agrees with Theorem 14. Fig. 3.4(b)-(c) show e_{nom} and e_{adv} as a function of p_a, respectively, while Fig. 3.4(d) shows the ratio of the abstain region to the input space, denoted by $\mathcal{A}$, as a function of p_a. As shown in Fig. 3.4(d), $\mathcal{A}$ increases at a low rate from zero to 0.1 for $p_a \in [0.5, 0.95]$ for the classifier with $\alpha = 300$, then it increases at a high rate till it reaches 1 for $p_a \in (0.95, 1]$. The rate at which $\mathcal{A}$ increases becomes more uniform as α decreases, where for the classifier with $\alpha = 5$, $\mathcal{A}$ increases with an almost uniform rate from zero at $p_a = 0.5$ to 1 at $p_a = 1$. This is because as we decrease α in (3.20), the learned function becomes more smooth, and the change of the output probability vector over the input space becomes smoother. As observed in Fig. 3.4(b) and Fig. 3.4(c), e_{nom} increases, while e_{adv} decreases for $p_a \in [0.5, 1]$.

Chapter 4

Fundamental Tradeoff between Performance and Robustness in Perception-Based Control

In this chapter, we characterize a fundamental tradeoff between accuracy and robustness in a data-driven control problem. We consider a perception-based control scenario, Fig. 4.1, where a camera is used to partially measure the state of a dynamical system and construct an estimator of the full state. We assume that the output map between the high-dimensional camera stream and the system state has been learned accurately [24], although the estimated statistics of the measurement noise are inaccurate. Such inaccuracies, which can arise from limited training data, sudden changes in environmental conditions, and adversarial manipulation, are unknown to the estimator and induce incorrect confidence bounds on the estimated state variables. In turn, inaccurate confidence bounds can lead to harmful control decisions [60]. Further, we show that, because of the incorrect noise statistics, accuracy of the estimation algorithm can be improved only at the expenses of its robustness. Thus, estimation algorithms that are optimal in the nominal training phase may underperform in practice compared to suboptimal algorithms. Our analytical results provide an explanation as to why nominally suboptimal data-driven algorithms can exhibit better generalization and robust properties in practice [46]. The results of this chapter are reported in our published paper [65].

4.1 Problem setup and preliminary notions

Consider the discrete-time, linear, time-invariant system

$$x(t+1) = Ax(t) + w(t), \tag{4.1}$$

$$y(t) = Cx(t) + v(t), \qquad t \geq 0, \tag{4.2}$$

where $x(t) \in \mathbb{R}^n$ denotes the state, $y(t) \in \mathbb{R}^m$ the output, $w(t)$ the process noise, and $v(t)$ the measurement noise. We assume that $w(t) \sim \mathcal{N}(0, Q)$, with $Q \geq 0$, $v(t) \sim \mathcal{N}(0, R)$, with $R > 0$, and $x(0) \sim \mathcal{N}(0, \Sigma_0)$, with $\Sigma_0 \geq 0$, are independent of each other at all times $t \geq 0$.[1] Finally, we assume that A is stable, that is, $\rho(A) < 1$. Note that this implies that (A, C) is detectable and $(A, Q^{\frac{1}{2}})$ is stabilizable.

We use a linear filter with constant gain $K \in \mathbb{R}^{n \times m}$ to estimate the state of the system (5.1) from the measurements (4.2):

$$\hat{x}(t+1) = A\hat{x}(t) + K[y(t+1) - CA\hat{x}(t)] \quad t \geq 0, \tag{4.3}$$

[1] See Section 4.3 for numerical examples showing that our main results seem to be valid also when some of these assumptions are not satisfied.

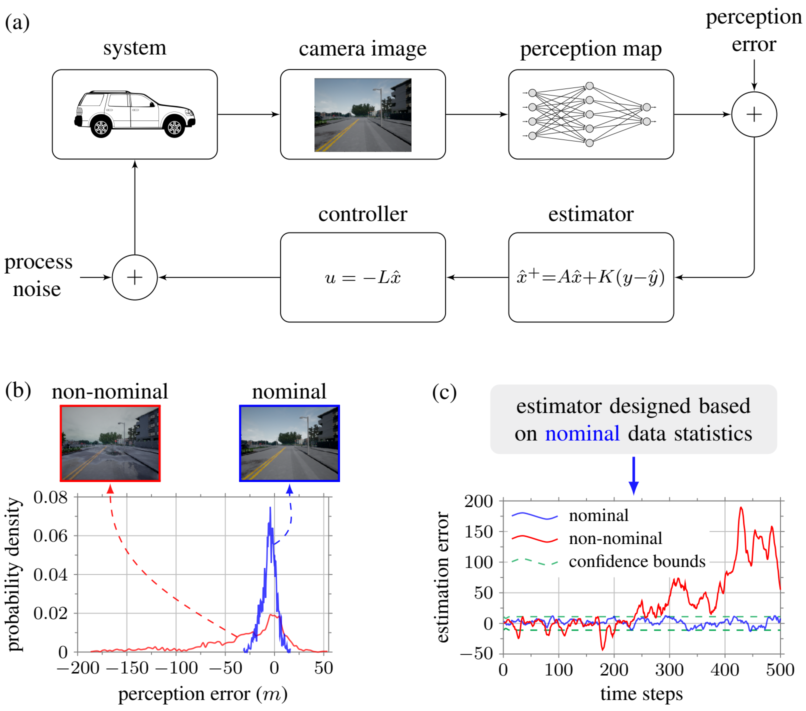


Figure 4.1: Panel (a) shows a perception-based control scenario, where the partial state of a dynamical system (vehicle) is extracted from the measurements of a high-dimensional sensor (camera) and used to implement a feedback control algorithm. A perception map is learned from a set of training data of finite size, which relates the sensor's readings to the system's state. Panel (b) shows the probability density functions of the perception error when operating in nominal (clear weather, as represented by the training data) and non-nominal (rainy weather, as it may occur in practice) conditions (error statistics are computed numerically using the simulator CARLA [27]). Due to inaccuracies and uncertainties in the sensed data, the error statistics of the perception map differ from the statistics learned during the training phase. As shown in panel (c), discrepancies in the error statistics lead to poor estimation performance in practical conditions. As we prove in this paper, a fundamental tradeoff exists between accuracy and robustness of a linear estimator (consequently, in the considered perception-based control setting), so that estimators that perform well on the training data may exhibit poor performance with non-nominal conditions, while robust estimators may exhibit mediocre yet robust performance in a broad set of conditions.

where $\hat{x}(t)$ denotes the state estimate at time t. Let $e(t) = x(t) - \hat{x}(t)$ and $P(t) = \mathbb{E}[e(t)e(t)^\mathsf{T}]$

denote the estimation error and its covariance, respectively. For $t \geq 0$, we have

$$e(t+1) = A_K e(t) + B_K w(t) - K v(t+1), \tag{4.4}$$

$$P(t+1) = A_K P(t) A_K^{\mathsf{T}} + B_K Q B_K^{\mathsf{T}} + K R K^{\mathsf{T}}, \tag{4.5}$$

where $A_K \triangleq A - KCA$ and $B_K \triangleq I_n - KC$. We assume that the gain K is chosen such that A_K is stable, that is, $\rho(A_K) < 1$. Under this assumption, $\lim_{t\to\infty} P(t) \triangleq P(K) \geq 0$ exists, and satisfies the Lyapunov equation

$$P(K) = A_K P(K) A_K^{\mathsf{T}} + B_K Q B_K^{\mathsf{T}} + K R K^{\mathsf{T}}. \tag{4.6}$$

The performance of the filter is quantified by $\mathcal{P}(K) \triangleq \mathsf{tr}(P(K))$, where a lower value of $\mathcal{P}(K)$ is desirable. Note that the steady-state gain K_{kf} of the Kalman filter [49] minimizes $\mathcal{P}(K)$ and depends on the matrices A, C, Q, R.

We allow for perturbations to the covariance matrix R, which may result from (i) modeling and estimation errors, as in the case of perception-based control, or (ii) accidental or adversarial tampering of the sensor, as in the case of false data injection attacks [77]. To quantify the effect of such perturbations to the covariance matrix R on the performance of the estimator, we define the following sensitivity metric:

$$\mathcal{S}(K) \triangleq \mathsf{tr}\left[\frac{d}{dR}\mathcal{P}(K)\right]. \tag{4.7}$$

Intuitively, if $\mathcal{S}(K)$ is large, then a small change in R can result in a large change (possibly, large increment) in $\mathcal{P}(K)$.

Remark 19 ***(Comparison with adversarial robustness)*** *In adversarial settings, the adversary designs a small deterministic perturbation added to a given observation (e.g.,* *pixels of an image) to deteriorate the performance of a machine learning algorithm. This perturbed observation can be viewed as a realization of a multi-dimensional distribution. Instead, in this work we consider perturbations to the sensor's noise covariance, which accounts for all possible realizations. Thus, our sensitivity metric captures the average performance change over all possible perturbations, rather than the degradation caused by a single worst-case perturbation.* □

Lower values of sensitivity $\mathcal{S}(K)$ are desirable, and indicate that the filter (4.3) is more robust to perturbations. This motivates the following optimization problem:

$$\begin{aligned} \mathcal{S}^*(\delta) = \min_{K} \quad & \mathcal{S}(K) \\ \text{s.t.} \quad & \mathcal{P}(K) \leq \delta, \end{aligned} \tag{4.8}$$

where $\delta \geq \mathcal{P}(K_{\text{kf}})$ for feasibility. In what follows, we characterize the solution K^* to (5.3), and the relations between the sensitivity $\mathcal{S}(K^*)$ and the error $\mathcal{P}(K^*)$ as δ varies. To facilitate the discussion, in the remainder of the paper we use *accuracy* to refer to any decreasing function of the error $\mathcal{P}(K)$ obtained by the gain K, and *robustness* to denote any decreasing function of the sensitivity $\mathcal{S}(K)$ of the gain K.

4.2 Accuracy vs robustness tradeoff in linear estimation algorithms

We begin by characterizing the sensitivity $\mathcal{S}(K)$.

Lemma 20 ***(Characterization of sensitivity)*** *Let the sensitivity $\mathcal{S}(K)$ be as in (4.7). Then, $\mathcal{S}(K) = \mathsf{tr}(S(K))$, where $S(K) \geq 0$ satisfies the following Lyapunov equation:*

$$S(K) = A_K S(K) A_K^\mathsf{T} + KK^\mathsf{T}. \tag{4.9}$$

Lemma 20 allows us to compute the sensitivity of the linear estimator (4.3) as a function of its gain. Before proving Lemma 20, we present the following technical result.

Lemma 21 ***(Property of the solution to Lyapunov equation)*** *Let A, B, Q be matrices of appropriate dimension with $\rho(A) < 1$. Let Y satisfy $Y = AYA^\mathsf{T} + Q$. Then, $\mathsf{tr}(BY) = \mathsf{tr}(Q^\mathsf{T} M)$, where M satisfies $M = A^\mathsf{T} M A + B^\mathsf{T}$.*

Proof. Since $\rho(A) < 1$, Y and M can be written as

$$Y = \sum_{i=0}^{\infty} A^i Q (A^\mathsf{T})^i \text{ and } M = \sum_{i=0}^{\infty} A^i B (A^\mathsf{T})^i. \tag{4.10}$$

The result follows by pre-multiplying Y and M by B and Q^T respectively, and using the cyclic property of trace. ■

Proof of Lemma 20: Taking the differential of (4.6) with respect to the variable R, we get

$$\begin{aligned} dP(K) &= A_K dP(K) A_K^\mathsf{T} + K dR K^\mathsf{T} \\ \Rightarrow d\mathsf{tr}(P(K)) &= \mathsf{tr}(dP(K)) \overset{(a)}{=} \mathsf{tr}(K dR K^\mathsf{T} M), \end{aligned} \tag{4.11}$$

where $M > 0$ satisfies: $M = A_K^\mathsf{T} M A_K + I_n$, and (a) follows from Lemma 21. From (4.11), we get

$$d\mathcal{P}(K) = \mathsf{tr}(K^\mathsf{T} M K dR) \Rightarrow \frac{d}{dR}\mathcal{P}(K) = K^\mathsf{T} M K. \tag{4.12}$$

Using (4.12) and (4.7), we have that $\mathcal{S}(K) = \mathsf{tr}(K^\mathsf{T} M K) = \mathsf{tr}(KK^\mathsf{T} M) = \mathsf{tr}(S(K))$, where $S(K)$ is defined in (4.9) and the last equality follows from Lemma 21. To conclude, the property $S(K) \geq 0$ follows by inspection from (4.9). ■

Notice that, since $S(K) \geq 0$, $\mathcal{S}(K) = \mathsf{tr}(S(K))$ is a valid norm of $S(K)$ and captures the size of $S(K)$. Further, $S(K) = 0$ for $K = 0$, that is, $K = 0$ achieves the lowest possible value of sensitivity. This implies that δ in the optimization problem (5.3) can be restricted to $[\mathcal{P}(K_{\text{kf}}), \mathcal{P}(0)]$ to characterize the accuracy-robustness tradeoff.

Next, we characterize the optimal solution to (5.3). We will show that, despite not being convex, the minimization problem (5.3) exhibits a unique local minimum. This implies that the local minimum is also the global minimum.

Theorem 22 ***(Solution to the minimization problem*** **(5.3)*)*** *Let $\delta \in [\mathcal{P}(K_{kf}), \mathcal{P}(0)]$ and $\lambda \geq 0$. Let $X \geq 0$ be the unique solution to the following Riccati equation:*

$$X = AXA^\mathsf{T} - AXC^\mathsf{T}(CXC^\mathsf{T} + I_m + \lambda R)^{-1} CXA^\mathsf{T} + \lambda Q. \tag{4.13}$$

Then, the global minimum of problem (5.3) is given by

$$K^*(\lambda) = XC^\mathsf{T}\Big(CXC^\mathsf{T} + I_m + \lambda R\Big)^{-1}, \tag{4.14}$$

where λ is selected such that $\mathcal{P}(K^*(\lambda)) \triangleq \mathcal{P}^*(\lambda) = \delta$.

Proof. *First-order necessary conditions:* We begin by computing the derivatives of $\mathcal{P}(K)$ and $\mathcal{S}(K)$ with respect to the variable K. For notational convenience, we denote $A_K, B_K, P(K)$ and $S(K)$ by $\bar{A}, B, P$ and S, respectively. Taking the differential of (4.9), we get

$$dS = \bar{A}dS\bar{A}^{\mathsf{T}} - dKCAS\bar{A}^{\mathsf{T}} - \bar{A}S(dKCA)^{\mathsf{T}} + dKK^{\mathsf{T}} + KdK^{\mathsf{T}} \triangleq \bar{A}dS\bar{A}^{\mathsf{T}} + Z \tag{4.15}$$

$$\Rightarrow d\mathcal{S}(K) \overset{(a)}{=} \mathsf{tr}(dS) \overset{(b)}{=} \mathsf{tr}(Z^{\mathsf{T}}M)$$

$$= 2\mathsf{tr}[(-CAS\bar{A}^{\mathsf{T}} + K^{\mathsf{T}})MdK]$$

$$\Rightarrow \frac{d}{dK}\mathcal{S}(K) = 2M(K - \bar{A}SA^{\mathsf{T}}C^{\mathsf{T}}), \tag{4.16}$$

where $M > 0$ satisfies $M = A_K^{\mathsf{T}}MA_K + I_n$, and (a) and (b) follow from Lemmas 20 and 21, respectively. A similar analysis of (4.6) yields

$$\frac{d}{dK}\mathcal{P}(K) = 2M(KR - \bar{A}PA^{\mathsf{T}}C^{\mathsf{T}} - BQC^{\mathsf{T}}). \tag{4.17}$$

Define the Lagrange function of problem (5.3) as

$$\mathcal{L}(K, \lambda) = \mathcal{S}(K) + \lambda\Big(\mathcal{P}(K) - \delta\Big), \tag{4.18}$$

where λ is the Karush-Kuhn-Tucker (KKT) multiplier. The stationary KKT condition implies $\frac{d}{dK}\mathcal{L}(K, \lambda) = 0$, which using (4.16) and (4.17) becomes

$$2M[K - \bar{A}SA^{\mathsf{T}}C^{\mathsf{T}} + \lambda(KR - \bar{A}PA^{\mathsf{T}}C^{\mathsf{T}} - BQC^{\mathsf{T}})] = 0. \tag{4.19}$$

Substituting $\bar{A} = A - KCA$ in the above equation, defining $X \triangleq A(S + \lambda P)A^{\mathsf{T}} + \lambda Q$, and using $M > 0$, we obtain (4.14). Next, we show that X satisfies (4.13). From (4.6) and (4.9):

$$S + \lambda P = \bar{A}(S + \lambda P)\bar{A}^{\mathsf{T}} + \lambda BQB^{\mathsf{T}} + K(I_m + \lambda R)K^{\mathsf{T}}$$

$$\Rightarrow X = A(S + \lambda P)A^{\mathsf{T}} + \lambda Q$$

$$= A\left[\bar{A}(S + \lambda P)\bar{A}^{\mathsf{T}} + \lambda BQB^{\mathsf{T}} + K(I_m + \lambda R)K^{\mathsf{T}}\right]A^{\mathsf{T}}$$

$$+ \lambda Q.$$

Using $\bar{A} = A - KCA$ and substituting the gain K in (4.14) in the above equation, we obtain the Riccati equation (4.13).

The KKT condition for dual feasibility implies that $\lambda \geq 0$, so (4.13) has a unique stabilizing solution. Further, the KKT condition for complementary slackness implies $\lambda[\mathcal{P}(K^*(\lambda)) - \delta] = 0$. Thus, if $\lambda > 0$, then $\mathcal{P}(K^*(\lambda)) = \delta$. If $\lambda = 0$, then the solution to (4.13) is $X = 0$. This implies that $K^*(0) = 0$, which is feasible only if $\delta = \mathcal{P}(0)$. Thus, for any $\delta \in [\mathcal{P}(K_{\text{kf}}), \mathcal{P}(0)]$, it holds $\mathcal{P}(K^*(\lambda)) = \delta$.

Second-order sufficient conditions: We show that the stationary point (4.14) corresponds to a local minimum. We begin by computing the second-order differential of $\mathcal{S}(K)$. Taking the differential of (4.15) and noting that $d^2K = 0$, we get

$$\begin{aligned} d^2S &= \bar{A}d^2S\bar{A}^\mathsf{T} - 2dKCAdS\bar{A}^\mathsf{T} - 2\bar{A}dS(dKCA)^\mathsf{T} \\ &\quad + 2dK(I_p + CASA^\mathsf{T}C^\mathsf{T})dK^\mathsf{T} \triangleq \bar{A}d^2S\bar{A}^\mathsf{T} + Y \\ \Rightarrow d^2\mathcal{S}(K) &= \mathsf{tr}(d^2S) = \mathsf{tr}(YM) = -4\mathsf{tr}(dKCAdS\bar{A}^\mathsf{T}M) \\ &\quad + 2\mathsf{tr}(dK(I_p + CASA^\mathsf{T}C^\mathsf{T})dK^\mathsf{T}M). \end{aligned} \tag{4.20}$$

Similar analysis of (4.6) yields

$$\begin{aligned} d^2\mathcal{P}(K) &= -4\mathsf{tr}(dKCAdP\bar{A}^\mathsf{T}M) \\ &\quad + 2\mathsf{tr}[dK(R + CAPA^\mathsf{T}C^\mathsf{T} + CQC^\mathsf{T})dK^\mathsf{T}M]. \end{aligned} \tag{4.21}$$

Adding (4.20) and (4.21), we get

$$\begin{aligned} d^2\mathcal{L} &= -4\mathsf{tr}(dKCA\underbrace{(dS + \lambda dP)}_{\overset{(a)}{=}0.}\bar{A}^\mathsf{T}M) \\ &\quad + 2\mathsf{tr}[dKWdK^\mathsf{T}M] = \text{vec}^\mathsf{T}(dK)(2W \otimes M)\text{vec}(dK), \end{aligned}$$

where $W \triangleq I_p + \lambda R + CA(S + \lambda P)A^\mathsf{T}C^\mathsf{T} + \lambda CQC^\mathsf{T}$, and where (a) holds because $d\mathcal{L}(K, \lambda) = 0$ at the stationary point. The above expression implies that the Hessian of the Lagrangian is given by $H = 2W \otimes M$, which is positive-definite because $W > 0$ and $M > 0$. Thus, the considered stationary point corresponds to a local minimum.

Uniqueness of λ: Next, we show that for a given δ, the equation $\mathcal{P}(K^*(\lambda)) = \delta$ has a unique solution. Note that for a given $\lambda > 0$, the optimal gain $K^*(\lambda)$ in (4.14) is the unique minimizer of the cost $\mathcal{C}(K) = \mathcal{S}(K) + \lambda\mathcal{P}(K)$. Let $\lambda_2 > \lambda_1 > 0$. Then, we have

$$\mathcal{S}(K^*(\lambda_1)) + \lambda_1\mathcal{P}(K^*(\lambda_1)) < \mathcal{S}(K^*(\lambda_2)) + \lambda_1\mathcal{P}(K^*(\lambda_2)),$$

$$\mathcal{S}(K^*(\lambda_2)) + \lambda_2\mathcal{P}(K^*(\lambda_2)) < \mathcal{S}(K^*(\lambda_1)) + \lambda_2\mathcal{P}(K^*(\lambda_1)).$$

Adding the above two equations, we get $\mathcal{P}(K^*(\lambda_2)) < \mathcal{P}(K^*(\lambda_1))$. Thus, $\mathcal{P}(K^*(\lambda))$ is a strictly decreasing function of λ, and therefore, it is one-to-one.

To conclude the proof, since the necessary and sufficient conditions for a local minimum are satisfied by a unique gain, the local minimum is also the global minimum. ■

Corollary 23 ***(Properties of*** $\mathcal{P}^*(\lambda)$***)*** *The error $\mathcal{P}^*(\lambda)$ defined in Theorem 22 is a strictly decreasing function of λ.*

Theorem 22 shows that the optimal gain can be characterized in terms of a scalar parameter λ, which depends on the performance level δ according to the relation $\mathcal{P}^*(\lambda) = \delta$. Notice that $\lambda = 0$ if $\delta = \mathcal{P}(0)$, and λ approaches infinity as δ approaches $\mathcal{P}(K_{\text{kf}})$. In other words, $\lim_{\lambda \to \infty} K^*(\lambda) = K_{\text{kf}}$. Further, Corollary 23 implies that for a given δ, the solution of $\mathcal{P}^*(\lambda) = \delta$ can be found efficiently. For instance, one can use the bisection algorithm on the interval $[0, \lambda_{\max}]$, where $\mathcal{P}^*(\lambda_{\max}) > \delta$. These results also imply a fundamental tradeoff between performance and robustness of the estimator.

Theorem 24 ***(Accuracy vs robustness tradeoff)*** *Let $\mathcal{S}^*(\delta)$ denote the solution of (5.3). Then, $\mathcal{S}^*(\delta)$ is a strictly decreasing function of δ in the interval $\delta \in [\mathcal{P}(K_{\text{kf}}), \mathcal{P}(0)]$.*

Proof. From the proof of Theorem 22, we have

$$\left.\frac{\partial \mathcal{S}(K)}{\partial K}\right|_{K^*(\lambda)} = -\lambda \left.\frac{\partial \mathcal{P}(K)}{\partial K}\right|_{K^*(\lambda)} . \tag{4.22}$$

Since $\lambda > 0$ for $\delta \in [\mathcal{P}(K_{\text{kf}}), \mathcal{P}(0)]$ and $\mathcal{P}^*(\lambda) = \delta$, (4.22) implies that the sensitivity decreases when the error increases, and vice versa, so that a strict tradeoff exists. ■

Theorem 24 implies that there exists a fundamental tradeoff between the accuracy and robustness of a linear filter against perturbations to measurement noise covariance matrix. Therefore, the robustness of the linear filter in (4.3) in uncertain or adversarial environments can be improved only at the expenses of its accuracy in nominal conditions. Conversely, improving the robustness of the filter leads to a lower accuracy in nominal conditions.

Remark 25 ***(Design of optimally robust filters)*** *Let $\Delta R \geq 0$ denote a sufficiently small perturbation to R such that the approximation $\Delta\mathcal{P}(K) \approx \mathsf{tr}(K^{\mathsf{T}} M K \Delta R)$ holds (see (4.12)). Further, let ΔR be bounded as $\mathsf{tr}(\Delta R) \leq \gamma$. Then, we have*

$$\begin{aligned}\Delta\mathcal{P}(K) = \mathsf{tr}(K^{\mathsf{T}} M K \Delta R) &\leq \mathsf{tr}(K^{\mathsf{T}} M K)\rho(\Delta R) \\ &= \mathsf{tr}(S(K))\rho(\Delta R) \leq \gamma \mathcal{S}(K).\end{aligned}$$

Thus, given a gain K, the worst case performance degradation due to a bounded perturbation to R is given by $\mathcal{P}_{\text{worst}}(K) = \mathcal{P}(K) + \gamma\mathcal{S}(K)$. Therefore, a filter that is optimally robust (that *is, it exhibits optimal worst-case performance in the presence of norm-bounded perturbations of the noise statistics) can be obtained by minimizing* $\mathcal{P}_{\text{worst}}(K)$. *Note that this minimization problem is akin to the problem* (5.3), *and that its solution is given by* (4.14) *with* $\lambda = \gamma^{-1}$.

□

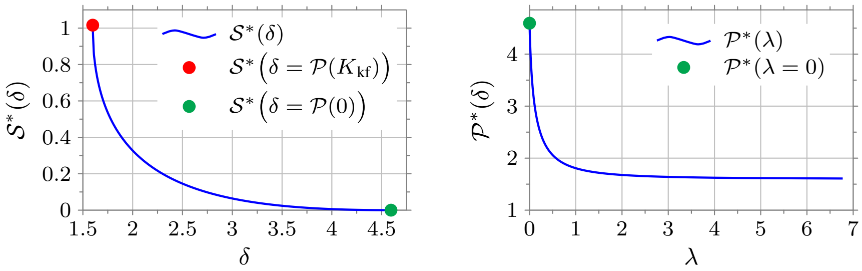
Remark 26 ***(Analysis when the system matrix*** A ***is unstable)*** *The accuracy-robustness tradeoff shown above also holds when* A *is unstable and* (A, C) *is detectable. The analysis for this case follows the same reasoning as above, except that the range of interest for the error becomes* $\delta \in [\mathcal{P}(K_{\text{kf}}), \mathcal{P}(K^*_{\mathcal{S}})]$, *with* $K^*_{\mathcal{S}} = \arg\min_K \mathcal{S}(K)$. *If* A *does not have eigenvalues on the unit circle, then the Riccati equation* (4.13) *has a unique solution for* $\lambda = 0$ *[41] (Theorem 12.6.2), and* $K^*_{\mathcal{S}} = K^*(0)$ *(c.f.* (4.14)*). In this case,* $\mathcal{P}(K^*_{\mathcal{S}})$ *is finite. The case when* A *has eigenvalues on the unit circle is more involved, finding* $K^*_{\mathcal{S}}$ *is not trivial, and* $\mathcal{P}(K^*_{\mathcal{S}})$ *may become arbitrarily large. This aspect is left for future research (see Section 4.3 for an example with unit eigenvalues).* □

We conclude this section with an illustrative example.

Example 27 ***(Robustness versus performance tradeoff)*** *Consider the system in* (5.1) *and* (4.2) *with matrices*

$$\begin{aligned} A &= \begin{bmatrix} 0.9 & 0 \\ 0.02 & 0.8 \end{bmatrix}, \qquad C = \begin{bmatrix} 0.5 & -0.8 \\ 0 & 0.7 \end{bmatrix}, \\ Q &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.7 \end{bmatrix}, \qquad R = \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & 0.8 \end{bmatrix}. \end{aligned} \tag{4.23}$$

Fig. 4.2(a) shows the values $\mathcal{S}^*(\delta)$ obtained from (5.3) over the range $\delta \in [\mathcal{P}(K_{kf}), \mathcal{P}(0)]$. Several comments are in order. First, as predicted by Theorem 24, the plot shows a tradeoff between accuracy and robustness. Second, in accordance with Theorem 22, the solution to the minimization problem (5.3) implies that the equality constraint in (5.3) is active. Third, when $\delta = \mathcal{P}(K_{\mathrm{kf}})$, the minimization problem (5.3) returns the Kalman gain. Fourth, although the Kalman filter (depicted by the red dot) achieves the highest accuracy, it features the highest sensitivity (thus, lowest robustness) among the solutions of (5.3) over the range $\delta \in [\mathcal{P}(K_{\mathrm{kf}}), \mathcal{P}(0)]$. Thus, the estimator that is most accurate on the nominal data, is also the most sensitive to perturbations. Fifth, the linear filter obtained when $\delta = \mathcal{P}(0)$ exhibits the worst nominal performance, but is the most robust to changes in the noise statistics. Fig. 4.2(b) shows the values of $\mathcal{P}^*(\lambda)$ as a function of λ. We observe that $\mathcal{P}^*(\lambda)$ is a strictly decreasing function in λ in accordance with Corollary 23. We also observe that the linear filter obtained when $\delta = \mathcal{P}(0)$, depicted by the green dot, has $\lambda = 0$. Finally, the value $\mathcal{P}^*(\lambda)$ obtained when $\delta = \mathcal{P}(K_{kf})$ cannot be shown since it requires $\lambda = \infty$. □



(a) (b)

Figure 4.2: Panel (a) shows the accuracy versus robustness tradeoff for the linear estimator (4.3) and the system described in Example 27. The red dot denotes the Kalman filter, and the green dot denotes the linear filter with zero gain. The Kalman filter achieves optimal performance with the nominal data, yet it is the most sensitive to changes of the noise statistics. The opposite tradeoff holds for the filter with zero gain. Panel (b) shows the estimation error as a function of λ for the system described in Example 27. The green dot denotes the filter with zero gain. The performance of the Kalman filter does not appear in the plot since it requires $\lambda = \infty$.

4.3 Accuracy vs robustness tradeoff in perception-based control

In this section we illustrate the implication of our theoretical results to the perception-based control setting shown in Fig. 4.1. We consider a vehicle obeying the dynamics [24]

$$x(t+1) = \underbrace{\begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 & 0 \\ T_s & 0 \\ 0 & 0 \\ 0 & T_s \end{bmatrix}}_{B} u(t) + w(t), \tag{4.24}$$

where $x(t) \in \mathbb{R}^4$ contains the vehicle's position and velocity in cartesian coordinates, $u(t) \in \mathbb{R}^2$ is the input signal, $w(t) \in \mathbb{R}^4$ is the process noise which follows the same assumptions as in (5.1), and T_s is the sampling time. We let the vehicle be equipped with a camera, whose images are used to extract measurements of the vehicle's position. In particular, let

$$y(t) = f_p\big(Z(t)\big) \tag{4.25}$$

denote the measurement equation, where $y(t) \in \mathbb{R}^2$ contains measurements of the vehicle's position, $Z(t) \in \mathbb{R}^{p\times q}$ describes the $p\times q$ pixel images taken by camera, and $f_p : \mathbb{R}^{p\times q} \to \mathbb{R}^2$ is the perception map between the camera's images and the vehicle's position. We approximate (4.25) with the following linear measurement model (see also [24]):

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{C} x(t) + v(t), \tag{4.26}$$

where $v(t) \in \mathbb{R}^2$ denotes the measurement noise, which is assumed to follow the same assumptions as in (4.2).

We consider the problem of tracking a reference trajectory using the measurements (4.26) and the dynamic controller

$$\begin{aligned} x_{\mathrm{c}}(t+1) =&(I - KC)(A - BL)x_{\mathrm{c}}(t) \\ &+ K(y(t+1) - Cx_{\mathrm{d}}(t+1)), \\ u(t) =& - Lx_{\mathrm{c}}(t) + u_{\mathrm{d}}(t), \end{aligned} \tag{4.27}$$

where L denotes the Linear-Quadratic-Regulator gain with error and input weighing matrices $W_x > 0$ and $W_u > 0$, K the gain of a stable linear estimator as in (4.3),[2] x_{d} the desired state trajectory, and u_{d} the control input generating x_{d}.

[2] If K equals the gain of the Kalman filter for the given system, then the controller (4.27) corresponds to the Linear-Quadratic-Gaussian regulator.

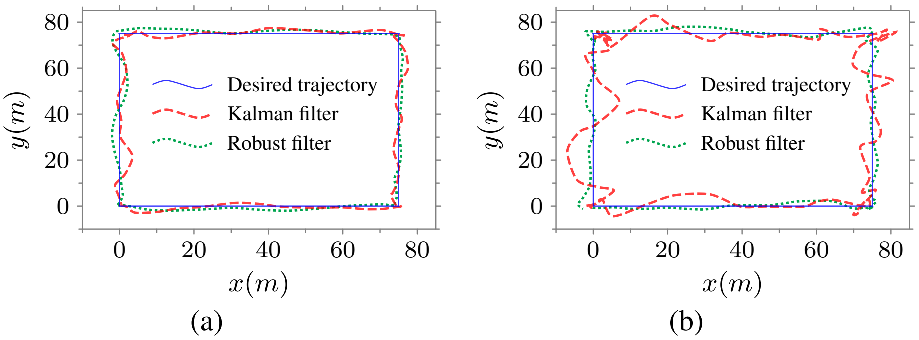


Figure 4.3: Panel (a) shows the trajectory tracking performance for the controller (4.27) with the Kalman filter (dashed red line) and a robust filter (dotted green line) in nominal noise statistics (the desired trajectory is shown by the solid blue line). The controller with the Kalman filter outperforms the other. Panel (b) shows the tracking performance for the two controllers using non-nominal noise statistics. In non-nominal conditions, the controller with the Kalman filter performs worse than the controller with the robust filter. The performance of a controller is measured based on the mean squared deviation between the controlled and nominal trajectories (see also Fig. 4.4).

The statistics of the measurement noise in (4.26) depend on how the perception map is trained and the data samples used for the training. We aim to show that, if the estimator's gain in (4.27) is designed to minimize the estimation error based on the learned noise statistics, then the performance of the perception-based controller (4.27) degrades significantly if the learned statistics differ from the actual noise statistics. Conversely, if the estimator's gain in (4.27) is designed based on Remark 25, then the performance of the perception-based controller (4.27) remains robust across different values of the noise statistics, although lower than the performance of the optimal estimator operating with the nominal noise statistics. Fig. 4.3 shows the trajectory tracking performance for the controller (4.27) for the Kalman filter and a robust filter with $T_s = 1, Q = 0.1I_4, R = 0.1I_2, W_x = \mathrm{diag}(100, 10^{-3}, 100, 10^{-3}), W_u = 10^{-3}I_2$. The robust filter corresponds to $\lambda = 0.307$ (see (4.14)). The non-nominal covariance is $\bar{R} = 2.5I_2$. We observe that the controller based on the Kalman filter performs better in nominal conditions, while the controller

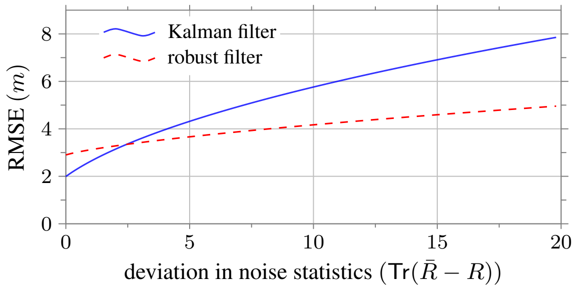


Figure 4.4: This figure shows the root mean square error (RMSE) of the controller (4.27) with the Kalman filter (solid blue line) and the robust filter (dashed red line), as a function of deviation between the measurement noise statistics. For small deviations, the controller using the Kalman filter outperforms the other. For large deviations, the controller using the robust filter outperforms the controller using the Kalman filter.

based on the robust filter performs better in non-nominal conditions, as predicted by our theoretical results. Fig. 4.4 shows the error of the Kalman filter and the robust filter as a function of the changes of the measurement noise covariance. We notice that for small deviations (near-nominal conditions), the controller based on the Kalman filter performs better than the controller based on the robust filter. However, when the deviation of the noise statistics becomes substantially large, the controller based on the robust filter performs better, thereby validating our theoretical tradeoff.

As shown in Fig. 4.1(b), the perception error may not be normally distributed, especially in the case of non-nominal measurements. Although our theoretical results were obtained under the assumption that the measurement (perception) error is normally distributed, we next numerically show that a tradeoff still exists when the measurement (perception) error is not Gaussian. To this aim, we consider the system in (5.10) and (4.26), where the measurement noise is distributed as in Fig. 4.1(b) (these distributions are computed numerically using the simulator CARLA [27]). We design 6 estimators using (4.14)

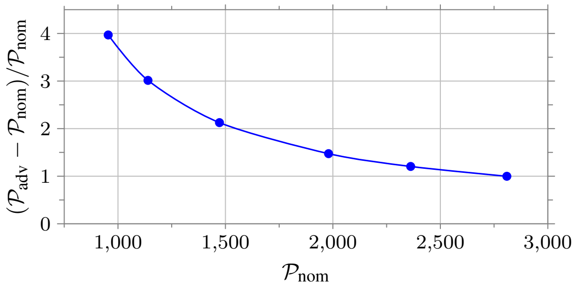


Figure 4.5: For the system (5.10) and (4.26) with measurement error distributed as in Fig. 4.1(b), this figure shows the performance $\mathcal{P}_{\text{nom}}$ (i.e., trace of estimation error covariance) and the approximate sensitivity $(\mathcal{P}_{\text{adv}} - \mathcal{P}_{\text{nom}})/\mathcal{P}_{\text{nom}}$ for 6 different estimators obtained from (4.14) by varying the desired accuracy δ. Although the measurement error is not normally distributed, a tradeoff still emerges between the accuracy of the estimators and their sensitivity.

with different values of δ, and test the performance of each estimator in nominal and non-nominal conditions. The performance of each estimator in nominal and non-nominal environments, denoted by $\mathcal{P}_{\text{nom}}$ and $\mathcal{P}_{\text{adv}}$, respectively, is computed using the sample error covariance computed from the obtained samples of the estimation error in nominal and non-nominal conditions. We approximate the sensitivity of these estimators as the relative degradation of the nominal performance when operating in non-nominal conditions, that is, as $(\mathcal{P}_{\text{adv}} - \mathcal{P}_{\text{nom}})/\mathcal{P}_{\text{nom}}$. Fig. 4.5 shows the performance and approximate sensitivity of the estimators. It can be seen that, even when the measurement error is not normally distributed, the estimator with largest (respectively, smallest) accuracy also has highest (respectively, smallest) sensitivity. These numerical results suggest that a tradeoff exists independently of the statistical properties of the measurement error.

We conclude by showing that the identified tradeoff between accuracy and robustness of linear estimators also constrain the performance of closed-loop perception-based

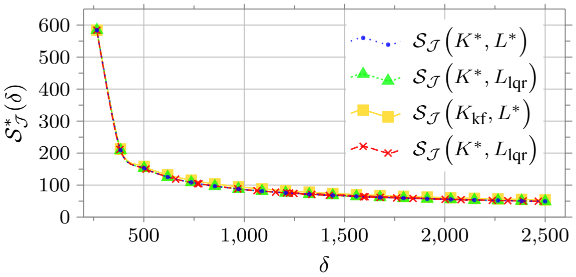


Figure 4.6: This figure shows the accuracy versus robustness tradeoff in the closed loop setting described in Section 4.3. The blue, green, and yellow lines denote the solution of (4.30), where in the blue line we optimize over both gains, in the green line we fix the controller to the LQR gain and optimize over the estimator only, and in the yellow line we fix the estimator to the Kalman gain and optimize over the controller only. The red line denotes the tradeoff between the accuracy in (4.28) and the sensitivity in (4.29) with the estimator gain given in (4.14) and the controller fixed to the LQR gain.

control algorithms. To this aim, consider the system (5.10) with controller (4.27), where both the estimator gain K and the controller gain L are now design parameters. For weighing matrices $W_x > 0$ and $W_u > 0$, let the performance of (4.27) be

$$\mathcal{J}(K, L) = \mathbb{E}\left[\frac{1}{T}\left(\sum_{t=0}^{T} x(t)^{\mathsf{T}} W_x x(t) + u(t)^{\mathsf{T}} W_u u(t)\right)\right], \tag{4.28}$$

where T denotes the time horizon. Notice that a lower value of the cost J is desirable, and the minimum (for $T \to \infty$) is achieved by choosing the Kalman gain K_{kf} with the linear quadratic regulator gain L_{lqr} for the matrices W_x and W_u. We adopt the following definition of sensitivity (this metric is the equivalent of (4.7) for the closed-loop performance):

$$\mathcal{S}_{\mathcal{J}}(K, L) \triangleq \mathsf{tr}\left[\frac{\partial \mathcal{J}(K, L)}{\partial R}\right], \tag{4.29}$$

where R is the noise covariance matrix of (4.26). To see if a tradeoff exists beween performance and sensitivity of the closed-loop controller, we solve the following problem:

$$\begin{aligned} \mathcal{S}^*_{\mathcal{J}}(\delta) = \min_{K,L} \quad & \mathcal{S}_{\mathcal{J}}(K, L) \\ \text{s.t.} \quad & \mathcal{J}(K, L) \leq \delta, \end{aligned} \tag{4.30}$$

where δ is a constant satisfying $\delta \geq \mathcal{J}(K_{\mathrm{kf}}, L_{\mathrm{lqr}})$. Notice that the minimization problem (4.30) is similar to (5.3) for the considered closed-loop control setting. The results of the minimization problem (4.30) are reported in Fig. 4.6, where it can be seen that a tradeoff between the performance of the controller (4.27) and its sensitivity still exists. Interestingly, our numerical results show that the tradeoff curve can be obtained, equivalently, by optimizing over both the controller and the estimator gain, by fixing the controller gain to be the LQR gain and optimizing over the estimator gain, or by fixing the estimator gain to be the Kalman gain and optimizing over the controller gain. Further, if the controller gain is chosen to be the optimal LQR gain, then the estimator gain that solves (4.30) coincides with the estimator gain obtained in Theorem 22. We leave a formal characterization of these properties as the subject of future investigation.

Part II

Benchmarking and Data-driven Control Design

Chapter 5

Learning Lipschitz Feedback Policies From Expert Demonstrations: Closed-Loop Guarantees, Robustness and Generalization

In this chapter, we propose a framework in which we use a Lipschitz-constrained loss minimization scheme to learn feedback control policies with guarantees on closed-loop stability, adversarial robustness, and generalization. These policies are learned directly from expert demonstrations, contained in a dataset of state-control input pairs, without any prior knowledge of the task and system model. Our analysis exploits the Lipschitz property of the learned policies to obtain closed-loop guarantees on stability, adversarial robustness, and generalization over scenarios unexplored by the expert. In particular, first, we establish robust closed-loop stability under the learned control policy, where we provide guarantees that the closed-loop trajectory under the learned policy stays within a bounded region around the expert trajectory and converges asymptotically to a bounded region around the origin. Second, we derive bounds on the closed-loop regret with respect to the expert policy and on the deterioration of the closed-loop performance under bounded (adversarial) disturbances to the state measurements. These bounds provide certificates for closed-loop performance and adversarial robustness for learned policies. Third, we derive a (probabilistic) bound on generalization error for the learned policies. Numerical results validate our analysis and demonstrate the effectiveness of our robust feedback policy learning framework. Finally, our results support the existence of a potential tradeoff between nominal closed-loop performance and adversarial robustness, and that improvements in nominal closed-loop performance can only be made at the expense of robustness to adversarial perturbations. The results of this chapter are reported in our published paper [68].

5.1 Problem formulation and outline of the approach

In this section, we setup the problem of learning robust feedback control policies from expert demonstrations and present an outline of our approach.

5.1.1 Problem setup

We begin by specifying the properties of the system, the control task and the dataset of expert demonstrations. Consider a discrete-time nonlinear system of the form:

$$x_{t+1} = f(x_t, u_t), \qquad y_t = x_t + \delta_t, \tag{5.1}$$

where the map $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the dynamics, $x_t \in \mathbb{R}^n$ the state, $u_t \in \mathbb{R}^m$ the control input and $y_t \in \mathbb{R}^n$ the full-state measurement at time $t \in \mathbb{N}$, respectively, with disturbance $\|\delta_t\| \leq \zeta$ for any $t \in \mathbb{N}$.[1]

Assumption 28 (*System properties*) *The following properties hold for System (5.1):*

(i) ***Fixed point at origin:*** *The map f in (5.1) has a fixed point at the origin (i.e., $f(0,0) = 0$).*

(ii) ***Lipschitz continuous dynamics:*** *The map f in (5.1) is Lipschitz continuous with constants ℓ_f^x and ℓ_f^u (i.e., $\|f(x_1, u_1) - f(x_2, u_2)\| \leq \ell_f^x \|x_1 - x_2\| + \ell_f^u \|u_1 - u_2\|$ for any $x_1, x_2 \in \mathbb{R}^n$ and $u_1, u_2 \in \mathbb{R}^m$).*

(iii) ***Exponential stabilizability by Lipschitz feedback:*** *System (5.1) is uniformly exponentially stabilizable by Lipschitz feedback, i.e., there exists a Lipschitz continuous feedback policy π and constants $M \in \mathbb{R}_{\geq 0}$, $\beta \in (0,1)$ such that $\left\|f_\pi^t(x)\right\| \leq M\beta^t \|x\|$.* □

We now explain the motivation behind the above assumptions on the properties of System (5.1). The control task is often formulated as one of stabilizing the system to the origin. Assumption 28-(i) states that the origin, in the absence of control input, is indeed

[1] The output equation allows for the modeling of sensors that are susceptible to bounded (adversarial) disturbances [77]. Note that in this work we consider perturbations that only appear in the output equation.

a fixed point of the system. The Lipschitz continuity Assumption 28-(ii) specifies the level of regularity intrinsic to the system dynamics and is fairly standard in the literature. From a control design perspective it is crucial that the system indeed possesses the desired stabilizability properties from within this class of feedback policies considered in design. In this paper, we seek to learn feedback policies with Lipschitz regularity and Assumption 28 specifies that this is the case and that System (5.1) is exponentially stabilizable by Lipschitz feedback.

The task is one of infinite-horizon discounted optimal control of System (5.1) by a Lipschitz-continuous feedback policy, with stage cost $c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ and discount factor $\gamma \in (0, 1)$:

$$\begin{aligned} \min_{\pi \in \mathrm{Lip}(\mathbb{R}^n;\mathbb{R}^m)} \quad & \sum_{t=0}^{\infty} \gamma^t c(x_t, u_t), \\ \text{s.t.} \quad & \begin{cases} x_{t+1} & = f(x_t, u_t), \\ u_t & = \pi(x_t + \delta_t), \end{cases} \end{aligned} \tag{5.2}$$

where $\mathrm{Lip}(\mathbb{R}^n; \mathbb{R}^m)$ is the space of Lipschitz-continuous feedback policies. Furthermore, we would like the closed-loop performance to be robust to the disturbance δ.

Assumption 29 (*Task properties*) *The following hold for Task (5.2) and System (5.1):*

(i) ***Strong convexity and smoothness of stage cost:*** *The stage cost $c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is μ-strongly convex and λ-smooth. Furthermore, $c(x, u) = 0$ if and only if $x = 0$ and $u = 0$.*

(ii) ***Existence of optimal feedback policy:*** *For every $\gamma \in (0, 1)$, there exists a minimizer $\pi^* \in \mathrm{Lip}(\mathbb{R}^n; \mathbb{R}^m)$ to the optimal control problem (5.2) with $\delta \equiv 0$.* □

The choice of optimal control cost function plays an important role in determining the properties of the optimal feedback policy. Assumption 29-(i) specifies the convexity and smoothness properties of the control cost. Existence of an optimal feedback policy within the considered class, as specified in Assumption 29-(ii) is a minimum requirement for control design.

We now verify the properties in Assumptions 28 and 29 in the Linear-Quadratic control setting.

Example 30 (*Linear quadratic control*) *For a linear system with $f(x,u) = Ax + Bu$ such that (A, B) is a controllable pair, it can be seen that the properties in Assumption 28 readily follow. It can be seen that a quadratic stage cost $c(x,u) = x^\top Qx + 2x^\top Wu + u^\top Ru$ (with $Q \succ 0$ and $R - W^\top Q^{-1} W \succ 0$) is strongly convex and has a Lipschitz-continuous gradient with $\mu = \lambda_{\min}(H)$, $\lambda = \lambda_{\max}(H)$, and $H = 2\begin{bmatrix} Q & W \\ W^\top & R \end{bmatrix}$, thereby satisfying Assumption 29-(i). Furthermore, we note that Assumption 29-(ii) readily follows from the existence of an optimal feedback gain for the discounted infinite-horizon LQR problem, and the fact that the corresponding optimal value function is quadratic.*

In this paper, we consider the problem of data-driven feedback control, where we have access neither to the underlying dynamics f nor to the task cost function (stage cost c and discount factor γ). Instead, we have access to $N < \infty$ expert demonstrations of an (unknown) optimal feedback policy π^* on System (5.1) over a finite horizon of length T. The initial state of the demonstrations is sampled uniformly i.i.d. from $B_r(0) \subset \mathbb{R}^n$, the ball of radius r centered at the origin. The data is collected in the form of matrices X, U

as follows:

$$X = \begin{bmatrix} \mathbf{x}^{(1)} & \dots & \mathbf{x}^{(N)} \end{bmatrix}, \quad U = \begin{bmatrix} \mathbf{u}^{(1)} & \dots & \mathbf{u}^{(N)} \end{bmatrix},$$

where $\mathbf{x}^{(i)} = (x_0^{(i)}, \dots, x_T^{(i)})$ and $\mathbf{u}^{(i)} = (u_0^{(i)}, \dots, u_{T-1}^{(i)})$ are the state and input vectors from the i-th demonstration, satisfying $u_t^{(i)} = \pi^*(x_t^{(i)})$ for all $i \in \{1, \dots, N\}$ and $t \in \{0, \dots, T-1\}$.

5.1.2 Outline of the approach

Our objective is to learn a feedback policy from the dataset X, U of expert demonstrations to solve the control task (5.2) while remaining robust to (adversarial) disturbances δ of full-state measurements. To this end, we seek an optimization-based learning formulation that allows us to explicitly constrain the sensitivity of the learned policy to (adversarial) disturbances. The Lipschitz constant of the learned policy serves as a measure of its sensitivity to disturbances, and we thereby formulate the (adversarially) robust policy learning problem as a Lipschitz-constrained policy learning problem [52]:

$$\begin{aligned} \min_{\pi \in \mathrm{Lip}(B_r(0);\mathbb{R}^m)} \quad & \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=0}^{T-1} L\left(\pi(x_t^{(i)}), u_t^{(i)}\right), \\ \text{s.t.} \quad & \mathrm{lip}(\pi) \leq \alpha, \end{aligned} \tag{5.3}$$

where L is a strictly convex and Lipschitz continuous loss function for the learning problem, $\mathrm{lip}(\pi)$ is the Lipschitz constant of the policy π, and $\alpha \in \mathbb{R}_{\geq 0}$ is a target upper bound for the Lipschitz constant of the learned policy $\widehat{\pi}$ (the minimizer in (5.3)). The Lipschitz constraint in (5.3) serves as a mechanism to induce robustness of the learned policy to disturbances δ (the smaller the parameter α, the more robust the policy $\widehat{\pi}$ is to the disturbances δ [30]). Figure 5.1 illustrates our setup.

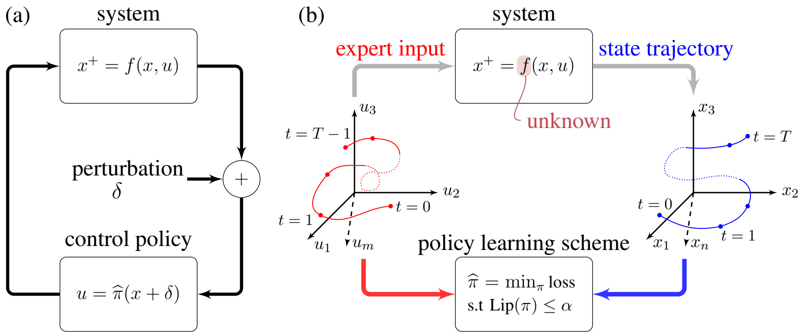


Figure 5.1: The block diagram in panel (a) corresponds to the implementation of the learned control policy $\widehat{\pi}$ in non-nominal conditions under adversarial perturbations δ on the state measurement. Panel (b) illustrates the Lipschitz-constrained policy learning scheme implemented on the expert generated dataset to obtain policy $\widehat{\pi}$.

From here, we divide our analysis into two parts. In the first part of our analysis we consider the closed-loop control aspect of the problem. In particular, for a given worst-case learning error bound $\|\widehat{\pi} - \pi^*\|_\infty \triangleq \sup_{x \in B_r(0)} \|\widehat{\pi}(x) - \pi^*(x)\| \leq \varepsilon$ and a bound on the Lipschitz constant of the learned policy, $\text{lip}(\widehat{\pi}) \leq \alpha$, we establish (i) a robust closed-loop stability bound as a function of ε and α, and (ii) bounds on the closed-loop performance and robustness as a function of ε and α. In the second part of our analysis, we tackle the learning aspect of the problem. In particular, we use the robustness of the learned policy (imposed by the Lipschitz constraint in (5.3)), along with a bound on the training error, to obtain a probabilistic bound on the violation of $\|\widehat{\pi} - \pi^*\|_\infty \leq \varepsilon$. The two parts of our analysis complement each other, for instance, in order to satisfy target bounds on the closed-loop stability and performance, our analysis can be used to obtain a target bound on ε which must be satisfied by the learned policy, if some additional information on the system and task are available.

Note that, the loss function L defines the learning problem (5.3) and eventually shapes the learned policy and its deviation from the true policy, $\|\widehat{\pi} - \pi^*\|_\infty$. While the properties of L

do not enter our analysis, the bound on learning error ε does. Thus, since our focus is not to understand how the learning error depends on the properties of the loss function L, we find useful to use ε as a proxy for the learning performance and to quantify how closed-loop stability, performance and robustness depend on it.

We now develop appropriate notions of closed-loop performance and robustness under the feedback policies learned from expert demonstrations. We note that the control task (5.2), being one of optimal control of System (5.1), has a natural performance metric given by the value function. In what follows, we make use of the following notations, we let $f_\pi(x) = f(x, \pi(x))$ and $c_\pi(x) = c(x, \pi(x))$. Let $V^{\widehat{\pi}}$ be the value function associated with the learned feedback policy $\widehat{\pi}$ for System (5.1):

$$\begin{aligned} V^{\widehat{\pi}}(x_0) &= \sum_{t=0}^{\infty} \gamma^t c_{\hat{\pi}} \left(f^t_{\widehat{\pi}}(x_0) \right), \\ \text{where} \quad f^t_\pi(x) &\triangleq \underbrace{f_\pi \circ \ldots \circ f_\pi}_{t \text{ times}}(x). \end{aligned} \tag{5.4}$$

Since the expert implements the optimal policy π^*, the performance of the learned policy can be measured by its regret with respect to the expert policy π^*. The regret associated with the learned policy $\widehat{\pi}$ relative to the expert policy π^* is:

$$\mathcal{R}(\widehat{\pi}) = \sup_{x \in B_r(0)} \left\{ V^{\widehat{\pi}}(x) - V^*(x) \right\}. \tag{5.5}$$

When $\mathcal{R}(\widehat{\pi}) = 0$, the performance of the learned policy equals the performance of an optimal policy for the control task (5.2). Conversely, the performance of the learned policy degrades as $\mathcal{R}(\widehat{\pi})$ increases. Naturally, the objective of the policy learning problem is now to minimize the regret incurred by the learned policy $\widehat{\pi}$. Note that this is a more important performance metric in the closed-loop setting than the loss function L used for learning in (5.3), as it encodes the cost incurred by the evolution of the system under the learned feedback policy. We now note that the regret $\mathcal{R}$ only measures the performance of the learned policy under nominal conditions (in the absence of perturbations on the state measurements) and does not shed light on its performance in the presence of adversarial perturbations. This calls for an appropriate robustness metric, for which we will use the regret associated with the policy $\widehat{\pi}$ when subject to perturbations relative to when deployed under nominal conditions, that is,

$$\mathcal{S}(\widehat{\pi}) = \sup_{\substack{x \in B_r(0) \\ \delta \in \mathbb{R}^n, \|\delta\| \leq \zeta}} \left\{ V^{\widehat{\pi}_\delta}(x) - V^{\widehat{\pi}}(x) \right\}, \tag{5.6}$$

where $\widehat{\pi}_\delta(x) = \widehat{\pi}(x + \delta)$. Intuitively, if $\mathcal{S}(\widehat{\pi})$ is small, then the performance of the policy $\widehat{\pi}$ under perturbation is close to its performance in nominal conditions, and $\widehat{\pi}$ is robust to feedback perturbations. Again, we note that this robustness metric measures closed-loop robustness by encoding the cost incurred by the evolution of the system under the learned feedback policy subject to feedback perturbations. We would ideally like to keep both $\mathcal{R}$ and $\mathcal{S}$ low, which would imply that the policy performs well both under nominal conditions and when subjected to feedback perturbations. However, we shall see later that there may exist tradeoffs between the two objectives, presenting an obstacle to such a goal. Note that we use the supremum in (5.5) and (5.6) since we are interested in performing worst-case analysis in order to provide closed-loop performance and robustness certificates.

We now address some crucial technical issues arising in the closed-loop dynamic setting in relation to minimizing the performance metrics $\mathcal{R}$ and $\mathcal{S}$. We note that the policy learning problem (5.3) is formulated over the set $B_r(0) \in \mathbb{R}^n$, which is the region of interest containing the data from expert demonstrations. Now, in order to measure the performance of a learned policy $\widehat{\pi}$ using the metrics $\mathcal{R}$ and $\mathcal{S}$, we must first ensure that the closed-loop trajectories of the system, under policy $\widehat{\pi}$, remain in $B_r(0)$ (for initial conditions in $B_r(0)$). In the absence of such a guarantee, the metrics $\mathcal{R}$ and $\mathcal{S}$ are likely to be unbounded, and would therefore not serve as useful measures of performance. We therefore obtain robust stability bounds that specify the conditions under which closed-loop trajectories remain bounded in $B_r(0)$.

5.2 Robust closed-loop stability and performance

In this section, we present the theoretical results underlying the robust feedback policy learning framework outlined in Section 5.1. The results are presented in three parts: (a) We first present a closed-loop stability analysis for System (5.1) under learned feedback control policies (learned using (5.3)) satisfying a given bound on their distance from the optimal feedback policy π^* (the minimizer in (5.2)). In Lemma 31-(ii), we establish that the closed-loop system under optimal feedback π^* is exponentially stable. In Theorem 34 we then establish a robust stability guarantee (to bounded adversarial disturbances on the state measurements) for feedback control policies satisfying a given bound on their distance from π^*. (b) We then present an analysis of performance on the control task (5.2) under feedback control policies satisfying a given bound on their distance from π^*. Theorem 37-(i) provides an upper bound on the regret incurred by a policy with respect to the expert policy π^*. Theorem 37-(ii) quantifies the robustness of the closed-loop performance in terms of the Lipschitz constant of the feedback policy. (c) We finally present an analysis of the Lipschitz-constrained policy learning problem (5.3). In Theorem 40, we provide a generalization bound for the maximum learning error incurred in the region of interest $B_r(0)$, i.e., $\|\widehat{\pi} - \pi^*\|_\infty$ in terms of the covering radius of the training dataset. This learning error bound is to be combined with the previous closed-loop stability and performance guarantees for policies which were established for policies satisfying a given error bound.

5.2.1 Robust stability with learned feedback policy

We first present the following result on the quadratic boundedness of the optimal value function:

Lemma 31 (Optimal value function and feedback policy) *(i) Quadratic boundedness of optimal value function: There exist $\underline{\kappa}, \bar{\kappa} \in \mathbb{R}_{\geq 0}$ with $\underline{\kappa} < \bar{\kappa}$ such that the optimal value function in (5.2), for any $\gamma \in (0,1)$, satisfies $\underline{\kappa}\|x\|^2 \leq V^*(x) \leq \bar{\kappa}\|x\|^2$.*

(ii) Exponential stability under optimal feedback policy: Let $\gamma' = 1 - \underline{\kappa}/\bar{\kappa}$. For any $\gamma \in (\gamma', 1)$, the closed-loop trajectory starting from any $x \in \mathbb{R}^n$ and generated by the optimal policy π^* in (5.2) satisfies:

$$\left\|f^t_{\pi^*}(x)\right\| \leq \sqrt{\frac{\bar{\kappa}}{\underline{\kappa}}}\left(\frac{\gamma'}{\gamma}\right)^{t/2}\|x\|.$$

We now make the following assumption on the existence of constants $\underline{\kappa}^*, \bar{\kappa}^*$ in Lemma 31 satisfying certain bounds:

Assumption 32 *There exist $\underline{\kappa}^*, \bar{\kappa}^* \in \mathbb{R}_{>0}$ (with $\underline{\kappa}^* < \bar{\kappa}^*$) such that for any $\gamma \in (1 - \underline{\kappa}^*/\bar{\kappa}^*, 1)$:*

(i) $\underline{\kappa}^*\|x\|^2 \leq V^*(x) \leq \bar{\kappa}^*\|x\|^2$,

(ii) f_{π^*} is contractive (i.e., $\ell_{f_{\pi^*}} < 1$).

We now verify the properties in Assumptions 32 in the Linear-Quadratic control setting.

Example 33 (*Linear quadratic control*) *Consider a linear system with $f(x_t, u_t) = Ax_t + Bu_t$ such that (A, B) is a controllable pair, and a quadratic stage cost $c(x_t, u_t) = x_t^\top Q x_t + u_t^\top R u_t$ (with $Q \succ 0$ and $R \succ 0$) for $t \geq 0$. For any $\gamma \in (1 - \underline{\kappa}^*/\bar{\kappa}^*, 1)$ and $x = x_0$, the optimal discounted LQR value function can be written as $V^*(x) = x^\mathsf{T} P^* x$, where P^* satisfies the following Riccati equation [14, section 4.3]:*

$$P^* = \gamma A^\mathsf{T} \big(P^* - \gamma^2 P^* B \big(\gamma B^\mathsf{T} P^* B + R\big)^{-1} B^\mathsf{T} P^*\big) A + Q.$$

For the value function $V^*(x)$, we have $\underline{\kappa}^* = |\lambda_{min}(P^*)|$ and $\bar{\kappa}^* = |\lambda_{max}(P^*)|$. Furthermore, the closed-loop dynamics, f_{π^*}, associated with the optimal LQR policy, $\pi^*(x_t) = -K^* x_t$, is contractive with $\ell_{f_{\pi^*}} = |\lambda_{max}(A - BK^*)| < 1$ for $t \geq 0$, where $K^* = \gamma(\gamma B^\mathsf{T} P^* B + R)^{-1} B^\mathsf{T} P^* A$.

The following theorem establishes robust stability of the closed-loop system under the learned policy $\widehat{\pi}$ from a bound on the policy error $\|\widehat{\pi}(x) - \pi^*(x)\|_\infty$ and measurement disturbances δ:

Theorem 34 (*Robust exponential stability under Lipschitz policy*) *Let π^* be the minimizer in (5.2) for some $\gamma \in (1 - \underline{\kappa}^*/\bar{\kappa}^*, 1)$, and let $\widehat{\pi}$ be any policy such that $\|\widehat{\pi} - \pi^*\|_\infty \leq \varepsilon$ and $\operatorname{lip}(\widehat{\pi}) \leq \alpha$. Let $\alpha\zeta + \varepsilon \leq \big(1 - \ell_{f_{\pi^*}}\big) r / \ell_f^u$ and let $\|\delta_t\| \leq \zeta$ for all $t \in \mathbb{N}$. For the closed-loop trajectory $f^t_{\widehat{\pi}_\delta}(x)$ starting from $x \in B_r(0)$ and generated by the policy $\widehat{\pi}_\delta$, the following holds:*

$$\|f^t_{\widehat{\pi}_\delta}(x) - f^t_{\pi^*}(x)\| \leq \left[\frac{1 - \ell^t_{f_{\pi^*}}}{1 - \ell_{f_{\pi^*}}}\right] \ell_f^u (\alpha\zeta + \varepsilon).$$

We refer the reader to subsection 5.5.2 for the proof. The robust stability result can be understood in the sense of input-to-state stability [47,88], in that we exploit the exponential stability result for the expert policy π^* and treat the learned policy $\widehat{\pi}$ as a perturbation on π^*. By obtaining boundedness of the learning error along the closed-loop trajectory, we establish that the closed-loop trajectory under the learned policy both stays within a bounded region around the optimal trajectory and converges asymptotically to a bounded region around the origin.

5.2.2 Regret and robustness with learned feedback policy

Having clarified the issue of robust stability, we now present a regret analysis for the learned control policy $\widehat{\pi}$. We first present the following lemma on an incremental exponential stability property of exponentially stabilizing Lipschitz feedback policies on $B_r(0)$:

Lemma 35 (*Incremental exponential stability*) *Let π be an exponentially stabilizing Lipschitz feedback policy for System (5.1) such that $\left\|f_\pi^t(x)\right\| \leq M\beta^t\|x\|$ for some $M \in \mathbb{R}_{\geq 0}$ and $\beta \in (0,1)$. For $x, x' \in B_r(0)$, there exists $\bar{M} \in \mathbb{R}_{\geq 0}$ such that $\left\|f_\pi^t(x) - f_\pi^t(x')\right\| \leq \bar{M}\beta^t\|x - x'\|$.*

From Lemmas 31-(ii) and 35, it follows that the optimal policy π^* is incrementally exponentially stable. We now make the following assumption that this property holds for policies in a neighborhood of π^*:

Assumption 36 (*Incremental exponential stability*) *Let $\gamma \in (1 - \underline{\kappa}^*/\bar{\kappa}^*, 1)$ be such that for any $\pi \in \mathrm{Lip}(B_r(0); \mathbb{R}^m)$ satisfying $\|\pi - \pi^*\|_\infty \leq \left(1 - \ell_{f_{\pi^*}}\right) r/\ell_f^u$, the closed-loop dynamics f_π is incrementally exponentially stable as in Lemma 35.*

The following theorem establishes a bound on the sub-optimality of the closed-loop performance of system (5.1) with $\widehat{\pi}$ and a robustness bound for the deterioration of the closed-loop performance under bounded disturbances:

Theorem 37 (***Regret and robustness of learned policy***) *Let $\bar{M}, \beta$ be as specified in Lemma 35 and Assumption 36, and let π^* be the minimizer in (5.2) for some $\gamma \in (1 - \underline{\kappa}^*/\bar{\kappa}^*\ ,\ 1)$ and $\operatorname{lip}(\pi^*) = \alpha^*$. Let $\widehat{\pi}$ be any policy such that $\|\widehat{\pi} - \pi^*\|_\infty \leq \varepsilon$ and $\operatorname{lip}(\widehat{\pi}) \leq \alpha$. Furthermore, let $\alpha\zeta + \varepsilon \leq \left(1 - \ell_{f_{\pi^*}}\right) r/\ell_f^u$.*

(i) Regret: *The regret $\mathcal{R}$ of the policy $\widehat{\pi}$ relative to π^*, as defined in (5.5), satisfies:*

$$\mathcal{R}(\widehat{\pi}) \leq \frac{\lambda}{1-\gamma}\left[c_1 r\sqrt{1 + |\max\{\alpha, \alpha^*\}|^2}\ \Delta + \frac{1}{2}c_2\Delta^2\right],$$

where

$$c_1 = 1 + \gamma\Theta\ell_f^u, \qquad c_2 = 1 + \gamma\Theta\sqrt{1 + \alpha^{*2}}\ (\ell_f^u)^2,$$

$$\Delta = \|\widehat{\pi} - \pi^*\|_\infty, \qquad \Theta = \bar{M}^2/(1 - \gamma\beta^2).$$

(ii) Robustness: *Let $\|\delta_t\| \leq \zeta$ for any $t \in \mathbb{N}$. For any $\gamma \in (0, 1)$, the robustness metric $\mathcal{S}$ of the policy $\widehat{\pi}$, as defined in (5.6), satisfies:*

$$\mathcal{S}(\widehat{\pi}) \leq \frac{\lambda\alpha}{1-\gamma}\left[d_1 r\alpha\sqrt{1+\alpha^2}\ \zeta + \frac{1}{2}d_2\alpha^2\zeta^2\right],$$

where

$$d_1 = 1 + \gamma\Theta\ell_f^u, \qquad d_2 = 1 + \gamma\Theta\sqrt{1+\alpha^2}\ (\ell_f^u)^2.$$

We refer the reader to subsection 5.5.4 for the proof. Theorem 37-(i) establishes that the regret bound for the learned policy scales quadratically with the deviation of the learned policy from the expert (optimal) policy. We also note that the regret bound scales with λ, the Lipschitz constant of the gradient of the stage cost, and the Lipschitz constant of the dynamics (w.r.t. u), as they modulate the sensitivity to variations of the input. Furthermore, we want the performance of the learned policy under disturbances to be close its nominal performance, i.e., a low value of $\mathcal{S}$. Theorem 37-(ii) establishes that the robustness of performance is determined by the sensitivity of the learned policy to disturbances, in particular that the robustness bound scales quadratically with the Lipschitz constant of the learned policy. Theorem 37-(ii) provides the designer with a robustness guarantee while implementing the learned policy in the presence of bounded (possibly adversarial) disturbances to measurements. We note that, the bounds in Theorem 37 might be loose. This is because we consider worst-case analysis (where we use supremum in (5.5) and (5.6)), which is unavoidable if we want to provide closed-loop performance and robustness certificates over all possible scenarios. Although the bounds might be loose, they are informative and intuitive, where they provide qualitative understanding of the properties that affect the closed-loop performance and robustness of the learned policy $\widehat{\pi}$. Further, the bounds in Theorem 37 provide insights on how to tune the Lipschitz bound, α, in (5.3) as pointed out next in Remark 39. We end this section with the following Remarks.

Remark 38 (*Tradeoff between* $\mathcal{R}(\widehat{\pi})$ *and* $\mathcal{S}(\widehat{\pi})$) *We note that in the limit of the expert demonstrations, $N \to \infty$, Theorem 37 suggests a tradeoff between the regret $\mathcal{R}(\widehat{\pi})$ and the robustness metric $\mathcal{S}(\widehat{\pi})$ as we vary the Lipschitz bound α in (5.3). As we decrease α, the deviation of the learned policy $\widehat{\pi}$ from the optimal policy π^* increases, and so does the bound in Theorem 37-(i) (via an increase in ε). Instead, as we increase α such that the constraint* *in* (5.3) *is no longer active, the learned policy converges to the optimal policy* π^*, *and the bound in Theorem 37-(i) decreases to zero. Similarly, as we decrease* α, *the Lipschitz constant of the learned policy,* $\ell_{\widehat{\pi}}$, *decreases, and so does the bound in Theorem 37-(ii). See Fig. 5.7 in Section 5.4 for an illustration of this tradeoff. Furthermore, we see that strong convexity of the cost induces stability properties and* λ*-smoothness allows for the tuning of regret.*

Remark 39 (***Selection of the Lipschitz bound*** α) *As noted in Remark 38, the regrets* $\mathcal{R}(\widehat{\pi})$ *and* $\mathcal{S}(\widehat{\pi})$, *as well as their upper bounds, exhibit a tradeoff relation upon tuning the Lipschitz bound* α *in* (5.3). *Hence, an "optimal choice of* α*" that optimizes both regrets generally does not exist. In practice, the selection of* α *depends on the control application at hand and its performance and robustness requirements. Note that in practice, we cannot compute the regrets* $\mathcal{R}(\widehat{\pi})$ *and* $\mathcal{S}(\widehat{\pi})$ *since we do not have access to the system dynamics nor the cost function of the control task. However, we can compute the upper bounds in Theorem 37, which we can use as a benchmark to tune* α *such that the control application requirements are met. We also note that even if the regret bounds are finite, Assumption 36 might still get violated for some values of* α *and hence stability will not be guaranteed. Therefore, when selecting* α, *Assumption 36 need to be checked in order to guarantee stability.*

5.3 Lipschitz-constrained policy learning

We now present results from our analysis of the Lipschitz constrained policy learning problem (5.3). We note that the training data for the feedback policy learning problem (5.3) consists of evaluations of the expert policy π^* over a finite set of points

$\{x_t^{(i)}\} \subset B_r(0)$ in the state space, and the objective is to generalize over the region of interest $B_r(0)$. The following theorem establishes a (maximum) generalization error bound for the minimizer $\widehat{\pi}$ over the region $B_r(0)$ in terms of the covering radius of the training dataset:

Theorem 40 (***Generalization error bound for Lipschitz policy learning***) *Let $\widehat{\pi}$ be a minimizer in (5.3), $X_N = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with $\mathbf{x}^{(i)} = \left(x_0^{(i)}, \ldots, x_{T-1}^{(i)}\right)$ and let*

$$\varepsilon_{\text{train}} = \max_{\substack{i \in \{1,\ldots,N\},\\ t \in \{0,\ldots,T-1\}}} \left\| \widehat{\pi}(x_t^{(i)}) - \pi^*(x_t^{(i)}) \right\|,$$

$$\rho(X_N, r) = \sup_{x \in B_r(0)} \min_{\substack{i \in \{1,\ldots,N\},\\ t \in \{0,\ldots,T-1\}}} \left| x - x_t^{(i)} \right|.$$

For any $\delta > 0$, the maximum learning error in $B_r(0)$ satisfies:

$$\mathbb{P}\left[\|\widehat{\pi} - \pi^*\|_\infty > (\alpha + \alpha^*)\delta + \varepsilon_{\text{train}} \right] \leq \mathbb{P}\left[\rho(X_N, r) > \delta\right].$$

We refer the reader to subsection 5.5.5 for a proof of this result. Theorem 40 shows that although a larger α allows for achieving a lower $\varepsilon_{\text{train}}$, it can result in worse generalization performance. This is due to the fact that the $(\alpha + \alpha^*)\delta$ term in the bound scales linearly with α, which can potentially result in a higher maximum learning error $\|\widehat{\pi} - \pi^*\|_\infty$. Furthermore, we note that the covering radius $\rho(X_N, r)$ of the training dataset X_N controls the (probabilistic) bound on the generalization error[2].

We finally note that Theorems 34 and 37 establish robust stability and performance bounds for policies $\widehat{\pi}$ that satisfy (i) $\|\widehat{\pi} - \pi^*\|_\infty \leq \varepsilon$, and (ii) $\text{lip}(\widehat{\pi}) \leq \alpha$, whereas Theorem 40 yields a (probabilistic) bound on the violation of the condition $\|\widehat{\pi} - \pi^*\|_\infty \leq \varepsilon$ for finite

[2] We refer the reader to [81] for finite sample estimates on the covering radius.

Algorithm 1 Graph-based Lipschitz policy learning

```
Input: Training data, Graph size $\eta$, Number of edges $|\mathcal{E}|$, Lipschitz bound $\alpha$, Number of iterations $k$
1: Sample $\eta$ points (graph vertices) uniformly i.i.d. from $B_r(0)$
2: Partition training dataset as in (5.7)
3: Implement $k$ iterations of primal-dual algorithm (5.9)
Output: Minimizer $\widehat{\mathbf{u}}$
```

datasets of N expert trajectories (while the Lipschitz bound still holds). Therefore, by combining the bounds in Theorems 34 and 37 with the finite sample bound in Theorem 40, we obtain the desired closed-loop generalization and robustness bounds.

We now present a graph-based Lipschitz policy learning algorithm to solve (5.3). We sample η points $\{X_i\}_{i=1}^{\eta}$, uniformly i.i.d from $B_r(0)$. Considering the points $\{X_i\}_{i=1}^{\eta}$ as the (embedding of) vertices, we construct an undirected, weighted, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = \{1, \ldots, \eta\}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, we consider the weights of all edges to have a unit value. We then define a partition $\mathcal{W} = \{\mathcal{W}_i\}_{i=1}^{\eta}$ of the training dataset $D = \{x_t^{(i)}\}$ (set of points in the state space where evaluations of the expert policy are available) as follows:

$$\mathcal{W}_i = \{x \in D \mid \ |x - X_i| \leq |x - X_j| \ \ \forall \ j \in \mathcal{V} \setminus \{i\}\} \, . \tag{5.7}$$

Finally, we write the discrete (empirical) Lipschitz-constrained policy learning problem over the graph $\mathcal{G}$ as follows (which can be viewed as the discretization of (5.3) over the graph $\mathcal{G}$):

$$
\begin{aligned}
&\min_{\substack{\widehat{\mathbf{u}}=(\widehat{u}_1,\ldots,\widehat{u}_\eta),\\ \widehat{u}_i\in\mathbb{R}^m}} \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{I}(\mathcal{W}_i)} L(\widehat{u}_i, u_j),\\
&\text{s.t.} \quad |\widehat{u}_r - \widehat{u}_s| \leq \alpha \, |X_r - X_s| \,, \quad \forall \; (r, s) \in \mathcal{E},
\end{aligned} \tag{5.8}
$$

where u_j correspond to the input coming from the expert policy. We note that Problem (5.8) is convex (strictly convex objective function with convex constraints) and it can be solved using any off-the-shelf convex optimization solver. Note that the solution to Problem (5.8) is a set of input vectors $\hat{u} = (\hat{u}_1, \cdots \hat{u}_\eta)$, where the value of $\hat{u}_i$ corresponds to the vertex i of $\mathcal{G}$ $\forall i \in \mathcal{V}$. The learned policy $\widehat{\pi}$ corresponds to the learned graph $\mathcal{G}$ with $\{\hat{u}\}_{i=1}^{\eta}$ being the input values obtained by solving (5.8). The learned policy $\widehat{\pi}$ takes a state measurement x_t as feedback and thereupon perform first-order interpolation among the nearest vertices to get the corresponding input u_t.[3]

5.4 Numerical experiments

In this section, first, we present the primal-dual algorithm to solve Problem (5.8), then, we present the results from numerical experiments applying our algorithm to (i) learn the Linear Quadratic Regulator (LQR), and (ii) learn nonlinear control for a nonholonomic system (differential drive mobile robot).

[3] Note that, the first-order interpolation do not violate the Lipschitz bound on lip($\hat{\pi}$)

5.4.1 Primal-dual algorithm

The Lagrangian of Problem (5.8) is given by:

$$\begin{aligned}\mathcal{L}_{\mathcal{G}}(\widehat{\mathbf{u}},\Phi) = &\sum_{i\in\mathcal{V}}\Bigg[\sum_{s\in\mathcal{I}(\mathcal{W}_i)} L(\widehat{u}_i,u_s)\\ &+\frac{1}{2}\sum_{j\in\mathcal{V}}\phi_{ij}\left(|\widehat{u}_i-\widehat{u}_j|^2-\alpha\,|X_i-X_j|^2\right)\Bigg],\end{aligned}$$

where $\Phi = [\phi_{ij}]_{i,j=1}^{\eta}$ is the matrix of Lagrange multiplier for the pairwise Lipschitz constraints. Define a primal-dual dynamics for the Lagrangian $\mathcal{L}_{\mathcal{G}}(\widehat{\mathbf{u}},\Phi)$ with time-step sequence $\{h(k)\}_{k\in\mathbb{N}}$:

$$\begin{aligned}\widehat{\mathbf{u}}(k+1) &= \widehat{\mathbf{u}}(k) - h(k)\ \nabla_{\widehat{\mathbf{u}}}\mathcal{L}_{\mathcal{G}}\left(\widehat{\mathbf{u}}(k),\Lambda(k)\right),\\ \Phi(k+1) &= \max\{0\ ,\ \Phi(k) + h(k)\ \nabla_{\Phi}\mathcal{L}_{\mathcal{G}}\left(\widehat{\mathbf{u}}(k),\Phi(k)\right)\}.\end{aligned} \tag{5.9}$$

The primal dynamics is a discretized heat flow over the graph $\mathcal{G}$ with a weighted Laplacian, where $\nabla_{\widehat{\mathbf{u}}}\mathcal{L}_{\mathcal{G}}\left(\widehat{\mathbf{u}}(k),\Phi(k)\right) = \left(\Delta(\Phi)\otimes I_{\dim(\mathbb{Y})}\right)\widehat{\mathbf{u}} + \nabla_1 L(\widehat{\mathbf{u}},\mathbf{u})$, and $\Delta(\Phi)$ is the Φ-weighted Laplacian of the graph $\mathcal{G}$. The convergence of the solution $\{(\widehat{\mathbf{u}}(k),\Phi(k))\}_{k\in\mathbb{N}}$ of the primal-dual dynamics (5.9) to the saddle point of the Lagrangian $\mathcal{L}_{\mathcal{G}}$ follows [5] from the convexity of Problem (5.8). The primal-dual algorithm that solves Problem (5.8) is presented in Alg. 1.

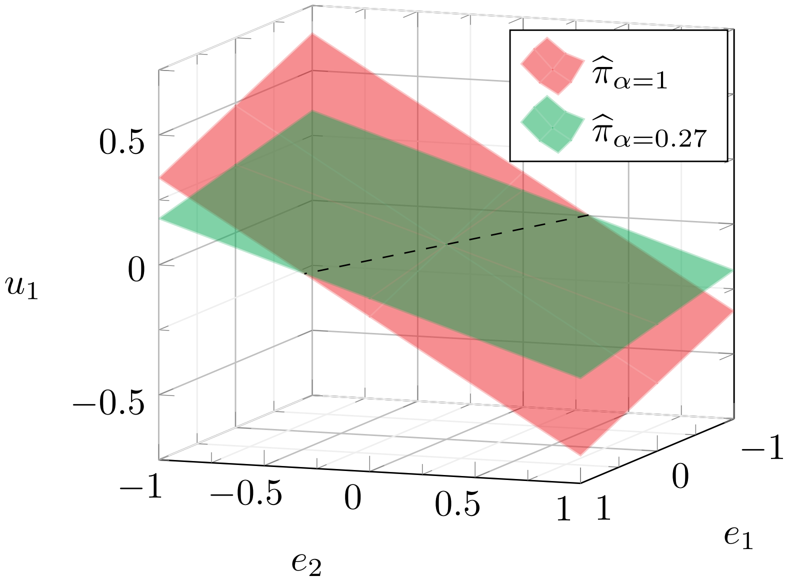
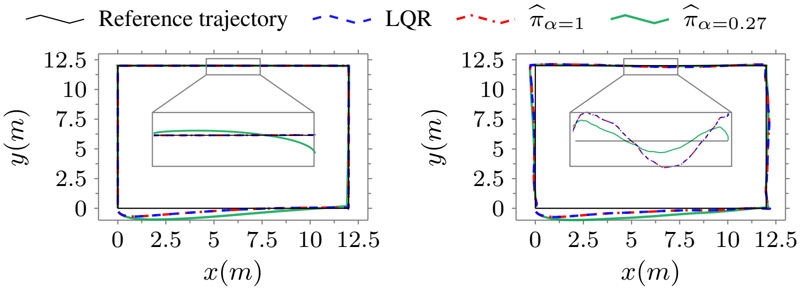


Figure 5.2: This figure shows the surface of policy $\widehat{\pi}$ in the state space for system (5.11), which is learned using Alg. 1 with $\alpha = 1$ (red surface) and $\alpha = 0.27$ (green surface), and the expert being the LQR for system (5.11).



(a)

(b)

Figure 5.3: Panel (a) and panel (b) show the trajectory tracking performance for the LQR (dashed blue line), the learned policy $\widehat{\pi}$ learned using Alg. 1 with $\alpha = 1$ (dash-dotted red line) and $\alpha = 0.27$ (solid green line) for the settings described in section 5.4.2. In panel (a), the policies are deployed in nominal conditions. The policy $\widehat{\pi}_{\alpha=1}$ performs as good as the LQR while the policy $\widehat{\pi}_{\alpha=0.27}$ performs poorly compared to the LQR and $\widehat{\pi}_{\alpha=1}$. In panel (b), the policies are deployed in non-nominal conditions. The performance of the LQR and policy $\widehat{\pi}_{\alpha=1}$ is worse than when deployed in nominal conditions, while the performance of policy $\widehat{\pi}_{\alpha=0.27}$ in non-nominal conditions remains almost the same as in nominal conditions.

5.4.2 Learning the Linear Quadratic Regulator

We consider a vehicle obeying the following dynamics (see also [65] and [4]):

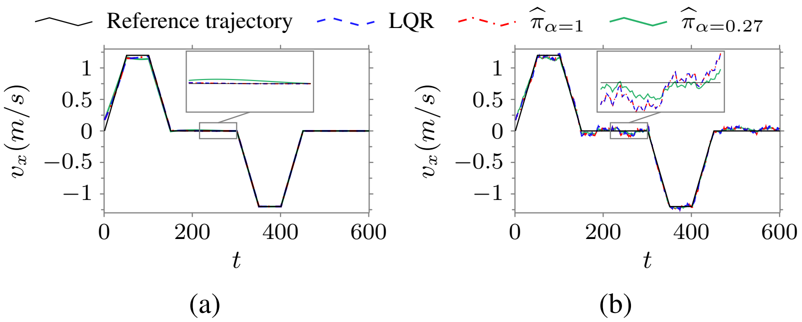
$$x_{t+1} = \underbrace{\begin{bmatrix} 1 & T_\mathrm{s} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_\mathrm{s} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix} 0 & 0 \\ T_\mathrm{s} & 0 \\ 0 & 0 \\ 0 & T_\mathrm{s} \end{bmatrix}}_{B} u_t, \quad y_t = x_t + \delta_t \tag{5.10}$$

where $x_t \in \mathbb{R}^4$ contains the vehicle's position and velocity in cartesian coordinates, $u_t \in \mathbb{R}^2$ the input signal, $y \in \mathbb{R}^4$ the state measurement, $\delta_t \in \mathbb{R}^4$ bounded measurement noise with $\|\delta_t\| \leq \zeta$ and $\zeta \in \mathbb{R}_{\geq 0}$, and T_s the sampling time. We consider the problem of tracking a reference trajectory, and we write the error dynamics and the controller as

$$e_{t+1} = Ae_t + B\bar{u}_t, \qquad u_t = \underbrace{-K(e_t + \delta_t)}_{\bar{u}_t} + v_t, \tag{5.11}$$

where $e_t = x_t - \xi_t$ is the error between the system state and the reference state, $\xi_t \in \mathbb{R}^4$ at time t, $v_t \in \mathbb{R}^2$ is the control input generating ξ_t, and K denotes the control gain. We consider the expert policy to correspond to the optimal LQR gain, K_lqr, which minimizes a discounted value function as in (5.2) but with horizon T, quadratic stage cost $c(e_t, \bar{u}_t) = e_t^\mathsf{T} Q e_t + \bar{u}_t^\mathsf{T} R \bar{u}_t$ with error and input weighing matrices $Q \succeq 0$ and $R \succ 0$, respectively. Notice that the quadratic stage cost is strongly convex and Lipschitz bounded over bounded space $e \in B_r(0) \subset \mathbb{R}^4$ and $\bar{u} \in \mathbb{R}^2$.

Expert demonstrations. We generate $N = 3000$ expert trajectories with time horizon $T = 600$ using (5.11) with $K = K_\mathrm{lqr}$, $T_\mathrm{s} = 0.1$, $\gamma = 0.82$, $Q = 0.1I_4$, $R = 0.1I_2$, and $\delta_t = 0$,



(a) (b)

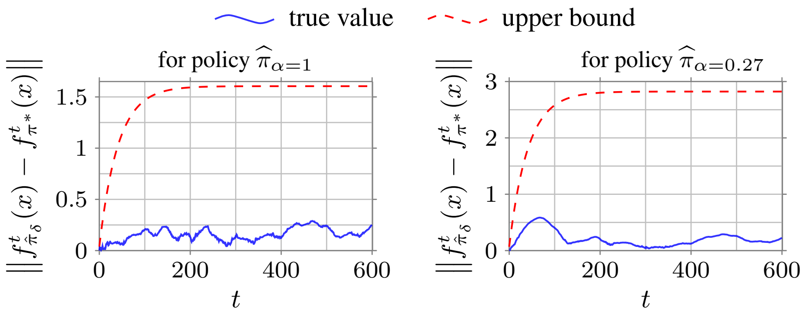
Figure 5.4: Panel (a) and panel (b) show the linear velocity tracking performance for the LQR (dashed blue line), the learned policy $\widehat{\pi}$ learned using Alg. 1 with $\alpha = 1$ (dash-dotted red line) and $\alpha = 0.27$ (solid green line) for the settings described in section 5.4.2. In panel (a), the policies are deployed in nominal conditions. The policy $\widehat{\pi}_{\alpha=1}$ performs as good as the LQR while the policy $\widehat{\pi}_{\alpha=0.27}$ performs poorly compared to the LQR and $\widehat{\pi}_{\alpha=1}$. In panel (b), the policies are deployed in non-nominal conditions. The performance of the LQR and policy $\widehat{\pi}_{\alpha=1}$ is worse than when deployed in nominal conditions, while the performance of policy $\widehat{\pi}_{\alpha=0.27}$ in non-nominal conditions remains almost the same as in nominal conditions.

contained in the data matrices E, U:

$$E = \begin{bmatrix} \mathbf{e}^{(1)} & \dots & \mathbf{e}^{(N)} \end{bmatrix}, \quad U = \begin{bmatrix} \mathbf{u}^{(1)} & \dots & \mathbf{u}^{(N)} \end{bmatrix},$$

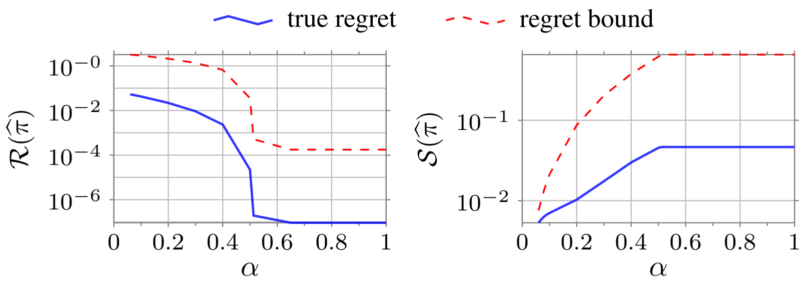
with $\mathbf{e}^{(i)} = (e_0^{(i)}, \dots, e_T^{(i)})$ and $\mathbf{u}^{(i)} = (u_0^{(i)}, \dots, u_{T-1}^{(i)})$. Each trajectory is generated with random initial condition, $e_0^{(i)} \in B_2(0)$ for $i = 1, \dots, N$. Note that, since the initial conditions, $e_0^{(i)}$, are inside $B_2(0)$ and $K = K_{\text{lqr}}$ is stabilizing, then, all the data points in E are inside $B_2(0)$. Furthermore, K_{lqr} obeys Assumption 32 (see Example 33) with $\underline{\kappa}^* = 3.73$, $\bar{\kappa}^* = 7.74$, and $\ell_{f_{\pi^*}} = 0.976 < 1$. Moreover, we have $\gamma = 0.82 \in (1 - \underline{\kappa}^*/\bar{\kappa}^*, 1)$.

Policy learning. For the learning Problem (5.8), we use the squared error loss $L = (\widehat{\pi}(x) - \pi^*(x))^2$. Using Alg. 1, we learn policy $\widehat{\pi}$ with $\alpha = 1$ and $\alpha = 0.27$ denoted by $\widehat{\pi}_{\alpha=1}$ and $\widehat{\pi}_{\alpha=0.27}$, respectively. Fig. 5.2 shows the surface of the learned policies $\widehat{\pi}_{\alpha=1}$ and



(a) (b)

Figure 5.5: For the settings described in 5.4.2, panel (a) and panel (b) show the deviation of the trajectory incurred by the learned policy $\widehat{\pi}$ from the expert trajectory (solid blue line) for $\alpha = 1$ and $\alpha = 0.27$, respectively, along with the corresponding robust stability bounds derived in Theorem 34 (dashed red line).



(a) (b)

Figure 5.6: Panel (a) and panel (b) show the true regrets $R(\widehat{\pi})$ and $S(\widehat{\pi})$ in (5.5) and (5.6) (solid blue line), and the regret bounds in Theorem 37 (dashed red line) as a function of the Lipschitz bound, α, in (5.3), respectively, for the settings described in section 5.4.2. The regret $R(\widehat{\pi})$ and the bound in Theorem 37-(i) decrease as α increases, as shown in panel (a), while The regret $S(\widehat{\pi})$ the bound in Theorem 37-(ii) increase with α, as shown in panel (b).

$\widehat{\pi}_{\alpha=0.27}$ in the state space. Note that, since the Lipschitz constant of the expert policy, $\pi^* = K_{\text{lqr}}$, is $\alpha^* = \|K_{\text{lqr}}\|_2 = 0.51 < \alpha = 1$, we get $\|\widehat{\pi}_{\alpha=1} - \pi^*\|_2 = 0$, which implies that $\widehat{\pi}_{\alpha=1}$ learns exactly the expert policy. On the other hand, since $\alpha = 0.27 < \alpha^* = 0.51$, we get $\|\widehat{\pi}_{\alpha=0.27} - \pi^*\|_2 = \epsilon$ for $\epsilon > 0$, which implies that $\widehat{\pi}_{\alpha=0.27}$ learns the expert policy with some learning error ϵ. As observed in Fig. 5.2, the Lipschitz constant constraints the slope of the learned surface, where $\widehat{\pi}_{\alpha=0.27}$ has smaller slope than $\widehat{\pi}_{\alpha=1}$, and hence more robust to perturbations in the states. However, smaller Lipschitz constant implies larger learning error, and hence poorer nominal performance.

Stability. Both policies $\widehat{\pi}_{\alpha=1}$ and $\widehat{\pi}_{\alpha=0.27}$ satisfy Assumption 36, i.e., $\|\widehat{\pi}_{\alpha=1} - \pi^*\|_\infty = 0 \leq \left(1 - \ell_{f_{\pi^*}}\right) r/\ell_f^u$ and $\|\widehat{\pi}_{\alpha=0.27} - \pi^*\|_\infty = 0.4745 \leq \left(1 - \ell_{f_{\pi^*}}\right) r/\ell_f^u$, which imply that both $\widehat{\pi}_{\alpha=1}$ and $\widehat{\pi}_{\alpha=0.27}$ are incrementally exponentially stable. Fig. 5.3 shows the trajectory tracking performance for the optimal LQR controller, $\widehat{\pi}_{\alpha=1}$, and $\widehat{\pi}_{\alpha=0.27}$. The policies are deployed in nominal conditions, Fig. 5.3(a), and in non-nominal conditions with $\zeta = 0.5$, Fig. 5.3(b). We observe in Fig. 5.3(a) that $\widehat{\pi}_{\alpha=1}$ performs better than $\widehat{\pi}_{\alpha=0.27}$ in nominal conditions. On the other hand, we observe in Fig. 5.3(b) that the performance of $\widehat{\pi}_{\alpha=1}$ degrades when deployed in non-nominal conditions, while the performance of $\widehat{\pi}_{\alpha=0.27}$ remains almost the same, as predicted by [52]. Fig. 5.4 shows the tracking performance of the vehicle's velocity in the x-direction for the optimal LQR controller, $\widehat{\pi}_{\alpha=1}$, and $\widehat{\pi}_{\alpha=0.27}$ versus time. The conditions in Fig. 5.4(a) and Fig. 5.4(b) are the same as in Fig. 5.3(a) and Fig. 5.3(b), respectively. Similarly as in Fig. 5.3, the policy $\widehat{\pi}_{\alpha=1}$ performs better than $\widehat{\pi}_{\alpha=0.27}$ in nominal conditions as observed in Fig. 5.4(a). Further, we observe in Fig. 5.4(b) that the performance of $\widehat{\pi}_{\alpha=1}$ degrades when deployed in non-nominal conditions, while the performance of $\widehat{\pi}_{\alpha=0.27}$ remains almost the same. Fig. 5.5 shows the deviation of the trajectories incurred by $\widehat{\pi}_{\alpha=1}$ and $\widehat{\pi}_{\alpha=0.27}$ from the expert trajectory, along with the corresponding stability bounds derived in Theorem 34. We observe that both learned policies do not violate the stability bound, which agrees with Theorem 34. Furthermore, we observe that the deviation between the expert trajectory and the trajectory incurred by $\widehat{\pi}_{\alpha=1}$, and the corresponding stability bounds (Fig. 5.5(a)) are lower than those of $\widehat{\pi}_{\alpha=0.27}$

(Fig. 5.5(b)), which is expected since $\|\widehat{\pi}_{\alpha=1} - \pi^*\|_\infty < \|\widehat{\pi}_{\alpha=0.27} - \pi^*\|_\infty$.

Regret bounds. The parameters of the bounds in Theorem 37 are obtained as follows, $\lambda = \max\{\rho(2Q), \rho(2R)\}$, $\alpha^* = \|K_{\text{lqr}}\|_2$, $\ell_f^u = \|B\|_2$, and $\Theta = \frac{1}{1-\gamma\rho(A+BK)^2}$, where K is a stabilizing gain. Fig. 5.6 shows the regrets $\mathcal{R}(\widehat{\pi})$ and $\mathcal{S}(\widehat{\pi})$ in (5.5) and (5.6), and the corresponding upper bounds derived in Theorem 37 as a function of the Lipschitz bound, α, in (5.3). As can be seen, the regret $\mathcal{R}(\widehat{\pi})$ and the corresponding upper bound in Theorem 37-(i) decrease as α increases, while the regret $\mathcal{S}(\widehat{\pi})$ and the corresponding upper bound in Theorem 37-(ii) increase with α. Further, the regrets and the bounds remains constant for $\alpha \geq 0.51$, since the constraint in (5.3) becomes inactive and $\widehat{\pi}$ converges to the optimal LQR controller. Fig. 5.7 shows the tradeoff between the regrets, as well as the tradeoff between the regrets upper bounds as we vary the Lipschitz bound, α, in (5.3). This suggests that improving the robustness of the learned policy to perturbations comes at the expenses of its nominal performance. We note that for $\alpha < 0.27$ Assumption 36 is violated, which implies that even if the regret bounds are not violated for $\alpha < 0.27$ in Fig. 5.6 and Fig. 5.7, stability is not guaranteed. Therefore, when selecting α using Fig. 5.7, Assumption 36 should be checked in order to guarantee stability. See Remark 39.

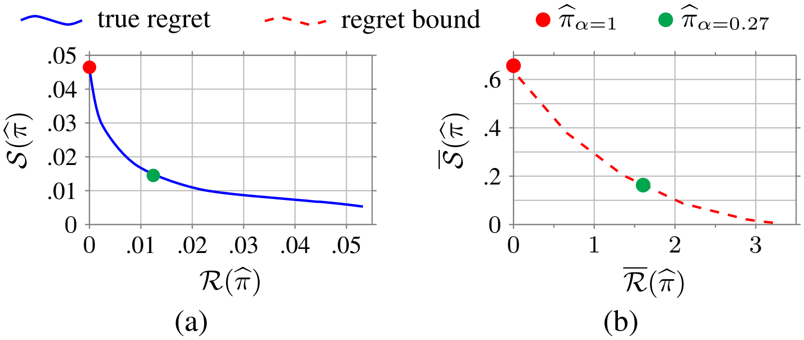


Figure 5.7: Panel (a) shows the tradeoff between the regrets $R(\widehat{\pi})$ and $S(\widehat{\pi})$ in (5.5) and (5.6), respectively, for the settings described in section 5.4.2. Panel (b) shows the tradeoff between the upper bounds of $R(\widehat{\pi})$ and $S(\widehat{\pi})$ derived in Theorem 37, respectively. In both panels, the red and the green dots represent the learned policies $\widehat{\pi}_{\alpha=1}$ and $\widehat{\pi}_{\alpha=0.27}$ used in Fig. 5.3, respectively.

5.4.3 Learning nonlinear control for nonholonomic system

We consider nonholonomic differential drive mobile robot (see Fig. 5.8) obeying the following discrete-time nonlinear dynamics

$$\begin{aligned} x_{t+1} &= T_s v_t \cos(\theta_t) + x_t, \qquad \text{for } t \geq 0 \\ y_{t+1} &= T_s v_t \sin(\theta_t) + y_t, \\ \theta_{t+1} &= \theta_t + T_s \omega_t \end{aligned} \tag{5.12}$$

where $x_t \in \mathbb{R}$ and $y_t \in \mathbb{R}$ are the position of the robot's centroid in the cartesian coordinate frame $(O; x, y)$, $\theta_t \in \mathbb{R}$ is the robot's orientation, $v_t \in \mathbb{R}$ and $\omega_t \in \mathbb{R}$ are the robot's forward and angular velocity at time t, respectively, which are the system's inputs, and $T_s > 0$ is the sampling time. The dynamics in (5.12), can be written in the following vector form

$$q_{t+1} = f(q_t, u_t), \qquad u_t = \pi(q_t + \delta_t) \qquad \text{for } t \geq 0, \tag{5.13}$$

where $q_t = [x_t, y_t, \theta_t]^\mathsf{T}$ is the state, $u_t = [v_t, \omega_t]^\mathsf{T}$ is the input, $f : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ is the dynamics, $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the control policy, and $\delta_t \in \mathbb{R}^3$ is a bounded perturbation, with

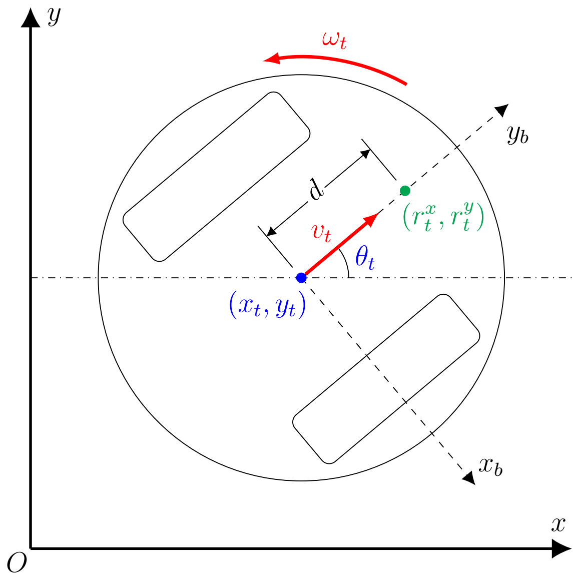


Figure 5.8: Differential drive mobile robot described in (5.12).

$\|\delta_t\| \leq \zeta \in \mathbb{R}_{\geq 0}$. Let $r_t = [r_t^x, r_t^y]^\mathsf{T}$ be a point fixed on the robot at a fixed distance d from $[x_t, y_t]^\mathsf{T}$ (see Fig. 5.8). The task is to stabilize the point r_t at $[0,0]^\mathsf{T}$, which is described by the following regulator problem

$$\begin{aligned} \min_{\pi \in \mathrm{Lip}(\mathbb{R}^3;\mathbb{R}^2)} \quad & \lim_{T\to\infty} \sum_{t=0}^{T} \gamma^t \left({r_t}^\mathsf{T} Q r_t + {u_t}^\mathsf{T} S(\theta_t)^\mathsf{T} R S(\theta_t) u_t \right), \\ \text{s.t.} \quad & \begin{cases} q_{t+1} & = f(q_t, u_t), \\ u_t & = \pi(q_t + \delta_t), \end{cases} \\ \text{where} \quad & S(\theta_t) = \begin{bmatrix} T_s \cos(\theta_t) & -dT_s \sin(\theta_t) \\ T_s \sin(\theta_t) & dT_s \cos(\theta_t) \end{bmatrix}, \end{aligned} \tag{5.14}$$

with γ denoting the discount factor, and $Q \succeq 0$ and $R \succ 0$ are weighing matrices.

Expert demonstrations. We consider the expert policy, π^*, to be the minimizer of (5.14). The derivation of π^* for this example is presented in subsection 5.5.6. We use π^* to generate

$N = 40$ expert trajectories with time horizon $T = 500$.

Policy learning. For the learning Problem (5.8), we use the squared error loss $L = (\widehat{\pi}(x) - \pi^*(x))^2$. Using Alg. 1, we learn policy $\widehat{\pi}$ with $\alpha = 50$ and $\alpha = 0.5$ denoted by $\widehat{\pi}_{\alpha=50}$ and $\widehat{\pi}_{\alpha=0.5}$, respectively. Fig. 5.9 shows the surface of the learned policies $\widehat{\pi}_{\alpha=50}$ and $\widehat{\pi}_{\alpha=0.5}$ that correspond to the input ω in the subspace $[x, y]^\intercal$ for $\theta = 0$. Since the Lipschitz constant of the expert policy, π^*, is $\alpha^* = 16.65 < \alpha = 50$, the policy $\widehat{\pi}_{\alpha=50}$ learns exactly the expert policy. On the other hand, since $\alpha = 0.5 < \alpha^* = 16.65$, the policy $\widehat{\pi}_{\alpha=0.5}$ learns the expert policy with some learning error. As observed in Fig. 5.9, the Lipschitz constant constraints the slope of the learned surface, where $\widehat{\pi}_{\alpha=0.5}$ has smaller slope than $\widehat{\pi}_{\alpha=50}$, and hence more robust to perturbations in the states. However, since $\widehat{\pi}_{\alpha=0.5}$ has larger learning error, it has poorer nominal performance. Fig. 5.10 shows the trajectory of the point (r_t^x, r_t^y) (see Fig. 5.8) induced by the expert policy, $\widehat{\pi}_{\alpha=50}$, and $\widehat{\pi}_{\alpha=0.5}$ starting from initial position $(1, 1)$ and an orientation $\theta = 180°$. The policies are deployed in nominal conditions, Fig. 5.10(a), and in non-nominal conditions with $\zeta_r = 0.7$ for the position and $\zeta_\theta = \pi/180$ for the orientation, Fig. 5.10(b). We observe in Fig. 5.10(a) that $\widehat{\pi}_{\alpha=50}$ performs as good as the expert and better than $\widehat{\pi}_{\alpha=0.5}$ in nominal conditions. On the other hand, we observe in Fig. 5.10(b) that the performance of $\widehat{\pi}_{\alpha=50}$ and that of the expert degrade when deployed in non-nominal conditions, while the performance of $\widehat{\pi}_{\alpha=0.5}$ remains almost the same.

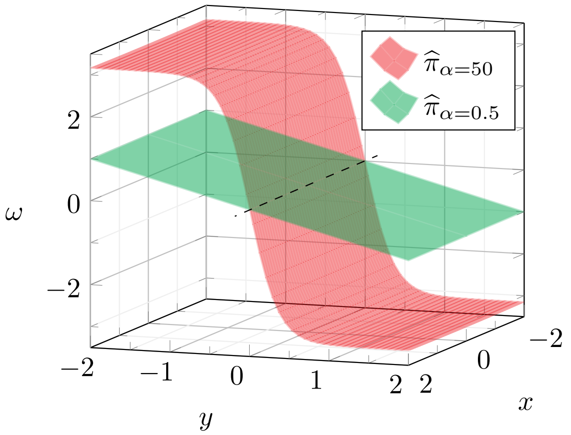


Figure 5.9: This figure shows the surface of policy $\widehat{\pi}$ that correspond to the input ω in the subspace $[x, y]^\mathsf{T}$ for system (5.13) for $\theta = 0$. Two policies are learned using Alg. 1 with $\alpha = 50$ (red surface) and $\alpha = 0.5$ (green surface), and the expert demonstrations are generated as in subsection 5.5.6.

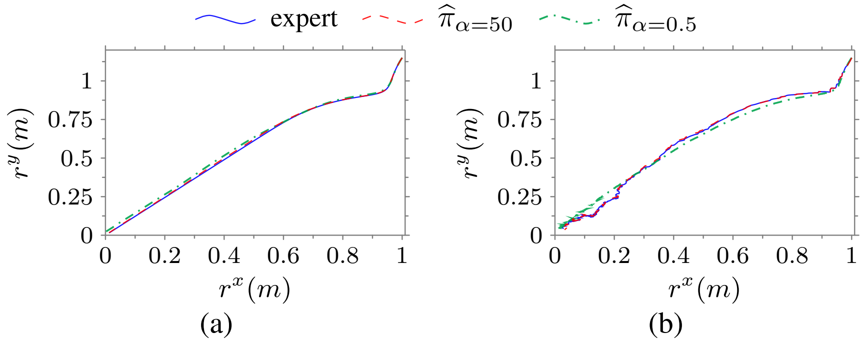


Figure 5.10: Panel (a) and panel (b) show the trajectory of the expert (solid blue line), the learned policy $\widehat{\pi}$ learned using Alg. 1 with $\alpha = 50$ (dashed red line) and $\alpha = 0.5$ (dash-dotted green line). In panel (a), the policies are deployed in nominal conditions. The policy $\widehat{\pi}_{\alpha=50}$ outputs the same trajectory as the expert while the policy $\widehat{\pi}_{\alpha=0.5}$ outputs a different trajectory towards the equilibrium. In panel (b), the policies are deployed in non-nominal conditions. The performance of the expert and policy $\widehat{\pi}_{\alpha=50}$ is worse than when deployed in nominal conditions, while the performance of policy $\widehat{\pi}_{\alpha=0.5}$ in non-nominal conditions remains almost the same as in nominal conditions.

5.5 Proofs of the main results and supplementary material

5.5.1 Proof of Lemma 31

(i) Quadratic bounds: We first note that $V^*(x) \geq 0$ for any $x \in \mathbb{R}^n$, and $\pi^*(0) = 0$. To see this, we first recall that $V^*(0) \leq \sum_{t=0}^{\infty} \gamma^t c(x_t, u_t)$ for $x_{t+1} = f(x_t, u_t)$ with $x_0 = 0$ and any $\{u_t\}$. In particular, with $u_t = 0$ for all $t \in \mathbb{N}$, we get that $\sum_{t=0}^{\infty} \gamma^t c(x_t, u_t) = 0$ (since $x_0 = 0$, $f(0,0) = 0$ and $c(0,0) = 0$), and since $0 \leq V^*(0) \leq \sum_{t=0}^{\infty} \gamma^t c(x_t, u_t) = 0$, it follows that $V^*(0) = 0$. Now, we have that $\pi^*(0) \in \arg\min_{u \in \mathbb{R}^m} c(0,u) + V^*(f(0,u))$, from which we clearly get that $\pi^*(0) = 0$ is the only minimizer. Now, since c is μ-strongly convex, with $c(0,0) = 0$, we have $c_{\pi^*}(x) \geq \mu \|x\|^2/2$. Furthermore, since $V^*(x) = \min_{u \in \mathbb{R}^m} c(x,u) + \gamma V^*(f(x,u))$, we get:

$$\begin{aligned} V^*(x) &= c_{\pi^*}(x) + \gamma V^*(f_{\pi^*}(x)) \\ &\geq \frac{\mu}{2}\|x\|^2 + \gamma V^*(f_{\pi^*}(x)) \geq \frac{\mu}{2}\|x\|^2. \end{aligned}$$

Let π be a Lipschitz-continuous (with constant α), exponentially stabilizing feedback policy as in Assumption 28. We then have:

$$\begin{aligned} & V^*(x) \leq V_\pi(x) = \sum_{t=0}^{\infty} \gamma^t c_\pi\left(f_\pi^t(x)\right) \leq \frac{\lambda}{2}(1+\alpha^2) \sum_{t=0}^{\infty} \gamma^t \left\|f_\pi^t(x)\right\|^2 \\ & \leq \frac{\lambda}{2}(1+\alpha^2) \sum_{t=0}^{\infty} \left\|f_\pi^t(x)\right\|^2 \leq \frac{\lambda}{2} M(1+\alpha^2) \sum_{t=0}^{\infty} \beta^{2t} \|x\|^2 \\ & = \frac{\lambda M(1+\alpha^2)}{2(1-\beta^2)} \|x\|^2. \end{aligned}$$

We then have:

$$\underbrace{\frac{\mu}{2}}_{\underline{\kappa}} \|x\|^2 \leq V^*(x) \leq \underbrace{\frac{\lambda M(1+\alpha^2)}{2(1-\beta^2)}}_{\bar{\kappa}} \|x\|^2,$$

and the statement of the lemma follows. *(ii) Exponential stability under expert (optimal) feedback policy:* We have $V^*(x) = c_{\pi^*}(x) + \gamma V^*\left(f_{\pi^*}(x)\right)$ and $\underline{\kappa}\|x\|^2 \leq V^*(x) \leq \bar{\kappa}\|x\|^2$. It then follows that:

$$\begin{aligned} V^*(f_{\pi^*}(x)) - V^*(x) &\leq -\frac{\mu}{2\gamma}\|x\|^2 + \frac{\bar{\kappa}(1-\gamma)}{\gamma}\|x\|^2 \\ &= -\frac{\bar{\kappa}}{\gamma}\left[\gamma - \left(1 - \frac{\underline{\kappa}}{\bar{\kappa}}\right)\right]\|x\|^2 = -\bar{\kappa}\left(1 - \frac{\gamma'}{\gamma}\right)\|x\|^2. \end{aligned}$$

It follows from the above inequality and the quadratic boundedness of V^* that f_{π^*} is uniformly globally exponentially convergent [92]. In what follows, we obtain an estimate for the upper bound on $\left\|f^t_{\pi^*}(x)\right\|$. From the above inequality and the fact that $V^*(x) \leq \bar{\kappa}\|x\|^2$ we get $V^*(f_{\pi^*}(x)) - V^*(x) \leq -(1-\gamma'/\gamma)V^*(x)$ and $V^*(f_{\pi^*}(x)) \leq \gamma'/\gamma V^*(x)$. It then follows that $V^*(f^t_{\pi^*}(x)) \leq (\gamma'/\gamma)^t V^*(x)$ which implies $\underline{\kappa}\left\|f^t_{\pi^*}(x)\right\|^2 \leq \bar{\kappa}(\gamma'/\gamma)^t\|x\|^2$. Therefore, we get:

$$\left\|f^t_{\pi^*}(x)\right\| \leq \sqrt{\frac{\bar{\kappa}}{\underline{\kappa}}}\left(\sqrt{\frac{\gamma'}{\gamma}}\right)^t \|x\|.$$

5.5.2 Proof of Theorem 34

Let $\gamma' = 1 - \underline{\kappa}^*/\bar{\kappa}^*$. For $x \in B_r(0)$, let $\widehat{x}_t = f^t_{\widehat{\pi}}(x)$ and $x^*_t = f^t_{\pi^*}(x)$. From Lemma 31-(ii), we have:

$$\begin{aligned} &\|f_{\widehat{\pi}}(x)\| \leq \|f_{\pi^*}(x)\| + \|f_{\widehat{\pi}}(x) - f_{\pi^*}(x)\| \\ &\leq \ell_{f_{\pi^*}}\|x\| + \ell^u_f\|\widehat{\pi}(x) - \pi^*(x)\| \leq \ell_{f_{\pi^*}} r + \ell^u_f \varepsilon. \end{aligned}$$

We see that $\|f_{\widehat{\pi}}(x)\| \leq r$ for $\varepsilon \leq \frac{1}{\ell_f^u}\left(1 - \ell_{f_{\pi^*}}\right) r$. It then follows that $\|\widehat{\pi}(\widehat{x}_t) - \pi^*(\widehat{x}_t)\| \leq \varepsilon$ for any $t \in \mathbb{N}$. We then have:

$$\begin{aligned}\|\widehat{x}_t\| &\leq \ell_{f_{\pi^*}}^t \|x\| + \ell_f^u \sum_{\tau=0}^{t-1} \ell_{f_{\pi^*}}^{t-\tau-1} \varepsilon \\ &= \ell_{f_{\pi^*}}^t \|x\| + \ell_f^u \left[\frac{1 - \ell_{f_{\pi^*}}^t}{1 - \ell_{f_{\pi^*}}}\right] \varepsilon.\end{aligned}$$

Furthermore, we have from part (i) that $x_t^* \in B_r(0)$ for any $t \in \mathbb{N}$. Therefore, $B_r(0)$ is invariant under f_{π^*} and $f_{\widehat{\pi}}$, and we immediately obtain the uniform bound $\|\widehat{x}_t - x_t^*\| \leq 2r$. We now have:

$$\begin{aligned}&\|\widehat{x}_t - x_t^*\| = \left\|f(\widehat{x}_{t-1}, \widehat{\pi}(\widehat{x}_{t-1})) - f(x_{t-1}^*, \pi^*(x_{t-1}^*))\right\| \\ &\leq \|f(\widehat{x}_{t-1}, \widehat{\pi}(\widehat{x}_{t-1})) - f(\widehat{x}_{t-1}, \pi^*(\widehat{x}_{t-1}))\| \\ &\quad + \left\|f(\widehat{x}_{t-1}, \pi^*(\widehat{x}_{t-1})) - f(x_{t-1}^*, \pi^*(x_{t-1}^*))\right\| \\ &\leq \left\|f_{\pi^*}(\widehat{x}_{t-1}) - f_{\pi^*}(x_{t-1}^*)\right\| \\ &\quad + \|f(\widehat{x}_{t-1}, \widehat{\pi}(\widehat{x}_{t-1})) - f(\widehat{x}_{t-1}, \pi^*(\widehat{x}_{t-1}))\| \\ &\leq \ell_{f_{\pi^*}} \left\|\widehat{x}_{t-1} - x_{t-1}^*\right\| + \ell_f^u \|\widehat{\pi}(\widehat{x}_{t-1}) - \pi^*(\widehat{x}_{t-1})\| \\ &\leq \ell_{f_{\pi^*}} \left\|\widehat{x}_{t-1} - x_{t-1}^*\right\| + \ell_f^u \varepsilon \\ &\leq \ell_f^u \sum_{\tau=0}^{t-1} \ell_{f_{\pi^*}}^{t-\tau-1} \varepsilon = \ell_f^u \left[\frac{1 - \ell_{f_{\pi^*}}^t}{1 - \ell_{f_{\pi^*}}}\right] \varepsilon.\end{aligned}$$

Now, for the policy $\widehat{\pi}_\delta$, we have $\|\widehat{\pi}_\delta - \pi^*\|_{(B_r(0),\infty)} \leq \|\widehat{\pi}_\delta - \widehat{\pi}\|_{(B_r(0),\infty)} + \|\widehat{\pi} - \pi^*\|_{(B_r(0),\infty)} \leq \alpha\zeta + \varepsilon$, and the earlier analysis now carries through with this bound, and the statement of the theorem follows.

5.5.3 Proof of Lemma 35

From the exponential stability of f_π, for $x, x' \in B_r(0)$, we have $\left\|f_\pi^t(x) - f_\pi^t(x')\right\| \leq \left\|f_\pi^t(x)\right\| + \left\|f_\pi^t(x')\right\| \leq 2M\beta^t r$. Furthermore, let ℓ_{f_π} be the Lipschitz constant of f_π on $B_r(0)$. This implies that $\left\|f_\pi^t(x) - f_\pi^t(x')\right\| \leq \ell_{f_\pi}^t \left\|x - x'\right\|$. We then have $\left\|f_\pi^t(x) - f_\pi^t(x')\right\| \leq \min\left\{\ell_{f_\pi}^t \left\|x - x'\right\|, 2M\beta^t r\right\}$. We can then obtain an $\bar{M}$ such that $\left\|f_\pi^t(x) - f_\pi^t(x')\right\| \leq \bar{M}\beta^t\|x - x'\|$.

5.5.4 Proof of Theorem 37

The following lemma establishes a difference bound for the value function under a Lipschitz feedback policy:

Lemma 41 (*Value function difference bound*) *Let $\pi \in \mathrm{Lip}\left(B_r(0); \mathbb{R}^m\right)$ be a Lipschitz feedback policy such that $\pi(0) = 0$, $\mathrm{lip}(\pi) \leq \alpha$. For the value function $V_\pi(x) = \sum_{t=0}^\infty \gamma^t c_\pi\left(f_\pi^t(x)\right)$ of policy π, the following holds:*

$$\left|V_\pi(x') - V_\pi(x)\right| \leq \Theta\lambda r\sqrt{1+\alpha^2}\left\|x' - x\right\|\left[1 + \frac{\|x' - x\|}{2r}\right],$$

where $\Theta = \sum_{t=0}^\infty \gamma^t\theta_t^2$ and $\theta_t = \bar{M}\beta^t$.

Proof. We first note that $(0,0) \in B_r(0) \times \mathbb{R}^m$ is a strict minimizer of c (by Assumption 29) and since c is differentiable, we have $\nabla c(0,0) = 0$. For any $x \in B_r(0)$:

$$\begin{aligned}\|\nabla c_\pi(x)\| &= \|\nabla c(x, \pi(x)) - \nabla c(0,0)\| \\ &\leq \lambda \left\|(x, \pi(x))\right\| \leq \lambda\sqrt{1+\alpha^2}\|x\|,\end{aligned}$$

since $\|\pi(x)\| = \|\pi(x) - \pi(0)\| \leq \alpha\|x\|$. For any $x, x' \in B_r(0)$, let p be the straight line segment between x and x', such that $p(t) = x + t(x' - x)$ for $t \in [0,1]$. From the λ-

smoothness of c, we have:

$$\begin{aligned}
c_\pi(x') - c_\pi(x) &= \int_0^1 \nabla c_\pi(p(t)) \cdot \dot{p}(t) dt \\
&\leq \int_0^1 \|\nabla c_\pi(p(t))\| \, dt \cdot \|x' - x\| \\
&\leq \lambda\sqrt{1+\alpha^2} \int_0^1 \|p(t)\| dt \cdot \|x' - x\| \\
&\leq \lambda\sqrt{1+\alpha^2} \int_0^1 \|x + t(x' - x)\| dt \cdot \|x' - x\| \\
&\leq \lambda\sqrt{1+\alpha^2} \, \|x\| \left\|x' - x\right\| + \frac{\lambda}{2}\sqrt{1+\alpha^2} \left\|x' - x\right\|^2
\end{aligned}$$

We also have:

$$\begin{aligned}
c_\pi(x) - c_\pi(x') \leq & \lambda\sqrt{1+\alpha^2} \, \|x'\| \left\|x' - x\right\| \\
& + \frac{\lambda}{2}\sqrt{1+\alpha^2} \left\|x' - x\right\|^2,
\end{aligned}$$

and therefore we get:

$$\begin{aligned}
\left|c_\pi(x) - c_\pi(x')\right| \leq & \lambda\sqrt{1+\alpha^2} \, \max\{\|x\|, \|x'\|\} \left\|x' - x\right\| \\
& + \frac{\lambda}{2}\sqrt{1+\alpha^2} \left\|x' - x\right\|^2.
\end{aligned}$$

We now have:

$$\begin{aligned}
V_\pi(x') - V_\pi(x) &= \sum_{t=0}^{\infty} \gamma^t \left[c_\pi\left(f_\pi^t(x')\right) - c_\pi\left(f_\pi^t(x)\right)\right] \\
&\leq \sum_{t=0}^{\infty} \gamma^t \Big[\lambda\sqrt{1+\alpha^2} \, \max\{\|f_\pi^t(x)\|, \|f_\pi^t(x')\|\} \left\|f_\pi^t(x') - f_\pi^t(x)\right\| \\
&\qquad + \frac{\lambda}{2}\sqrt{1+\alpha^2} \left\|f_\pi^t(x') - f_\pi^t(x)\right\|^2\Big] \\
&\leq \left[\sum_{t=0}^{\infty} \gamma^t \theta_t^2\right] \lambda\sqrt{1+\alpha^2} \, \max\{\|x\|, \|x'\|\} \left\|x' - x\right\| \\
&\quad + \left[\sum_{t=0}^{\infty} \gamma^t \theta_t^2\right] \frac{\lambda}{2}\sqrt{1+\alpha^2} \left\|x' - x\right\|^2 \\
&\leq \Theta\lambda\sqrt{1+\alpha^2} \, r \left\|x' - x\right\| + \frac{1}{2}\Theta\lambda\sqrt{1+\alpha^2} \left\|x' - x\right\|^2,
\end{aligned}$$

and the statement of the lemma follows. ■ *(i) Regret:* Let $\pi \in \mathrm{Lip}\,(B_r(0), \mathbb{R}^m)$ be a policy such that $\|\pi - \pi^*\| \leq \varepsilon$ and $\mathrm{lip}(\pi) \leq \alpha$. Since $\varepsilon \leq \left(1 - \sqrt{(\bar{\kappa}^*/\underline{\kappa}^* - 1)/\gamma}\right) r/\ell_f^u$, we get from Theorem 34 that $B_r(0)$ is f_π-invariant. The value function V_π corresponding to π satisfies $V_\pi(x) = c_\pi\big(x\big) + \gamma V_\pi\big(f_\pi(x)\big)$. We then have for any $x \in B_r(0)$:

$$\begin{aligned}
\mathcal{R}(\pi) &= \sup_{x \in B_r(0)} \{V_\pi(x) - V^*(x)\} \\
&= \sup_{x \in B_r(0)} \{c_\pi(x) - c_{\pi^*}(x) + \gamma\,(V_\pi(f_\pi(x)) - V^*(f_{\pi^*}(x)))\} \\
&\leq \sup_{x \in B_r(0)} \{c_\pi(x) - c_{\pi^*}(x) + \gamma\,(V_\pi(f_\pi(x)) - V^*(f_\pi(x))) \\
&\qquad\qquad + \gamma\,(V^*(f_\pi(x)) - V^*(f_{\pi^*}(x)))\} \\
&\leq \sup_{x \in B_r(0)} \{c_\pi(x) - c_{\pi^*}(x) + \gamma\,(V^*(f_\pi(x)) - V^*(f_{\pi^*}(x))) \\
&\qquad\qquad + \gamma \sup_{x \in B_r(0)} \{V_\pi(x) - V^*(x)\}\}.
\end{aligned}$$

It then follows that:

$$\begin{aligned}
\mathcal{R}(\pi) \leq \frac{1}{1-\gamma} \cdot \sup_{x \in B_r(0)} \{&c_\pi(x) - c_{\pi^*}(x) \\
&+ \gamma\,(V^*(f_\pi(x)) - V^*(f_{\pi^*}(x)))\}.
\end{aligned}$$

Furthermore, we have:

$$\begin{aligned}
&\sup_{x \in B_r(0)} \{c_\pi(x) - c_{\pi^*}(x)\} \\
&\quad \leq \lambda \sqrt{1 + |\max\{\alpha, \alpha^*\}|^2}\; r \, \|\pi - \pi^*\|_\infty + \frac{\lambda}{2} \|\pi - \pi^*\|_\infty^2 .
\end{aligned}$$

From Lemma 41, we also have:

$$
\begin{aligned}
&V^*(f_\pi(x)) - V^*(f_{\pi^*}(x)) \\
&\qquad \le \Theta\lambda r\sqrt{1+\alpha^{*2}}\,\|f_\pi(x) - f_{\pi^*}(x)\|\left[1 + \frac{\|f_\pi(x) - f_{\pi^*}(x)\|}{2r}\right] \\
&\qquad \le \Theta\lambda r\sqrt{1+\alpha^{*2}}\;\ell_f^u\,\|\pi - \pi^*\|_\infty\left[1 + \frac{\ell_f^u\,\|\pi - \pi^*\|_\infty}{2r}\right].
\end{aligned}
$$

The statement of the theorem follows from the above two inequalities.

(ii) Robustness: The value function for the policy π satisfies $V_\pi(x) = c_\pi\big(x\big) + \gamma V_\pi\big(f_\pi(x)\big)$. For convenience of notation, we denote $\pi \circ (\mathrm{Id} + \delta)$ by π_δ, i.e., $\pi_\delta(x) = \pi(x+\delta)$. For $\|\delta\|_\infty \le \zeta \in \mathbb{R}$, we have:

$$
\begin{aligned}
\mathcal{S}(\pi) &= \sup_{x\in B_r(0)} \big\{V_{\pi_\delta}\big(x\big) - V_\pi\big(x\big)\big\} \\
&= \sup_{x\in B_r(0)} \big\{c_{\pi_\delta}(x) - c_\pi(x) + \gamma\left(V_{\pi_\delta}\big(f_{\pi_\delta}(x)\big) - V_\pi\big(f_\pi(x)\big)\right)\big\} \\
&\le \sup_{x\in B_r(0)} \{c_{\pi_\delta}(x) - c_\pi(x) + \gamma\left(V_{\pi_\delta}\left(f_{\pi_\delta}(x)\right) - V_\pi\left(f_{\pi_\delta}(x)\right)\right) \\
&\qquad\qquad + \gamma\left(V_\pi\left(f_{\pi_\delta}(x)\right) - V_\pi\left(f_\pi(x)\right)\right)\} \\
&\le \sup_{x\in B_r(0)} \Bigg\{ c_{\pi_\delta}(x) - c_\pi(x) + \gamma\left(V_\pi\left(f_{\pi_\delta}(x)\right) - V_\pi\left(f_\pi(x)\right)\right) \\
&\qquad\qquad + \gamma \sup_{x\in B_r(0)} \left\{V_{\pi_\delta}\left(f_{\pi_\delta}(x)\right) - V_\pi\left(f_{\pi_\delta}(x)\right)\right\}\Bigg\}.
\end{aligned}
$$

It then follows that:

$$
\begin{aligned}
\mathcal{S}(\pi) = \frac{1}{1-\gamma} \cdot \sup_{x\in B_r(0)} &\{c_{\pi_\delta}(x) - c_\pi(x) \\
&+ \gamma\left(V_\pi\left(f_{\pi_\delta}(x)\right) - V_\pi\left(f_\pi(x)\right)\right)\}.
\end{aligned}
$$

Furthermore, we have:

$$\sup_{x\in B_r(0)} \{c_{\pi_\delta}(x) - c_\pi(x)\} \leq \lambda\alpha\sqrt{1+\alpha^2}\ r\zeta + \frac{\lambda}{2}\alpha^2\zeta^2.$$

From Lemma 41, we also have:

$$\begin{aligned} &V_\pi\left(f_{\pi_\delta}(x)\right) - V_\pi\left(f_\pi(x)\right) \\ &\qquad \leq \Theta\lambda r\sqrt{1+\alpha^2}\,\|f_{\pi_\delta}(x) - f_\pi(x)\|\left[1 + \frac{\|f_{\pi_\delta}(x) - f_\pi(x)\|}{2r}\right] \\ &\qquad \leq \Theta\lambda r\sqrt{1+\alpha^2}\ \ell_f^u\alpha\zeta\left[1 + \frac{\ell_f^u\alpha\zeta}{2r}\right]. \end{aligned}$$

The statement of the theorem follows from the above two inequalities.

5.5.5 Proof of Theorem 40

For any $i \in \{1,\ldots,N\}$ and $t \in \{0,\ldots,T-1\}$, we have:

$$\begin{aligned} &|\widehat{\pi}(x) - \pi^*(x)| \\ &= \left|\widehat{\pi}(x) - \widehat{\pi}(x_t^{(i)}) + \widehat{\pi}(x_t^{(i)}) - \pi^*(x_t^{(i)}) + \pi^*(x_t^{(i)}) - \pi^*(x)\right| \\ &\leq (\alpha+\alpha^*)\left|x - x_t^{(i)}\right| + \left|\widehat{\pi}(x_t^{(i)}) - \pi^*(x_t^{(i)})\right|. \end{aligned}$$

In particular, the following holds:

$$\begin{aligned} &|\widehat{\pi}(x) - \pi^*(x)| \\ &\leq \min_{\substack{i\in\{1,\ldots,N\},\\ t\in\{0,\ldots,T-1\}}} \left[(\alpha+\alpha^*)\,|x - x_i| + |\widehat{\pi}(x_i) - \pi^*(x_i)|\right] \\ &\leq (\alpha+\alpha^*) \min_{\substack{i\in\{1,\ldots,N\},\\ t\in\{0,\ldots,T-1\}}} |x - x_i| + \max_{\substack{i\in\{1,\ldots,N\},\\ t\in\{0,\ldots,T-1\}}} |\widehat{\pi}(x_i) - \pi^*(x_i)| \\ &\leq (\alpha+\alpha^*) \min_{\substack{i\in\{1,\ldots,N\},\\ t\in\{0,\ldots,T-1\}}} |x - x_i| + \varepsilon_{\text{train}}. \end{aligned}$$

where

$$\varepsilon_{\text{train}}(\widehat{\pi}) = \max_{\substack{i\in\{1,\ldots,N\},\\ t\in\{0,\ldots,T-1\}}} \left\|\widehat{\pi}(x_t^{(i)}) - \pi^*(x_t^{(i)})\right\|.$$

Therefore, we get:

$$\begin{aligned}&\sup_{x\in B_r(0)} |\widehat{\pi}(x) - \pi^*(x)| \\ &\qquad \leq (\alpha + \alpha^*) \sup_{x\in B_r(0)} \min_{\substack{i\in\{1,\ldots,N\},\\ t\in\{0,\ldots,T-1\}}} \left| x - x_t^{(i)} \right| + \varepsilon_{\text{train}}.\end{aligned}$$

From the previous inequality, we obtain the following probabilistic bound:

$$\mathbb{P}\left[\sup_{x\in B_r(0)} |\widehat{\pi}(x) - \pi^*(x)| > (\alpha + \alpha^*)\delta + \varepsilon_{\text{train}} \right] \leq \mathbb{P}\left[\rho(X_N, r) > \delta\right].$$

5.5.6 Expert policy for the system in subsection 5.4.3

In this subsection, we present more details for the numerical example in subsection 5.4.3. The expert's task is to stabilize the point $r_t = [r_t^x, r_t^y]^\top$ at $[0,0]^\top$ with minimal cost (5.14). Knowing that $r_t^x = x_t + d\cos(\theta_t)$ and $r_t^y = y_t + d\sin(\theta_t)$ and using (5.12), we can describe the dynamics of r_t^x and r_t^y as

$$\underbrace{\begin{bmatrix} r_{t+1}^x \\ r_{t+1}^y \end{bmatrix}}_{r_{t+1}} = \underbrace{\begin{bmatrix} r_t^x \\ r_t^y \end{bmatrix}}_{r_t} + \underbrace{\begin{bmatrix} T_s\cos(\theta_t) & -dT_s\sin(\theta_t) \\ T_s\sin(\theta_t) & dT_s\cos(\theta_t) \end{bmatrix}}_{S(\theta_t)} \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{u_t}, \tag{5.15}$$

Where we assumed that T_s is very small and used the approximation $\sin(T_s\omega_t) \approx T_s\omega_t$ and $\cos(T_s\omega_t) \approx 1$. Let $[v_t, \omega_t]^\top = S(\theta_t)^{-1}[\mu_t^x, \mu_t^x]^\top$, then (5.15) is written as

$$r_{t+1} = r_t + \mu_t, \quad \text{where} \quad \mu_t = [\mu_t^x, \mu_t^y]^\top. \tag{5.16}$$

To stabilize r_t at $[0,0]^\mathsf{T}$, we design $\mu_t = -Kr_t$, where K is a gain matrix that minimizes (5.14), which can be rewritten as

$$\begin{aligned} \min_{\mu \in \mathrm{Lip}(\mathbb{R}^2;\mathbb{R}^2)} \quad & \lim_{T\to\infty} \sum_{t=0}^{T} \gamma^t \left({r_t}^\mathsf{T} Q r_t + {\mu_t}^\mathsf{T} R \mu_t \right), \\ \text{s.t.} \quad & r_{t+1} = r_t + \mu_t, \end{aligned} \tag{5.17}$$

We generate N expert trajectories using (5.13) with $u_t = -S(\theta_t)^{-1}Kr_t$, where K is the LQR gain matrix that minimizes (5.17) with $T_s = 0.01$, $d = 0.15$, $\gamma = 0.8$, $Q = I_2$, $R = 300I_2$, and $\delta = 0$. The generated trajectories are contained in the matrices

$$E = \begin{bmatrix} \mathbf{q}^{(1)} & \dots & \mathbf{q}^{(N)} \end{bmatrix}, \quad U = \begin{bmatrix} \mathbf{u}^{(1)} & \dots & \mathbf{u}^{(N)} \end{bmatrix},$$

with $\mathbf{q}^{(i)} = (q_0^{(i)}, \dots, q_T^{(i)})$ and $\mathbf{u}^{(i)} = (u_0^{(i)}, \dots, u_{T-1}^{(i)})$. Each trajectory is generated with random initial condition, $[x_0^{(i)}, y_0^{(i)}]^\mathsf{T} \in B_2(0)$ and $\theta_0^{(i)} \in B_\pi(0)$ for $i = 1, \dots, N$.

Part III

Behavioral Perspectives for Linear Quadratic Gaussian (LQG) Control: Reformulation, Characterization, and Data-Driven Control

Chapter 6

Behavioral Representation for Optimal Linear Quadratic Gaussian (LQG) Control

In this work, we revisit the Linear Quadratic Gaussian (LQG) optimal control problem from a behavioral perspective. Motivated by the suitability of behavioral models for data-driven control, we begin with a reformulation of the LQG problem in the space of input-output behaviors and obtain a complete characterization of the optimal solutions. In particular, we show that the optimal LQG controller can be expressed as a static behavioral-feedback gain, thereby eliminating the need for dynamic state estimation characteristic of state space methods. The static form of the optimal LQG gain also makes it amenable to its computation by gradient descent, which we investigate via numerical experiments. Furthermore, we highlight the advantage of this approach in the data-driven control setting of learning the optimal LQG controller from expert demonstrations. The results of this chapter are reported in our published paper [67].

6.1 Problem setup and preliminary results

Consider the discrete-time, linear, time-invariant system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t), \\ y(t) &= Cx(t) + v(t), \qquad t \geq 0, \end{aligned} \tag{6.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state, $u(t) \in \mathbb{R}^m$ the control input, $y(t) \in \mathbb{R}^q$ the measured output, $w(t)$ the process noise, and $v(t)$ the measurement noise at time t. We assume that $w(t) \sim \mathcal{N}(0, Q_w)$, with $Q_w \succeq 0$, $v(t) \sim \mathcal{N}(0, R_v)$, with $R_v \succ 0$, and $x(0) \sim \mathcal{N}(0, \Sigma_0)$, with $\Sigma_0 \succeq 0$, are independent of each other at all times. For the system (6.1), the Linear Quadratic Gaussian (LQG) problem asks to find a control input that minimizes the cost

$$\mathcal{J} \triangleq \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\Big(\sum_{t=0}^{T-1} x(t)^{\mathsf{T}} Q_x x(t) + u(t)^{\mathsf{T}} R_u u(t)\Big)\right], \tag{6.2}$$

where $Q_x \succeq 0$ and $R_u \succ 0$ are weighing matrices of appropriate dimension.

Assumption 42 *The pairs (A, B) and $(A, Q_w^{1/2})$ are controllable, and (A, C) and $(A, Q_x^{1/2})$ are observable.* □

As a classic result [113], the optimal control input that solves the LQG problem can be generated by a dynamic controller of the form

$$\begin{aligned} x_c(t+1) &= Ex_c(t) + Fy(t), \\ u(t) &= Gx_c(t) + Hy(t), \end{aligned} \tag{6.3}$$

where $x_c(t)$ denotes the state at time t, and $E \in \mathbb{R}^{n\times n}$, $F \in \mathbb{R}^{n\times q}$, $G \in \mathbb{R}^{m\times n}$, and $H \in \mathbb{R}^{m\times p}$ are the dynamic, input, output and feedthrough matrices of the compensator, respectively. The optimal LQG controller can be conveniently obtained using the separation principle by concatenating the Kalman filter for (6.1) with the (static) controller that solves the Linear Quadratic Regulator problem for (6.1) with weight matrices Q_x and R_u. Specifically, after some manipulation, the optimal input that solves the LQG problem reads as (6.3), we refer the reader to subsection 6.3.2 for the details. In what follows, we will make use of an equivalent representation of the system (6.1). To this aim, let

$$z(t) \triangleq [U(t-1)^\mathsf{T}, Y(t)^\mathsf{T}, W(t-1)^\mathsf{T}, V(t)^\mathsf{T}]^\mathsf{T}, \tag{6.4}$$

where

$$\begin{aligned}
U(t-1) &\triangleq \left[u(t-n)^\mathsf{T}, \cdots, u(t-1)^\mathsf{T}\right]^\mathsf{T}, \\
Y(t) &\triangleq \left[y(t-n)^\mathsf{T}, \cdots, y(t)^\mathsf{T}\right]^\mathsf{T}, \\
W(t-1) &\triangleq \left[w(t-n)^\mathsf{T}, \cdots, w(t-1)^\mathsf{T}\right]^\mathsf{T}, \\
V(t) &\triangleq \left[v(t-n)^\mathsf{T}, \cdots, v(t)^\mathsf{T}\right]^\mathsf{T}.
\end{aligned}$$

We can write an equivalent representation of (6.1) in the behavioral space z as (see subsection 6.3.3 for the derivation):

$$
\begin{gathered}
\underbrace{\left[\begin{array}{c}
u(t-n+1)\\ \vdots\\ u(t-1)\\ u(t)\\ \hline
y(t-n+1)\\ \vdots\\ y(t)\\ y(t+1)\\ \hline
w(t-n+1)\\ \vdots\\ w(t-1)\\ w(t)\\ \hline
v(t-n+1)\\ \vdots\\ v(t)\\ v(t+1)
\end{array}\right]}_{z(t{+}1)}
=
\underbrace{\left[\begin{array}{c|c|c|c}
0\,I\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0\\
\vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots\\
0\,0\,0\cdots I & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0\\
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0\\ \hline
0\,0\,0\cdots 0 & 0\,I\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0\\
\vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots\\
0\,0\,0\cdots 0 & 0\,0\,0\cdots I & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0\\
\mathcal{A}_u & \mathcal{A}_y & \mathcal{A}_w & \mathcal{A}_v\\ \hline
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,I\,0\cdots 0 & 0\,0\,0\cdots 0\\
\vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots\\
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots I & 0\,0\,0\cdots 0\\
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0\\ \hline
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,I\,0\cdots 0\\
\vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\vdots\,\ddots\,\vdots\\
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots I\\
0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0 & 0\,0\,0\cdots 0
\end{array}\right]}_{\mathcal{A}}
\underbrace{\left[\begin{array}{c}
u(t-n)\\ \vdots\\ u(t-2)\\ u(t-1)\\ \hline
y(t-n)\\ \vdots\\ y(t-1)\\ y(t)\\ \hline
w(t-n)\\ \vdots\\ w(t-2)\\ w(t-1)\\ \hline
v(t-n)\\ \vdots\\ v(t-1)\\ v(t)
\end{array}\right]}_{z(t)}
+
\underbrace{\left[\begin{array}{c|c|c}
0 & 0 & 0\\ \vdots & \vdots & \vdots\\ 0 & 0 & 0\\ I & 0 & 0\\ \hline
0 & 0 & 0\\ \vdots & \vdots & \vdots\\ 0 & 0 & 0\\ CB & C & I\\ \hline
0 & 0 & 0\\ \vdots & \vdots & \vdots\\ 0 & 0 & 0\\ 0 & I & 0\\ \hline
0 & 0 & 0\\ \vdots & \vdots & \vdots\\ 0 & 0 & 0\\ 0 & 0 & I
\end{array}\right]}_{\left[\mathcal{B}_u\middle|\mathcal{B}_w\middle|\mathcal{B}_v\right]}
\begin{bmatrix} u(t)\\ w(t)\\ v(t+1)\end{bmatrix},
\\
y_z(t) = \underbrace{\left[\begin{array}{c|c}
I\,0\cdots 0 & 0\,0\cdots 0\\
0\,I\cdots 0 & 0\,0\cdots 0\\
\vdots\,\vdots\,\ddots\,\vdots & \vdots\,\vdots\,\ddots\,\vdots\\
0\,0\cdots I & 0\,0\cdots 0
\end{array}\right]}_{\mathcal{C}} z(t)
\end{gathered}
\tag{6.5}
$$

In fact, given a sequence of control inputs and noise values, the state z contains the system output y over time, and can be used to reconstruct the exact value of the system state x. This also implies that a controller for the system (6.1) can equivalently be designed using the dynamics (6.5). In subsection 6.3.4, we show that any *dynamic* controller for (6.1) can be equivalently represented as a *static* controller for (6.5):

$$u(t) = \mathcal{K} y_z(t), \tag{6.6}$$

where $\mathcal{K} \in \mathbb{R}^{m \times r}$ is the feedback gain and $r = (nm + np + p)$. Next, we reformulate the LQG problem (7.2) for the behavioral dynamics (6.5). The LQG problem (7.2) can be equivalently written in the behavioral space as:

$$\mathcal{J}_z \triangleq \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\Big(\sum_{t=n}^{T-1} z(t)^\mathsf{T} Q_z z(t) + u(t)^\mathsf{T} R_u u(t)\Big)\right], \tag{6.8}$$

subject to (6.5), where Q_z is presented in subsection 6.3.5 along with the derivation of (6.8), and R_u is as in (7.2). The solution to the LQG problem in the behavioral space is given by a static controller in the form of (6.6), which we characterize next.

Theorem 43 ***(Behavioral solution to the LQG problem)*** *Let u^* be the global minimizer of (6.8) subject to (6.5). Then, $u^* = \mathcal{K}^* y_z$ with*

$$\mathcal{K}^* = -\left(R_u + \mathcal{B}_u^\mathsf{T} M \mathcal{B}_u\right)^{-1} \mathcal{B}_u^\mathsf{T} M \mathcal{A} P \mathcal{C}^\mathsf{T} \left(\mathcal{C} P \mathcal{C}^\mathsf{T}\right)^\dagger + \alpha \mathcal{K}_0, \tag{6.9}$$

where $\mathcal{K}_0 \in \mathbb{R}^{d \times r}$ is a matrix whose rows span the left null space of $\mathcal{C}P\mathcal{C}^\mathsf{T}$, and $\alpha \in \mathbb{R}^{m\times d}$ is an arbitrary matrix with $d = nm - n$ and $r = nm + np + p$, and $M \succeq 0$ and $P \succeq 0$ are the unique solutions of the following coupled Riccati equations:

$$\begin{aligned} M &= \mathcal{A}^\mathsf{T} M \mathcal{A} - \mathcal{A}^\mathsf{T} M \mathcal{B}_u S_M \mathcal{B}_u^\mathsf{T} M \mathcal{A} + Q_z \\ &\quad + \left(I - P\mathcal{C}^\mathsf{T} S_P \mathcal{C}\right)^\mathsf{T} \mathcal{A}^\mathsf{T} M \mathcal{B}_u S_M \mathcal{B}_u^\mathsf{T} M \mathcal{A} \left(I - P\mathcal{C}^\mathsf{T} S_P \mathcal{C}\right) \\ P &= \mathcal{A} P \mathcal{A}^\mathsf{T} - \mathcal{A} P \mathcal{C}^\mathsf{T} S_P \mathcal{C} P \mathcal{A}^\mathsf{T} + \Sigma \\ &+ \left(I - M\mathcal{B}_u S_M \mathcal{B}_u^\mathsf{T}\right)^\mathsf{T} \mathcal{A} P \mathcal{C}^\mathsf{T} S_P \mathcal{C} P \mathcal{A}^\mathsf{T} \left(I - M\mathcal{B}_u S_M \mathcal{B}_u^\mathsf{T}\right) \end{aligned}$$

with $S_M \triangleq (R_u + \mathcal{B}_u^\mathsf{T} M \mathcal{B}_u)^{-1}$, $S_P \triangleq (\mathcal{C}P\mathcal{C}^\mathsf{T})^\dagger$, $\Sigma \triangleq \mathcal{B}_w Q_w \mathcal{B}_w^\mathsf{T} + \mathcal{B}_v R_v \mathcal{B}_v^\mathsf{T}$. □

A proof of Theorem 43 is postponed to subsection 6.3.6. The gain $\mathcal{K}$ is not unique since $\mathcal{C}P\mathcal{C}^\mathsf{T}$ is generally not invertible, which stems from the fact that y_z has components that are dependent on each other, which makes the left kernel of $\mathcal{C}P\mathcal{C}^\mathsf{T}$ non-empty. The following result shows that in some cases the gain, $\mathcal{K}$, in (6.9) can be unique, where $\mathcal{C}P\mathcal{C}^\mathsf{T}$ becomes invertible.

Corollary 44 ***(Uniqueness of the behavioral LQG gain $\mathcal{K}^*$)*** *The optimal behavioral LQG gain, $\mathcal{K}^*$, in (6.9) for system (6.5) with single-input (i.e., $m = 1$) is unique.* □

The proof of Corollary 44 follows by noting that $\mathcal{C}P\mathcal{C}^\mathsf{T}$ is full-rank when $m = 1$ (see Lemma 65). Note that, solving the coupled Riccati equations that characterize the LQG gain in Theorem 43 can be challenging. The next result allows us to compute the optimal gain, $\mathcal{K}$, by solving two separate Riccati equations.

Theorem 45 ***(Alternative solution to the behavioral LQG problem)*** *The optimal behavioral LQG gain, $\mathcal{K}^*$, in (6.9) can be written as*

$$\mathcal{K}^* = K_1 + K_2 P_{21} P_{11}^\dagger + \alpha \mathcal{K}_0, \tag{6.10}$$

where $K_1 \in \mathbb{R}^{m\times r}$ and $K_2 \in \mathbb{R}^{m\times q}$, with $r = nm + np + p$, $q = n^2 + np + p$, are computed as follows

$$[K_1, K_2] = K_{LQR} = -\left(R_u + \mathcal{B}_u^\mathsf{T} M_{LQR} \mathcal{B}_u\right)^{-1} \mathcal{B}_u^\mathsf{T} M_{LQR} \mathcal{A}, \tag{6.11}$$

where $M_{LQR} \succeq 0$ is the unique solution of the following Riccati equation

$$\begin{aligned} M_{LQR} =& \mathcal{A}^\mathsf{T} M_{LQR} \mathcal{A} + Q_z \\ & - \mathcal{A}^\mathsf{T} M_{LQR} \mathcal{B}_u \left(R_u + \mathcal{B}_u^\mathsf{T} M_{LQR} \mathcal{B}_u\right)^{-1} \mathcal{B}_u^\mathsf{T} M_{LQR} \mathcal{A}. \end{aligned}$$

The matrices $P_{11} \in \mathbb{R}^{r\times r}$ and $P_{21} \in \mathbb{R}^{q\times r}$ correspond to the $(1,1)$-block and the $(2,1)$-block of matrix P, respectively, $\mathcal{K}_0 \in \mathbb{R}^{d\times r}$ is a matrix whose rows span the left null space of P_{11}, *and $\alpha \in \mathbb{R}^{m \times d}$ is an arbitrary matrix with $d = nm - n$, and P satisfies the Riccati equation*

$$\begin{aligned} P = \mathcal{A}_K P \mathcal{A}_K^{\mathsf{T}} - \mathcal{A}_K P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \mathcal{A}_K^{\mathsf{T}} + \Sigma \\ + \left(\mathcal{A} + \mathcal{B}_u K_{LQR}\right) P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \left(\mathcal{A} + \mathcal{B}_u K_{LQR}\right)^{\mathsf{T}}, \end{aligned} \tag{6.12}$$

where $\mathcal{A}_K \triangleq \mathcal{A} + \mathcal{B}_u K_1 \mathcal{C}$ and $\Sigma \triangleq \mathcal{B}_w Q_w \mathcal{B}_w^{\mathsf{T}} + \mathcal{B}_v R_v \mathcal{B}_v^{\mathsf{T}}$. □

A proof of Theorem 45 is postponed to subsection 6.3.7.

Example 46 ***(LQG controller in the behavioral space)*** *Consider the system (6.1) with $A = 1.1$, $B = 1$, $C = 1$, $Q_w = 0.5$, and $R_v = 0.8$. Also, consider the optimal control problem (7.2) with $Q_x = R_u = 1$. The Kalman and the LQR gains are $K_{\mathrm{kf}} = 0.5474$ and $K_{\mathrm{lqr}} = 0.7034$, respectively, which can be written as (6.3) using (6.21) with $E = 0.1716$, $F = 0.0973$, $G = -0.7034$, and $H = -0.3991$. Using (6.4), we define the behavioral space as $z(t) \triangleq [u(t-1), y(t-1), y(t), w(t-1), v(t-1), v(t)]^{\mathsf{T}}$ for $t \geq 1$. Using Lemma 60, we write the equivalent dynamics of (6.1) in the behavioral space as (6.5) with $\mathcal{A}_u = 0.4977$, $\mathcal{A}_y = \begin{bmatrix} 0.5475 & 0.6023 \end{bmatrix}$, $\mathcal{A}_w = 0.4977$, and $\mathcal{A}_y = \begin{bmatrix} -0.5475 & -0.6023 \end{bmatrix}$. Using Theorem 43, the LQG gain is $\mathcal{K} = [0.1716, 0, -0.3991]$. Fig. 7.1(a) shows the free response of (6.1) and (6.5) with equal initial conditions. Fig. 7.1(b) shows the response of (6.1) and (6.5) to the LQG controllers (6.21) and (6.9), respectively.* □

Example 47 ***(Behavioral LQG controller for single-input system)*** *Consider the*

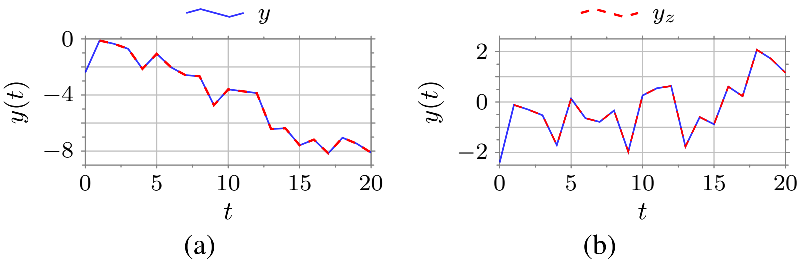


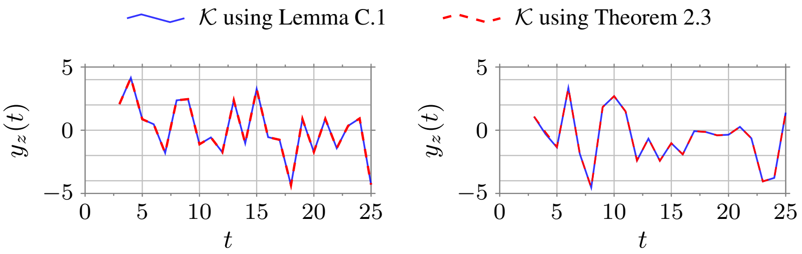
Figure 6.1: This figure shows the free response and the LQG feedback response of (6.1) and (6.5) for the setting defined in Example 47. In both panels, the solid blue line and the dashed red line represent the output of (6.1) and the output of (6.5) that corresponds to $y(t)$, respectively. Panel (a) shows the free response of (6.1) and (6.5), we observe that the response of both systems are equal, which agrees with Lemma 60. Panel (b) shows the feedback response of (6.1) and (6.5) to the LQG controller (6.21) and the behavioral LQG controller in Theorem 43, respectively. We observe that both systems have equal responses, which agrees with Lemma 62 and Theorem 43. Notice that the response of (6.5) starts at time $t = n = 1$ since we have to wait $n = 1$ time steps in order to get the equivalent initial condition for (6.5).

system (6.1) *with*

$$A = \begin{bmatrix} 0.738 & 0.002 & 0.001 \\ 0.694 & 0.875 & 0.766 \\ 0.198 & 1.011 & 0.309 \end{bmatrix}, \quad C = \begin{bmatrix} 0.009 & 0.802 & 0.766 \\ 0.895 & 0.602 & 0.933 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.247 \\ 0.677 \\ 0.757 \end{bmatrix}, \quad Q_x = Q_w = 2I_3, \;\; R_u = 1, \;\; \textit{and } R_v = I_2.$$

The LQG controller can be written in the form of (6.3) *using the classical separation prin-*



(a) (b)

Figure 6.2: This figure shows the output response of (6.5), which corresponds to the first element of $y(t)$, when driven by the behavioral LQG controller for the settings defined in Example 47 and Example 48. In both panels, the solid blue line and the dashed red line represent the output of (6.5) when driven by the LQG gain computed using Lemma 61 and Theorem 45, respectively. Panel (a) corresponds to the setting described in Example 47 where the gains computed using Lemma 61 and Theorem 45 are equal, which agrees with Corollary 44, and hence we observe equal output response. Panel (b) corresponds to the setting described in Example 48 where the optimal LQG gains computed using Lemma 61 and Theorem 45 are different. However, we observe that, although the gains are different, they have equal output response. Notice that the response of (6.5) starts at time $t = n = 3$ since we have to wait $n = 3$ time steps in order to write the behavioral representation in (6.5).

ciple as discussed in subsection 6.3.2 with

$$E = \begin{bmatrix} 0.339 & -0.172 & -0.041 \\ 0.237 & 0.030 & 0.284 \\ -0.371 & 0.101 & -0.195 \end{bmatrix}, \quad F = \begin{bmatrix} -0.241 & 0.181 \\ 0.010 & 0.199 \\ 0.14 & -0.252 \end{bmatrix},$$

$$G = \begin{bmatrix} -0.667 & -1.005 & -0.586 \end{bmatrix}, \quad H = \begin{bmatrix} -0.371 & -0.572 \end{bmatrix}.$$

Using Lemma 60, we can write the equivalent dynamics of (6.1) *in the behavioral space as* (6.5)*. Further, using Lemma 61, the behavioral LQG gain,* $\mathcal{K} = [-.003, .066, .174, 0, 0, .052, .064, .133, -.074$ *Alternatively, using Theorem 45, we obtain a gain equal to the one computed using Lemma 61, where* P_{11} *in* (6.10) *is full-rank and* $\mathcal{K}_{null} = 0$*, which agrees with Corollary 44. Fig. 7.1(a) shows the output response of each of the LQG gains.* □

Example 48 ***(Behavioral LQG controller for MIMO system)*** *We consider the system* (6.1) *with the same parameters as in Example 47, but with an additional input, i.e.,*

$$B = \begin{bmatrix} 0.247 & 0.009 \\ 0.677 & 0.895 \\ 0.757 & 0.802 \end{bmatrix}, \text{ and } R_u = I_2.$$

We write the equivalent behavioral dynamics (6.5) *using Lemma 60. Further, we compute the optimal LQG gain using Lemma 61 and Theorem 45 which are not equal. This is expected since the gain is not unique where P_{11} in* (6.10) *is rank deficient and $\mathcal{K}_{null}$ can be differer from zero. Fig. 7.1(b) shows that output response of both of the LQG gains are equal.* □

6.2 Implications of behavioral representation in numerical methods

In this section, we highlight some implications of our behavioral representation and results. In particular, we provide an analysis of learning the LQG controller from finite expert demonstrations, and an analysis of solving for the behavioral LQG gain via a gradient descent method. First, we present the following Lemma regarding the sparsity of the LQG gain in (6.9), which we use in our subsequent analysis.

Lemma 49 ***(Sparsity of the optimal LQG gain)*** *Consider the LQG gain written in the behavioral space as*

$$u(t) = \begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 & \mathcal{K}_3 \end{bmatrix} \begin{bmatrix} U(t-1) \\ y(t-n) \\ \overline{Y}(t) \end{bmatrix}, \tag{6.13}$$

where $\overline{Y}(t) \triangleq \left[y(t-n+1)^{\mathsf{T}}, \cdots, y(t)^{\mathsf{T}}\right]^{\mathsf{T}}$. *Then* $\mathcal{K}_2 = 0$. □

A proof of Lemma 49 is presented in subsection 6.3.8.

6.2.1 Learning LQG controller from expert demonstrations

Consider the system (6.1), assume that the system is stabilized by an expert that uses optimal LQG controller. We also assume that the system and the noise statistics are unknown. Our objective is to learn the optimal LQG controller from finite expert demonstrations, which are composed of input and output data. In the behavioral representation, this boils down to learning the gain $\mathcal{K}$ of the subspace $u(t) = \mathcal{K}y_z(t)$ for $t \geq n$. Using Lemma 49, we only need to learn $\mathcal{K}_1$ and $\mathcal{K}_3$, which are obtained as $[\mathcal{K}_1\ \mathcal{K}_3] = U_N Y_N^{\dagger} + \mathcal{K}_{\text{null}}$, where

$$\begin{aligned} U_N &\triangleq \begin{bmatrix} u(t) \cdots u(t+k-1) \end{bmatrix}, \\ Y_N &\triangleq \begin{bmatrix} u(t-n) & \cdots & u(t-n+k-1) \\ \vdots & \ddots & \vdots \\ u(t-1) & \cdots & u(t-2+k) \\ y(t-n+1) & \cdots & y(t-n+k) \\ \vdots & \ddots & \vdots \\ y(t) & \cdots & y(t-1+k) \end{bmatrix}, \end{aligned} \tag{6.14}$$

for $t \geq n$, where k is the number of columns, and $\mathcal{K}_{\text{null}}$ is any matrix with appropriate dimension whose rows belong to the left null space of Y_N. Note that $\mathcal{K}_{\text{null}}$ will disappear when multiplied by the feedback $y_z(t)$, i.e., $\mathcal{K}_{\text{null}} y_z(t) = 0$. Therefore, without loss of generality, we set $\mathcal{K}_{\text{null}} = 0$.

Lemma 50 ***(Sufficient number of expert data to compute the optimal LQG gain)*** *Consider input and output expert samples $U = [u(t), \cdots, u(t+N-1)]$ and $Y = [y(t), \cdots, y(t+N-1)]$ generated by LQG controller to stabilize system (6.1), such that U is full-rank. Then, $N = n + nm + np$ expert samples are sufficient to compute the LQG gain $\mathcal{K}$.* □

A proof of Lemma 50 is presented in subsection 6.3.9. We note that the rank condition on the input matrix U in the statement of Lemma 50 is a reasonable assumption owing to the fact that system (6.1) is driven by i.i.d. process noise w and that the measurement noise v is also i.i.d. Furthermore, note that we can learn the dynamic controller matrices E, F, G, and H in (6.3) (up to a similarity transformation) using subspace identification methods for deterministic systems (see [74]) with U and Y treated as the output and input signals to (6.3), respectively. Using [74, Theorem 2], we need at least $N = 2(n+1)(m+p+1) - 1$ expert samples to learn (6.3), which is more than the sufficient number of expert samples to learn $\mathcal{K}$ (Lemma 50).

Example 51 ***(Learning LQG controller from expert data)*** *Consider the system in Example 47 where the system dynamics and the noise statistics are assumed to be unknown. The system is driven by an expert that uses an LQG policy. According to Lemma 50, we collect $N = n + nm + np = 3$ expert input-output samples to form the data matrices*

$$U_N = \begin{bmatrix} -0.2269 & -0.1231 \end{bmatrix}, \quad Y_N = \begin{bmatrix} 1.7878 & -0.2269 \\ 1.3371 & 0.211 \end{bmatrix}.$$

Using the data, we obtain $[\mathcal{K}_1 \; \mathcal{K}_3] = [0.1716 \;\; -0.3991]$ *with* $\mathcal{K}_{null} = 0$*, which matches the LQG gain in Example 47.* □

6.2.2 Gradient descent in the behavioral space

In this section, we use gradient descent to solve for $\mathcal{K}$:

$$\mathcal{K}^{(i+1)} = \mathcal{K}^{(i)} - \alpha^{(i)} \nabla \mathcal{J}_z(\mathcal{K}^{(i)}) \quad \text{for } i = 0, 1, 2, \cdots \tag{6.15}$$

where the index i refers to the iteration number, $\alpha^{(i)}$ is the step size at iteration i, and $\nabla \mathcal{J}_z(\mathcal{K}^{(i)})$ is computed using (6.34). We initialize the gradient descent method with a stabilizing gain $\mathcal{K}^{(0)}$. We determine the step size $\alpha^{(i)}$ by the Armijo rule [12, Chapter 1.3]: initialize $\alpha^{(0)} = 1$, repeat $\alpha^{(i)} = \beta\alpha^{(i)}$ until

$$\mathcal{J}_z(\mathcal{K}^{(i+1)}) \leq \mathcal{J}_z(\mathcal{K}^{(i)}) - \sigma\alpha^{(i)} \left\| \nabla \mathcal{J}_z(\mathcal{K}^{(i)}) \right\|_{\mathrm{F}}^2$$

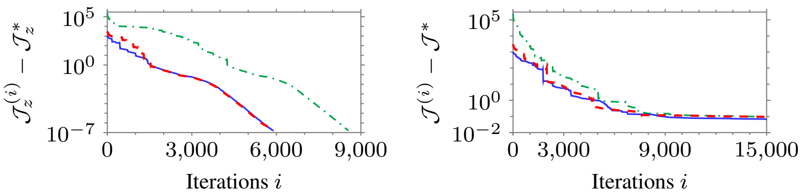
is satisfied, with $\beta, \sigma \in (0, 1)$.

Example 52 ***(Gradient descent)*** *We consider the example in [28] discretized with sampling time* $T_s = 0.4$*,*

$$A = \begin{bmatrix} 1.4918 & 0.5967 \\ 0 & 1.4918 \end{bmatrix}, \; B = \begin{bmatrix} 0.1049 \\ 0.4918 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$$Q_w = \begin{bmatrix} 4.6477 & 3.7575 \\ 3.7575 & 3.0639 \end{bmatrix}, \quad Q_x = \begin{bmatrix} 3.0639 & 3.7575 \\ 3.7575 & 4.6477 \end{bmatrix},$$

$R_v = 2.5$ *and* $R_u = 0.5966$*. The LQG gain from Theorem 43 is* $\mathcal{K} = [-0.0366, -0.103, 0, 5.8461, -4.7434]$*. Using Lemma 49, we only need to do the search over* $\mathcal{K}_1$ *and* $\mathcal{K}_3$ *since* $\mathcal{K}_2 = 0$*. We use gradient descent in (6.15) to solve for the LQG gain. We choose a stabilizing initial gain* $\mathcal{K}^{(0)}$



(a) (b)

Figure 6.3: This figure shows the convergence performance (measured by the sub-optimality gap) of the gradient descent applied to the system in Example 52. The solid blue line, dashed red line and the dash-dotted green line represent different initial conditions, respectively. Panels (a) and (b) show the convergence performance of the gradient descent over $\mathcal{K}$ and the gradient descent over the controller matrices E, F, G and H, respectively.

that randomly place the closed-loop eigenvalues within $[0.45, 0.92]$*. We use the Armijo rule to compute the step size with* $\alpha^{(0)} = 1$*,* $\beta = 0.8$*, and* $\sigma = 0.7$*. We set the stopping criteria to be when the gradient vanishes or when the maximum number of iterations is reached (in this example we set it to* 15000 *iterations). For numerical comparison, we use gradient descent to solve for the optimal LQG dynamic controller in the form of* (6.3) *as in [111], where we optimize the LQG cost* (7.2) *and apply the gradient search over the control parameters* E*,* F*,* G*, and* H*.*[1] *Fig. 6.3 shows the convergence performance of the gradient descent for different initial conditions. We observe that the gradient descent over* $\mathcal{K}$ *in Fig. 6.3(a) converges to* $\mathcal{K}^* = [-0.0366, -0.1030, 0, 5.8460, -4.7434]$ *before reaching the maximum number iterations for different initial conditions. Starting from initial conditions equivalent to the ones in Fig. 6.3(a), the gradient descent over the controller matrices* E*,* F*,* G *and* H *in Fig. 6.3(b) did not converge within* 15000 *iterations.* □

[1] In [111], $H = 0$ since it is assumed that the control input $u(t)$ at time t depends on the history $\{u(0), \cdots, u(t-1), y(0), \cdots, y(t-1)\}$. In this paper, $u(t)$ depends also on $y(t)$, therefore H is nonzero (see subsection 6.3.2). We computed the gradient of $\mathcal{J}$ w.r.t. the controller matrices E, F, G and H as in [111] adapted to the case where H is nonzero. We have not included the derivations in this paper due to space constraint.

6.3 Technical Lemmas and proofs of the main results

6.3.1 Linear Algebra facts and technical lemmas

Fact 53 *([11, **Fact 8.7.5**]) Let $A, B \in \mathbb{R}^{n\times n}$, assume that $A, B \succeq 0$. Then,*

$$\mathcal{R}(A{+}B){=}\mathcal{R}(A){+}\mathcal{R}(B), \quad \mathcal{N}(A{+}B){=}\mathcal{N}(A)\cap\mathcal{N}(B).$$

□

Fact 54 *([11, **Fact 2.10.10**]) Let $A, B \in \mathbb{R}^{n\times m}$, and let $\alpha \in \mathbb{R}$ be non-zero. Then,*

$$\begin{aligned}\mathcal{N}(A)\cap\mathcal{N}(B) &= \mathcal{N}(A)\cap\mathcal{N}(A+\alpha B)\\ &= \mathcal{N}(\alpha A+B)\cap\mathcal{N}(B).\end{aligned}$$

□

Fact 55 *([11, **Fact 7.4.10**]) Let $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times l}$, $C \in \mathbb{R}^{l\times k}$, and $D \in \mathbb{R}^{k\times n}$. Then,*

$$\mathsf{tr}\left[ABCD\right] = (vec(A))^{\mathsf{T}}\left(B\otimes D^{\mathsf{T}}\right)\left(vec\left(C^{\mathsf{T}}\right)\right).$$

□

Lemma 56 *([11, **Lemma 8.2.1**]) Let $A \triangleq \begin{bmatrix} A_{11} & A_{12}\\ A_{12}^{\mathsf{T}} & A_{22}\end{bmatrix} \succeq 0$ with $A_{11} \in \mathbb{R}^{n\times n}$, $A_{22} \in \mathbb{R}^{m\times m}$, and $A_{12} \in \mathbb{R}^{n\times m}$. Then,*

$$A_{12} = A_{11}A_{11}^{\dagger}A_{12} = A_{12}A_{22}A_{22}^{\dagger}.$$

□

Lemma 57 ***(Property of the solution to Lyapunov equation)*** *Let A, B, Q be matrices of appropriate dimensions with $\rho(A) < 1$. Let Y satisfy $Y = AYA^{\mathsf{T}} + Q$. Then, $\mathsf{tr}\,[BY] = \mathsf{tr}\left[Q^{\mathsf{T}}M\right]$, where M satisfies $M = A^{\mathsf{T}}MA + B^{\mathsf{T}}$.* □

Lemma 58 ***(Range space of the solution to Lyapunov equation)*** *Let A, Q be square matrices of appropriate dimensions with $\rho(A) < 1$ and $Q \succeq 0$. Let X satisfy $X = AXA^{\mathsf{T}} + Q$. Then, for $i \geq 1$,*

$$\mathcal{R}\left(A^i X\right) \subseteq \mathcal{R}\left(X\right), \quad \textit{and} \quad \mathcal{N}\left(X\right) \subseteq \mathcal{N}\left(X\left(A^i\right)^{\mathsf{T}}\right).$$

Proof. *From Fact 53, we have*

$$\mathcal{R}\left(X\right) = \mathcal{R}\left(Q\right) + \mathcal{R}\left(AXA^{\mathsf{T}}\right) = \mathcal{R}\left(Q\right) + \mathcal{R}\left(AX\right).$$

The above equality implies that $\mathcal{R}\left(AX\right) \subseteq \mathcal{R}\left(X\right)$. Then, the first claim follows from the fact that $\mathcal{R}\left(A^i\right) \subseteq \mathcal{R}\left(A\right)$ for $i \geq 1$. The second claim follows from the fact that the range space and the null space of any matrix are orthogonal complements. ■

6.3.2 Optimal LQG controller

The LQG controller that minimizes (7.2) can be written as

$$\begin{aligned} \hat{x}(t+1) &= (I_n - K_{\text{kf}}C)(A - BK_{\text{LQR}})\hat{x}(t) + K_{\text{kf}}y(t+1), \\ u(t) &= -K_{\text{LQR}}\hat{x}(t), \end{aligned} \tag{6.16}$$

where K_{kf} and K_{LQR} are the Kalman and LQR gains, respectively. To write the controller (6.16) in the form of (6.3), we need the following lemma.

Lemma 59 ***(Equivalent compensator forms)*** *Consider the compensator (6.3) and a compensator of the form:*

$$\begin{aligned} \xi_c(t+1) &= \overline{E}\xi_c(t) + \overline{F}y(t+1), \\ u(t) &= \overline{G}\xi_c(t), \end{aligned} \tag{6.17}$$

with $\xi_c \in \mathbb{R}^n$ denoting the state, and $\overline{E} \in \mathbb{R}^{n\times n}$, $\overline{F} \in \mathbb{R}^{n\times q}$, and $\overline{G} \in \mathbb{R}^{m\times n}$ denoting the dynamic, input, and output matrices of the compensator, respectively. Let $x_c(0) = \xi_c(0)$ and $y(0) = 0$, then, the compensators (6.3) and (6.17) output the same $u(t)$ given the same input $y(t)$ if:

$$E = \overline{E}, \quad F = \overline{E}\overline{F}, \quad G = \overline{G}, \quad H = \overline{G}\overline{F}. \tag{6.18}$$

Proof. Using (6.3) with $y(0) = 0$, we can write

$$u(t) = GE^t x_c(0) + \begin{bmatrix} GE^{t-2}F & \cdots & GF & H \end{bmatrix} y, \tag{6.19}$$

where $y = [y(1)^\mathsf{T}, \cdots, y(t)^\mathsf{T}]^\mathsf{T}$. Using (6.17), we can write

$$u(t) = \overline{G}\overline{E}^t \xi_c(0) + \begin{bmatrix} \overline{G}\overline{E}^{t-1}\overline{F} & \cdots & \overline{G}\overline{F} \end{bmatrix} y. \tag{6.20}$$

Under the same y, (6.19) is equal to (6.20) for $E = \overline{E}$, $F = \overline{E}\overline{F}$, $G = \overline{G}$, and $H = \overline{G}\overline{F}$. ■

Using Lemma 59 and (6.16), we can write the LQG controller in the form of (6.3) with

$$\begin{aligned} E &= (I_n - K_{\text{kf}}C)(A - BK_{\text{LQR}}), \\ F &= (I_n - K_{\text{kf}}C)(A - BK_{\text{LQR}})K_{\text{kf}}, \\ G &= -K_{\text{LQR}}, \\ H &= -K_{\text{LQR}}K_{\text{kf}}. \end{aligned} \tag{6.21}$$

6.3.3 System representation in the behavioral space

The following Lemma provides an equivalent representation of (6.1) in the behavioral space, z, which is written in (6.5).

Lemma 60 ***(Equivalent dynamics)*** *Let z be as in (6.4). Then,*

$$z(t+1) = \mathcal{A}z(t) + \mathcal{B}_u u(t) + \mathcal{B}_w w(t) + \mathcal{B}_v v(t+1),$$

where $\mathcal{A}$, $\mathcal{B}_u$, $\mathcal{B}_w$, and $\mathcal{B}_v$ are as in (6.5), and

$$\mathcal{A}_u \triangleq \mathcal{F}_2 - CA^{n+1}\mathcal{O}^{\dagger}\mathcal{F}_1, \qquad \mathcal{A}_y \triangleq CA^{n+1}\mathcal{O}^{\dagger},$$

$$\mathcal{A}_w \triangleq \mathcal{F}_4 - CA^{n+1}\mathcal{O}^{\dagger}\mathcal{F}_3, \qquad \mathcal{A}_v \triangleq -CA^{n+1}\mathcal{O}^{\dagger},$$

$$\mathcal{O} \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^n \end{bmatrix}, \quad \mathcal{F}_1 \triangleq \begin{bmatrix} 0 & \cdots & 0 \\ CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ CA^{n-1}B & \cdots & CB \end{bmatrix},$$

$$\mathcal{F}_2 \triangleq \begin{bmatrix} CA^nB & \cdots & CAB \end{bmatrix},$$

and the matrices $\mathcal{F}_3$ and $\mathcal{F}_4$ are obtained by replacing B with I in $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively.

Proof. *We can write the evolution of $y(t+1)$ as*

$$\begin{aligned} y(t+1) =& CA^{n+1}x(t-n) + \mathcal{F}_2U(t-1) + \mathcal{F}_4W(t-1) \\ &+ CBu(t) + Cw(t) + v(t+1), \end{aligned} \tag{6.22}$$

where, $\mathcal{F}_2$ and $\mathcal{F}_4$ are as in Lemma 60, and $U(t-1)$ and $W(t-1)$ are as in (6.4). Also, we can write $Y(t)$ in (6.4) in terms of $U(t-1)$, $W(t-1)$, and $V(t)$:

$$Y(t) = \mathcal{O}x(t-n) + \mathcal{F}_1 U(t-1) + \mathcal{F}_3 W(t-1) + V(t), \tag{6.23}$$

where $\mathcal{O}$, $\mathcal{F}_1$, and $\mathcal{F}_3$ are same as in Lemma 60 and $V(t)$ is as in (6.4). Then using (6.23), we substitute $x(t-n)$ into (6.22)

$$y(t+1) = \begin{bmatrix} \mathcal{A}_u & \mathcal{A}_y & \mathcal{A}_w & \mathcal{A}_v \end{bmatrix} \underbrace{\begin{bmatrix} U(t-1) \\ Y(t) \\ W(t-1) \\ V(t) \end{bmatrix}}_{z(t)} + CBu(t)$$

$$+ Cw(t) + v(t+1),$$

where $\mathcal{A}_u$, $\mathcal{A}_y$, $\mathcal{A}_w$, and $\mathcal{A}_v$ are as in Lemma 60. ■

6.3.4 From dynamic to static controller

Lemma 61 ***(From dynamic to static controllers)*** *Let the control input u be the output of the dynamic controller (6.3). Then, equivalently,*

$$u(t) = \underbrace{\begin{bmatrix} GE^n \mathcal{T}_1^{\dagger} & \mathcal{T}_2 - GE^n \mathcal{T}_1^{\dagger} \mathcal{M} \end{bmatrix}}_{\mathcal{K}} y_z(t), \tag{6.24}$$

where z is as in (6.4) and follows the dynamics (6.5), and

$$\mathcal{T}_1 \triangleq \begin{bmatrix} G \\ GE \\ \vdots \\ GE^{n-1} \end{bmatrix}, \quad \mathcal{T}_2 \triangleq \begin{bmatrix} GE^{n-1}F & \cdots & GF & H \end{bmatrix},$$

$$\mathcal{M} \triangleq \begin{bmatrix} H & 0 & 0 & \cdots & 0 & 0 \\ GF & H & 0 & \cdots & 0 & 0 \\ GEF & GF & H & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ GE^{n-2}F & GE^{n-3}F & \cdots & \cdots & H & 0 \end{bmatrix}.$$

Proof. Using (6.3), we can write

$$u(t) = GE^n x_c(t-n) + \mathcal{T}_2 Y(t), \tag{6.25}$$

where $\mathcal{T}_2$ and $Y(t)$ are as in Lemma 61 and (6.4), respectively. Further, we can write $U(t-1)$ in (6.4) as

$$U(t-1) = \mathcal{T}_1 x_c(t-n) + \mathcal{M} Y(t), \tag{6.26}$$

where $\mathcal{T}_1$ and $\mathcal{M}$ are as in Lemma 61. Using (6.26) we substitute $x_c(t-n)$ in (6.25) to get

$$u(t) = \begin{bmatrix} GE^n \mathcal{T}_1^\dagger & \mathcal{T}_2 - GE^n \mathcal{T}_1^\dagger \mathcal{M} \end{bmatrix} \underbrace{\begin{bmatrix} U(t-1) \\ Y(t) \end{bmatrix}}_{y_z}.$$

■

6.3.5 LQG problem in the behavioral space

Lemma 62 ***(LQG problem in the behavioral space)*** *The input u^* is the minimizer of (7.2) subject to (6.1) if and only if it is the minimizer of*

$$\mathcal{J}_z \triangleq \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\Big(\sum_{t=0}^{T-1} z(t)^{\mathsf{T}} Q_z z(t) + u(t)^{\mathsf{T}} R_u u(t)\Big)\right] \tag{6.27}$$

subject to (6.5), where $Q_z = \mathcal{H}^{\mathsf{T}} Q_x \mathcal{H}$ and

$$\mathcal{H} \triangleq \begin{bmatrix} \mathcal{G}_1 - A^n \mathcal{O}^\dagger \mathcal{F}_1 & A^n \mathcal{O}^\dagger & \mathcal{G}_2 - A^n \mathcal{O}^\dagger \mathcal{F}_3 & -A^n \mathcal{O}^\dagger \end{bmatrix},$$
$$\mathcal{G}_1 \triangleq \begin{bmatrix} A^{n-1}B & \dots & B \end{bmatrix}, \quad \mathcal{G}_2 \triangleq \begin{bmatrix} A^{n-1} & \dots & I_n \end{bmatrix}.$$

Proof. We begin by proving that the costs in (7.2) and (6.27) are equivalent. We can express $x(t)$ for $t \geq n$ as

$$x(t) = A^n x(t-n) + \mathcal{G}_1 U(t-1) + \mathcal{G}_2 W(t-1), \tag{6.28}$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ are as in Lemma 62, and $U(t-1)$ and $W(t-1)$ are as in (6.4). Using (6.23), we can substitute $x(t-n)$ in terms of $U(t-1)$, $Y(t)$, $W(t-1)$, and $V(t)$ into (6.28) to get $x(t) = \mathcal{H}z(t)$, where $\mathcal{H}$ is as in Lemma 62. Substituting $x(t) = \mathcal{H}z(t)$ into the cost (7.2) yields the cost (6.27). Further, Lemma 60 shows that the systems (6.1) and (6.5) are equivalent. Therefore, the minimizer of (7.2) subject to (6.1) is the minimizer of (6.27) subject to (6.5). ■

6.3.6 Proof of Theorem 43

For the proof of Theorem 43, we need the following technical results from the literature.

Lemma 63 ***(Steady-state cost)*** *For a controller $u(t) = \mathcal{K}y_z(t)$ with stabilizing gain $\mathcal{K}$, the cost (6.8) at steady-state is written as*

$$\mathcal{J}_z(\mathcal{K}) = \mathsf{tr}\left[Q_{\mathcal{K}}P\right], \tag{6.29}$$

where $Q_{\mathcal{K}} \triangleq Q_z + \mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}}R_u\mathcal{K}\mathcal{C}$, and $P \succeq 0$ is the unique solution of the following Lyapunov equation

$$P = \mathcal{A}_cP\mathcal{A}_c^{\mathsf{T}} + \Sigma. \tag{6.30}$$

with $\mathcal{A}_c \triangleq \mathcal{A} + \mathcal{B}_u\mathcal{K}\mathcal{C}$ and $\Sigma \triangleq \mathcal{B}_wQ_w\mathcal{B}_w^{\mathsf{T}} + \mathcal{B}_vR_v\mathcal{B}_v^{\mathsf{T}}$.

Proof. Since $u(t) = \mathcal{K}y_z(t)$ is stabilizing, the closed-loop matrix $\mathcal{A}_c = \mathcal{A} + \mathcal{B}_u\mathcal{K}\mathcal{C}$ is stable. We can write

$$\begin{aligned}\mathbb{E}\left[z(t)z(t)^{\mathsf{T}}\right] =& \mathcal{A}_c\mathbb{E}\left[z(t-1)z(t-1)^{\mathsf{T}}\right]\mathcal{A}_c^{\mathsf{T}} \\ &+ \underbrace{\mathcal{B}_wQ_w\mathcal{B}_w^{\mathsf{T}} + \mathcal{B}_vR_v\mathcal{B}_v^{\mathsf{T}}}_{\Sigma},\end{aligned}$$

where we have used the fact that $z(t-1)$, $w(t-1)$, and $v(t)$ are uncorrelated, and $\mathbb{E}\left[w(t-1)w(t-1)^{\mathsf{T}}\right] = Q_w$ and $\mathbb{E}\left[v(t)v(t)^{\mathsf{T}}\right] = R_v$. Since $\mathcal{A}_c$ is stable, $\lim_{t\to\infty}\mathbb{E}\left[z(t)z(t)^{\mathsf{T}}\right]$ converges to a finite value, and at steady state we have $P \triangleq \lim_{t\to\infty}\mathbb{E}\left[z(t)z(t)^{\mathsf{T}}\right] = \lim_{t\to\infty}\mathbb{E}\left[z(t-1)z(t-1)^{\mathsf{T}}\right]$, where P satisfies (6.30). The cost (6.8) is written as

$$\begin{aligned}\mathcal{J}_z &\triangleq \lim_{T\to\infty}\mathbb{E}\left[\frac{1}{T}\left(\sum_{t=0}^{T-1} z(t)^{\mathsf{T}}\left(Q_z + \mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}}R_u\mathcal{K}\mathcal{C}\right)z(t)\right)\right] \\ &= \lim_{t\to\infty}\mathbb{E}\left[\mathsf{tr}\left[z(t)^{\mathsf{T}}Q_{\mathcal{K}}z(t)\right]\right] = \mathsf{tr}\left[Q_{\mathcal{K}}\lim_{t\to\infty}\mathbb{E}\left[z(t)z(t)^{\mathsf{T}}\right]\right] \\ &= \mathsf{tr}\left[Q_{\mathcal{K}}P\right],\end{aligned}$$

where $Q_{\mathcal{K}} \triangleq Q_z + \mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}}R_u\mathcal{K}\mathcal{C}$. The proof is complete. ■

Assumption 64 ***(Observability of the dynamic LQG controller)*** *The pair (E, G) of the optimal LQG controller written in the form (6.3) is observable.* □

Lemma 65 ***(Rank of $\mathcal{C}P\mathcal{C}^\mathsf{T}$)*** *Let $\mathcal{K}^*$ be the minimizer of $\mathcal{J}_z$ in (6.29). Then,*

$$\text{Rank}\left[\mathcal{C}P\mathcal{C}^\mathsf{T}\right] = n + (n+1)p,$$

where $P \succeq 0$ is the solution of the Lyapunov equation in (6.30) that corresponds to $\mathcal{K}^*$.

Proof. *We begin by noting that the solution of (6.30) that corresponds to $\mathcal{K}^*$ can be written as $P = \lim\limits_{t\to\infty} \mathbb{E}\left[z(t)z(t)^\mathsf{T}\right]$, where $z(t)$ is the state of (6.5) at time $t \geq n$ when driven by the optimal controller $u^*(t) = \mathcal{K}^* y_z(t)$. Further, the input sequence $u^*(t)$ can also be generated by the dynamic controller defined in (6.21), which is written in the form of (6.3) (see Lemma 61 and Lemma 62). From (6.26), we can write*

$$\underbrace{\begin{bmatrix} \mathbf{U} \\ \mathbf{Y} \end{bmatrix}}_{\mathbf{Y_z}} = \underbrace{\begin{bmatrix} \mathcal{T}_1 & \mathcal{M} \\ 0 & I \end{bmatrix}}_{\mathcal{H}} \begin{bmatrix} \mathbf{X_c} \\ \mathbf{Y} \end{bmatrix}, \tag{6.31}$$

where $\mathbf{U} \triangleq \begin{bmatrix} U(n-1) & \cdots & U(T-1) \end{bmatrix}$, $\mathbf{Y} \triangleq \begin{bmatrix} Y(n) & \cdots & Y(T) \end{bmatrix}$, and $\mathbf{X_c} \triangleq \begin{bmatrix} x_c(0) & \cdots & x_c(T-n) \end{bmatrix}$. Under Assumption 64, the matrix $\mathcal{H}$ is full-column rank, with $\text{Rank}\left[\mathcal{H}\right] = n + (n+1)p$. Further, since $\mathbf{Y}$ has Gaussian measurement noise, we have $\text{Rank}\left(\begin{bmatrix} \mathbf{X_c} \\ \mathbf{Y} \end{bmatrix}\right) = n+(n+1)p$ using [102, Corollary 2-(ii)], which is full-row rank. Then, we have $\text{Rank}\left(\mathbf{Y_z}\right) = n+(n+1)p$. Finally, $\text{Rank}\left[\mathcal{C}P\mathcal{C}^\mathsf{T}\right] = n + (n+1)p$ since at steady state, for $t \geq n$, we have $\mathcal{C}P\mathcal{C}^\mathsf{T} = \mathbb{E}\left[y_z(t)y_z(t)^\mathsf{T}\right] = \lim\limits_{T\to\infty} \frac{1}{T-n+1}\mathbf{Y_z}\mathbf{Y_z}^\mathsf{T}$. ■

Proof of Theorem 43: First-order necessary conditions: Using Lemma 63, we can write the cost (6.8) at steady-state as (6.29). Next, we compute the derivative of $\mathcal{J}_z(\mathcal{K})$ with respect to the variable $\mathcal{K}$. Taking the differential of (6.30) with respect to the variable $\mathcal{K}$, we get

$$\begin{aligned} dP &= \mathcal{A}_c dP \mathcal{A}_c^{\mathsf{T}} + d\mathcal{A}_c P \mathcal{A}_c^{\mathsf{T}} + \mathcal{A}_c P d\mathcal{A}_c^{\mathsf{T}} \triangleq \mathcal{A}_c dP \mathcal{A}_c^{\mathsf{T}} + X \\ \implies \mathsf{tr}\left[Q_{\mathcal{K}} dP\right] &\overset{\text{(a)}}{=} \mathsf{tr}\left[XM\right] \overset{\text{(b)}}{=} 2\mathsf{tr}\left[\mathcal{C}P\mathcal{A}_c^{\mathsf{T}} M \mathcal{B}_u d\mathcal{K}\right], \end{aligned} \tag{6.32}$$

where $M \succeq 0$ satisfies $M = \mathcal{A}_c^{\mathsf{T}} M \mathcal{A}_c + Q_{\mathcal{K}}$, (a) follows from Lemma 57, and (b) follows from $\mathsf{tr}\left[d\mathcal{A}_c P \mathcal{A}_c^{\mathsf{T}} M\right] = \mathsf{tr}\left[(d\mathcal{A}_c P \mathcal{A}_c^{\mathsf{T}} M)^{\mathsf{T}}\right]$ and using the trace cyclic property. Taking the differential of $Q_{\mathcal{K}}$, we get

$$\begin{aligned} dQ_{\mathcal{K}} &= \mathcal{C}^{\mathsf{T}} d\mathcal{K}^{\mathsf{T}} R_u \mathcal{K}\mathcal{C} + \mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}} R_u d\mathcal{K}\mathcal{C} \\ \implies \mathsf{tr}\left[dQ_{\mathcal{K}} P\right] &\overset{\text{(c)}}{=} 2\mathsf{tr}\left[\mathcal{C}P\mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}} R_u d\mathcal{K}\right], \end{aligned} \tag{6.33}$$

where (c) follows similarly as (b). For notational convenience, we denote $\mathcal{J}_z(\mathcal{K})$ by $\mathcal{J}_z$. Taking the differential of $\mathcal{J}_z$ in (6.29), we get,

$$\begin{aligned} d\mathcal{J}_z &= d\mathsf{tr}\left[Q_{\mathcal{K}} P\right] = \mathsf{tr}\left[dQ_{\mathcal{K}} P\right] + \mathsf{tr}\left[Q_{\mathcal{K}} dP\right] \\ &= 2\mathsf{tr}\left[\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}} R_u + \mathcal{C}P\mathcal{A}_c^{\mathsf{T}} M \mathcal{B}_u\right) d\mathcal{K}\right] \\ \implies \frac{d\mathcal{J}_z}{d\mathcal{K}} &= 2\left(R_u \mathcal{K}\mathcal{C}P\mathcal{C}^{\mathsf{T}} + \mathcal{B}_u^{\mathsf{T}} M \mathcal{A}_c P \mathcal{C}^{\mathsf{T}}\right) \\ &= 2\left(R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u\right) \mathcal{K}\mathcal{C}P\mathcal{C}^{\mathsf{T}} + 2\mathcal{B}_u^{\mathsf{T}} M \mathcal{A} P \mathcal{C}^{\mathsf{T}}. \end{aligned} \tag{6.34}$$

The stationary optimality condition implies $\frac{d\mathcal{J}_z}{d\mathcal{K}} = 0$, we get

$$\mathcal{K}_{\mathrm{s}} = \underbrace{-\left(R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u\right)^{-1} \mathcal{B}^{\mathsf{T}} M \mathcal{A} P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right)^{\dagger}}_{\mathcal{K}^*} + \alpha\mathcal{K}_0, \tag{6.35}$$

where we have used the right pseudo inverse of $\mathcal{C}P\mathcal{C}^{\mathsf{T}}$ since it is rank deficient, $\mathcal{K}_0 \in \mathbb{R}^{d\times r}$ is a matrix whose rows span the left null space of $\mathcal{C}P\mathcal{C}^{\mathsf{T}}$, and $\alpha \in \mathbb{R}^{m\times d}$ is an arbitrary matrix, with $d = nm - n$ and $r = nm + np + p$ (see Lemma 65). Next we derive the Riccati equations of M and P. Let $S_M \triangleq (R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u)^{-1}$ and $S_P \triangleq (\mathcal{C}P\mathcal{C}^{\mathsf{T}})^{\dagger}$. Substituting the expression of $\mathcal{K}^*$ in (6.35) into (6.30), we get

$$\begin{aligned}
P =\, & \mathcal{A}P\mathcal{A}^\top - \mathcal{A}P\mathcal{C}^\top S_p\mathcal{C}P\mathcal{A}^\top M\mathcal{B}_u S_M\mathcal{B}_u^\top \\
& - \mathcal{B}_u S_M\mathcal{B}_u^\top M\mathcal{A}P\mathcal{C}^\top S_P\mathcal{C}P\mathcal{A}^\top + \mathcal{B}_w Q_w\mathcal{B}_w^\top + \mathcal{B}_v R_v\mathcal{B}_v^\top \\
& + \mathcal{B}_u S_M\mathcal{B}_u^\top M\mathcal{A}P\mathcal{C}^\top \underbrace{S_P\left(\mathcal{C}P\mathcal{C}^\top\right)S_P}_{\overset{\text{(d)}}{=}S_p}\mathcal{C}P\mathcal{A}^\top M\mathcal{B}_u S_M\mathcal{B}_u^\top \\
\overset{\text{(e)}}{=} & \mathcal{A}P\mathcal{A}^\top - \mathcal{A}P\mathcal{C}^\top S_p\mathcal{C}P\mathcal{A}^\top M\mathcal{B}_u S_M\mathcal{B}_u^\top \\
& \quad - \mathcal{B}_u S_M\mathcal{B}_u^\top M\mathcal{A}P\mathcal{C}^\top S_P\mathcal{C}P\mathcal{A}^\top + \mathcal{B}_w Q_w\mathcal{B}_w^\top + \mathcal{B}_v R_v\mathcal{B}_v^\top \\
& \quad + \mathcal{B}_u S_M\mathcal{B}_u^\top M\mathcal{A}P\mathcal{C}^\top S_P\mathcal{C}P\mathcal{A}^\top M\mathcal{B}_u S_M\mathcal{B}_u^\top \\
& \quad + \mathcal{A}P\mathcal{C}^\top S_p\mathcal{C}P\mathcal{A}^\top - \mathcal{A}P\mathcal{C}^\top S_p\mathcal{C}P\mathcal{A}^\top \\
= & \mathcal{A}P\mathcal{A}^\top - \mathcal{A}P\mathcal{C}^\top S_P\mathcal{C}P\mathcal{A}^\top + \mathcal{B}_w Q_w\mathcal{B}_w^\top + \mathcal{B}_v R_v\mathcal{B}_v^\top \\
+ & \left(I - M\mathcal{B}_u S_M\mathcal{B}_u^\top\right)^\top \mathcal{A}P\mathcal{C}^\top S_P\mathcal{C}P\mathcal{A}^\top\left(I - M\mathcal{B}_u S_M\mathcal{B}_u^\top\right),
\end{aligned}$$

where (d) follows from the Moore-Penrose conditions, and in (e) we have added and subtracted the term $\mathcal{A}P\mathcal{C}^\top S_p\mathcal{C}P\mathcal{A}^\top$. The Riccati equation of M is derived in similar manner.

Second-order sufficient conditions: We show the stationary points (6.35) correspond to a local minimum. We begin by noting that dP in (6.32) can be written as $dP = V + V^\top$ with

$$V = \sum_{i=0}^{\infty}\mathcal{A}_c^i\left(\mathcal{B}_u d\mathcal{K}\mathcal{C}P\mathcal{A}_c^\top\right)\left(\mathcal{A}_c^\top\right)^i. \tag{6.36}$$

Further, the first-order stationary condition in (6.34) implies

$$(\underbrace{\mathcal{C}^\top\mathcal{K}^\top R_u + \mathcal{A}_c^\top M\mathcal{B}_u}_{X})^\top P\mathcal{C}^\top = 0 \overset{\text{(f)}}{\Longrightarrow} X^\top\mathcal{A}_c{}^i P\mathcal{C}^\top = 0, \tag{6.37}$$

where (f) follows from Lemma 58 for $i = 1, 2, \cdots, \infty$. Then, using (6.37), we have,

$$VX = \sum_{i=0}^{\infty} \mathcal{A}_c^i (\mathcal{B}_u d\mathcal{K} \underbrace{\mathcal{C} P \mathcal{A}_c^{\mathsf{T}}) (\mathcal{A}_c^{\mathsf{T}})^i X}_{=0} = 0. \tag{6.38}$$

Next, we compute the second-order differential of P and $Q_{\mathcal{K}}$. Taking the differential of dP in (6.32) and noting that $d^2\mathcal{K} = 0$,

$$d^2P = \mathcal{A}_c d^2 P \mathcal{A}_c^{\mathsf{T}} + 2d\mathcal{A}_c dP \mathcal{A}_c^{\mathsf{T}} + 2\mathcal{A}_c dP d\mathcal{A}_c^{\mathsf{T}} + 2d\mathcal{A}_c P d\mathcal{A}_c^{\mathsf{T}}. \tag{6.39}$$

Taking the differential of $dQ_{\mathcal{K}}$ in (6.33), we get

$$d^2 Q_{\mathcal{K}} = 2\mathcal{C}^{\mathsf{T}} d\mathcal{K}^{\mathsf{T}} R_u d\mathcal{K}\mathcal{C}. \tag{6.40}$$

Now we are ready to compute the second-order differential of $\mathcal{J}_z$. Taking the differential of $d\mathcal{J}_z$ in (6.34), we get

$$\begin{aligned}
d^2\mathcal{J}_z =& \mathsf{tr}\left[d^2 Q_{\mathcal{K}} P\right] + 2\mathsf{tr}\left[dQ_{\mathcal{K}} dP\right] + \mathsf{tr}\left[Q_{\mathcal{K}} d^2 P\right] \\
&\overset{(g)}{=} 4\mathsf{tr}\left[d\mathcal{K}\mathcal{C}dP\left(\mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}}R_u + \mathcal{A}_c^{\mathsf{T}} M \mathcal{B}_u\right)\right] + 2\mathsf{tr}\left[d\mathcal{K}\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right) d\mathcal{K}^{\mathsf{T}}\left(R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u\right)\right] \\
&\overset{(h)}{=} 4\mathsf{tr}\left[d\mathcal{K}\mathcal{C}V^{\mathsf{T}}X\right] + 2\mathsf{tr}\left[d\mathcal{K}\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right) d\mathcal{K}^{\mathsf{T}}\left(R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u\right)\right] \\
&\overset{(i)}{=} 4\sum_{i=0}^{\infty} \mathsf{tr}\left[d\mathcal{K}\mathcal{C}\mathcal{A}_c^{i+1} P \mathcal{C}^{\mathsf{T}} d\mathcal{K}^{\mathsf{T}} \mathcal{B}_u^{\mathsf{T}} (\mathcal{A}_c^i)^{\mathsf{T}}\right] + 2\mathsf{tr}\left[d\mathcal{K}\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right) d\mathcal{K}^{\mathsf{T}}\left(R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u\right)\right] \\
&\overset{(j)}{=} \sum_{i=0}^{\infty} (\mathrm{vec}\,(d\mathcal{K}))^{\mathsf{T}} \left(4\left(\mathcal{C}\mathcal{A}_c^{i+1} P \mathcal{C}^{\mathsf{T}}\right) \otimes \left(X^{\mathsf{T}} \mathcal{A}_c^i \mathcal{B}_u\right)\right) \mathrm{vec}\,(d\mathcal{K}) \\
&\quad + (\mathrm{vec}\,(d\mathcal{K}))^{\mathsf{T}} \left(2\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right) \otimes \left(R_u + \mathcal{B}_u^{\mathsf{T}} M \mathcal{B}_u\right)\right) \mathrm{vec}\,(d\mathcal{K}),
\end{aligned} \tag{6.41}$$

where in step (g) we have used (6.39), (6.40), Lemma 57, and the cyclic property of trace, in step (h) we have used (6.38), in step (i) we have used (6.36), and in step (j) we have used Fact 55. The second-order differential in (6.41) can be rewritten as

$$d^2\mathcal{J}_z = (\mathrm{vec}\,(d\mathcal{K}))^{\mathsf{T}} Y \mathrm{vec}\,(d\mathcal{K}),$$

where,

$$Y \triangleq 4\sum_{i=0}^{\infty}\left(\mathcal{C}\mathcal{A}_c^{i+1}P\mathcal{C}^{\mathsf{T}}\right)\otimes\left(X^{\mathsf{T}}\mathcal{A}_c^{i}\mathcal{B}_u\right)+2\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right)\otimes\left(R_u+\mathcal{B}_u^{\mathsf{T}}M\mathcal{B}_u\right).$$

Since $\mathcal{J}_z$ is twice differentiable, using [63, Theorem 6 in Chapter 6], we can compute the Hessian of $\mathcal{J}_z$ as

$$\begin{aligned}H =&\frac{1}{2}\left(Y+Y^{\mathsf{T}}\right)\\ =&2\sum_{i=0}^{\infty}\left(\mathcal{C}\mathcal{A}_c^{i+1}P\mathcal{C}^{\mathsf{T}}\right)\otimes\left(X^{\mathsf{T}}\mathcal{A}_c^{i}\mathcal{B}_u\right)+2\sum_{i=0}^{\infty}\left(\mathcal{C}P(\mathcal{A}_c^{i+1})^{\mathsf{T}}\mathcal{C}^{\mathsf{T}}\right)\otimes\left(\mathcal{B}_u^{\mathsf{T}}(\mathcal{A}_c^{i})^{\mathsf{T}}X\right)\\ &+2\left(\mathcal{C}P\mathcal{C}^{\mathsf{T}}\right)\otimes\left(R_u+\mathcal{B}_u^{\mathsf{T}}M\mathcal{B}_u\right).\end{aligned}$$

The above expression implies that the Hessian, H, of $\mathcal{J}_z$ evaluated at the stationary point (6.35) is positive semi-definite since $\mathcal{C}P\mathcal{C}^{\mathsf{T}}\succeq 0$ and $\left(R_u+\mathcal{B}_u^{\mathsf{T}}M\mathcal{B}_u\right)\succ 0$. Let $\mathbb{K}_0\subseteq\mathbb{R}^{d\times r}$ denotes the left null space of $\mathcal{C}P\mathcal{C}^{\mathsf{T}}$, and let $\mathbb{K}_0^{\perp}\subseteq\mathbb{R}^{h\times r}$ denotes the orthogonal complement subspace to $\mathbb{K}_0$, with $h=n+(n+1)p$ (see Lemma 65). It can be seen that the Hessian is degenerate at the stationary points (6.35), where it is zero along the subspace $\mathbb{K}_0$[2] and positive along the directions of $\mathbb{K}_0^{\perp}$. Hence, the stationary point in (6.35) correspond to a local minima over the subspace $\mathbb{K}_0^{\perp}$, which is achieved by the unique gain $\mathcal{K}^*$. Next, we show that $\mathcal{J}_z(\mathcal{K}^*)$ remains constant along the directions of the subspace $\mathbb{K}_0$, i.e., we show that $\mathcal{J}_z(\mathcal{K}^*)$ remains constant over all stationary points $\mathcal{K}_{\mathrm{s}}$ in (6.35). From (6.35), we have $\mathcal{K}_{\mathrm{s}}=\mathcal{K}^*+\alpha\mathcal{K}_0$, let P and P_{s} be the solutions of the Lyapunov equation (6.30) that correspond to $\mathcal{K}^*$ and $\mathcal{K}_{\mathrm{s}}$, respectively. Since $\mathcal{K}_0\in\mathbb{K}_0$, we have

$$\mathcal{K}_0\mathcal{C}P\mathcal{C}^{\mathsf{T}}=\mathcal{K}_0\mathcal{C}\left(\sum_{i=0}^{\infty}\mathcal{A}_c^{*i}\Sigma\left(\mathcal{A}_c^{*\mathsf{T}}\right)^{i}\right)\mathcal{C}^{\mathsf{T}}=0, \tag{6.42}$$

[2] Let $\mathcal{K}_0\in\mathbb{K}_0$, then we have $\mathcal{K}_0\mathcal{C}P\mathcal{C}^{\mathsf{T}}=0\implies\mathcal{K}_0\mathcal{C}P=0\implies\mathcal{K}_0\mathcal{C}\mathcal{A}_c^{i}P=0$ for $i\geq 1$, where the last equality follows from Lemma 58.

where $\mathcal{A}_c^* = \mathcal{A} + \mathcal{B}_u \mathcal{K}^* \mathcal{C}$. Then, using Fact 53, we have

$$\begin{aligned} \mathcal{K}_0 \mathcal{C} \mathcal{A}_c^{*i} \Sigma \left(\mathcal{A}_c^{*\mathsf{T}} \right)^i \mathcal{C}^\mathsf{T} &= 0, \quad \text{for } i = 0, 1, \cdots, \infty, \\ \implies \mathcal{K}_0 \mathcal{C} \mathcal{A}_c^{*i} \Sigma^{1/2} &= 0, \quad \text{for } i = 0, 1, \cdots, \infty. \end{aligned} \tag{6.43}$$

Using (6.43), we can show that the following equation holds for $i = 1, 2, \cdots, \infty$

$$\mathcal{A}_c \mathcal{A}_c^{*i-1} \Sigma \left(\mathcal{A}_c^{*\mathsf{T}} \right)^{i-1} \mathcal{A}_c{}^\mathsf{T} = \mathcal{A}_c^{*i} \Sigma \left(\mathcal{A}_c^{*\mathsf{T}} \right)^i, \tag{6.44}$$

where $\mathcal{A}_c = \mathcal{A} + \mathcal{B}_u \mathcal{K}_\mathrm{s} \mathcal{C} = \mathcal{A}_c^* + \mathcal{B}_u \alpha \mathcal{K}_0 \mathcal{C}$. Using (6.44), we get

$$\begin{aligned} \mathcal{A}_c^{*i} \Sigma \left(\mathcal{A}_c^{*\mathsf{T}} \right)^i &= \mathcal{A}_c \mathcal{A}_c^{*i-1} \Sigma \left(\mathcal{A}_c^{*\mathsf{T}} \right)^{i-1} \mathcal{A}_c{}^\mathsf{T} \\ &= \mathcal{A}_c{}^2 \mathcal{A}_c^{*i-2} \Sigma \left(\mathcal{A}_c^{*\mathsf{T}} \right)^{i-2} \left(\mathcal{A}_c{}^\mathsf{T} \right)^2 \\ &= \cdots = \mathcal{A}_c{}^i \Sigma \left(\mathcal{A}_c{}^\mathsf{T} \right)^i, \quad \text{for } i = 1, 2, \cdots, \infty. \end{aligned}$$

Hence, we have $P_\mathrm{s} = P$. Finally, we can write

$$\begin{aligned} \mathcal{J}_z(\mathcal{K}_\mathrm{s}) &= \mathsf{tr} \left[\left(Q_z + \mathcal{C}^T (\mathcal{K}^* + \alpha \mathcal{K}_0)^\mathsf{T} R_u (\mathcal{K}^* + \alpha \mathcal{K}_0) \mathcal{C} \right) P_\mathrm{s} \right] \\ &= \mathsf{tr} \left[\left(Q_z + \mathcal{C}^\mathsf{T} \mathcal{K}^{*\mathsf{T}} R_u \mathcal{K}^* \mathcal{C} \right) P \right] = \mathcal{J}_z(\mathcal{K}^*), \end{aligned}$$

where we have used (6.42). To conclude the proof, since over the subspace $\mathbb{K}_0^\perp$ the necessary and sufficient conditions for a local minimum are satisfied by a unique gain $\mathcal{K}^*$, the local minimum is also a global minimum over $\mathbb{K}_0^\perp$. Further, this minimum is flat along the directions of the subspace $\mathbb{K}_0$. Therefore, the stationary points in (6.35) are the global minimizers of (6.8). ∎

6.3.7 Proof of Theorem 45

We begin by showing that the gain in (6.10) belongs to a stationary point of the behavioral LQG problem. Using Lemma 63, we can write the behavioral LQG problem as

$$\begin{aligned} \min_{\mathcal{K}} \quad & \mathcal{J}_z(\mathcal{K}) \triangleq \mathsf{tr}\left[\left(Q_z + \mathcal{C}^{\mathsf{T}}\mathcal{K}^{\mathsf{T}}R_u\mathcal{K}\mathcal{C}\right)P\right] \\ \text{s.t.} \quad & P = \mathcal{A}_c P \mathcal{A}_c^{\mathsf{T}} + \Sigma, \end{aligned} \tag{6.45}$$

where $\mathcal{A}_c \triangleq \mathcal{A} + \mathcal{B}_u\mathcal{K}\mathcal{C}$ and $\Sigma \triangleq \mathcal{B}_w Q_w \mathcal{B}_w^{\mathsf{T}} + \mathcal{B}_v R_v \mathcal{B}_v^{\mathsf{T}}$. Let

$$\mathcal{A} \triangleq \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad \mathcal{B}_u \triangleq \begin{bmatrix} B_1 \\ 0 \end{bmatrix},$$

$$P \triangleq \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad Q_z \triangleq \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix},$$

where $A_{11}, P_{11}, Q_{11} \in \mathbb{R}^{r\times r}$, $A_{22}, P_{22}, Q_{22} \in \mathbb{R}^{q\times q}$, $A_{12}, P_{12}, P_{21}^{\mathsf{T}}, Q_{12}, Q_{21}^{\mathsf{T}} \in \mathbb{R}^{r\times q}$, $B_1 \in \mathbb{R}^{r\times m}$, $r = nm + np + p$, and $q = n^2 + np + p$. We write $\mathcal{J}_z$ in (6.45) as

$$\begin{aligned} \mathcal{J}_z(\mathcal{K}) =& \mathsf{tr}\left[(Q_{11} + \mathcal{K}^{\mathsf{T}}R_u\mathcal{K})P_{11}\right] + \mathsf{tr}\left[Q_{12}P_{21}\right] \\ & + \mathsf{tr}\left[Q_{21}P_{12}\right] + \mathsf{tr}\left[Q_{22}P_{22}\right], \end{aligned}$$

and we can write P in (6.45) as

$$\begin{aligned} P =& \begin{bmatrix} A_{11}+B_1\mathcal{K} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{21}P_{11}^{\dagger}P_{12} \end{bmatrix} \begin{bmatrix} (A_{11}+B_1\mathcal{K})^{\mathsf{T}} & 0 \\ A_{12}^{\mathsf{T}} & A_{22}^{\mathsf{T}} \end{bmatrix} \\ & + \begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix} (P/P_{11}) \begin{bmatrix} A_{12}^{\mathsf{T}} & A_{22}^{\mathsf{T}} \end{bmatrix} + \Sigma, \end{aligned}$$

where $P/P_{11} = P_{22} - P_{21}P_{11}^{\dagger}P_{12}$ is the generalized Schur complement of the block P_{11}. Note that P_{22} does not depend on $\mathcal{K}$, and it follows that the term $\mathsf{tr}\left[Q_{22}P_{22}\right]$ in the cost $\mathcal{J}_z$ does not affect control design. Without loss of generality, let

$$\mathcal{K} = K_1 + K_2 P_{21} P_{11}^{\dagger} + \mathcal{K}_{\text{null}} \tag{6.46}$$

where $\mathcal{K}_{\text{null}}$ is a matrix with appropriate dimension whose rows belong to the left null space of P_{11}. By substituting (6.46) into (6.45), we can equivalently write the LQG problem as

$$\begin{aligned}
\min_{K_1,K_2}\ & \mathsf{tr}\left[(Q_{11}+K_1^{\mathsf{T}} R_u K_1)P_{11}\right] + \mathsf{tr}\left[(Q_{12}+K_1^{\mathsf{T}} R_u K_2)P_{21}\right] \\
& +\mathsf{tr}\left[(Q_{21}+K_2^{\mathsf{T}} R_u K_1)P_{12}\right] + \mathsf{tr}\left[K_2^{\mathsf{T}} R_u K_2 P_{21} P_{11}^{\dagger} P_{12}\right] \\
\text{s.t.}\ \ P = & \underbrace{\begin{bmatrix} A_{11}+B_1K_1 & A_{12}+B_1K_2 \\ 0 & A_{22}\end{bmatrix}}_{\mathcal{A}_1} \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{21}P_{11}^{\dagger}P_{12}\end{bmatrix}}_{\overline{P}} \cdot \begin{bmatrix} (A_{11}+B_1K_1)^{\mathsf{T}} & 0 \\ (A_{12}+B_1K_2)^{\mathsf{T}} & A_{22}^{\mathsf{T}}\end{bmatrix} \\
& + \underbrace{\begin{bmatrix} A_{12} \\ A_{22}\end{bmatrix}}_{\mathcal{A}_2} (P/P_{11}) \begin{bmatrix} A_{12}^{\mathsf{T}} & A_{22}^{\mathsf{T}}\end{bmatrix} + \Sigma,
\end{aligned} \tag{6.47}$$

where we have used $P_{11}^{\dagger} = P_{11}^{\dagger} P_{11} P_{11}^{\dagger}$, which follows from the Moore-Penrose conditions for pseudo-inverse, and $P_{21} = P_{21} P_{11}^{\dagger} P_{11}$, which follows from Lemma 56. Define the Lagrange function of Problem (6.47) as

$$\begin{aligned}
\mathcal{L}(K_1, K_2, P, \Lambda) =& \mathsf{tr}\left[(Q_{11}+K_1^{\mathsf{T}} R_u K_1)P_{11}\right] \\
& + \mathsf{tr}\left[(Q_{12}+K_1^{\mathsf{T}} R_u K_2)P_{21}\right] + \mathsf{tr}\left[(Q_{21}+K_2^{\mathsf{T}} R_u K_1)P_{12}\right] \\
& +\mathsf{tr}\left[K_2^{\mathsf{T}} R_u K_2 P_{21} P_{11}^{\dagger} P_{12}\right] + \mathsf{tr}\left[\Lambda P\right] - \mathsf{tr}\left[\Lambda \mathcal{A}_1 \overline{P} \mathcal{A}_1^{\mathsf{T}}\right] \\
& - \mathsf{tr}\left[\Lambda \mathcal{A}_2 (P/P_{11}) \mathcal{A}_2^{\mathsf{T}}\right] - \mathsf{tr}\left[\Lambda \Sigma\right],
\end{aligned} \tag{6.48}$$

where $\Lambda \in \mathbb{R}^{(r+q)\times(r+q)}$ is the Lagrange multiplier matrix, which we write as a block matrix $\Lambda \triangleq \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22}\end{bmatrix}$, with $\Lambda_{11} \in \mathbb{R}^{r\times r}$, $\Lambda_{12}, \Lambda_{21}^{\mathsf{T}} \in \mathbb{R}^{r\times q}$, and $\Lambda_{22} \in \mathbb{R}^{q\times q}$. Let $K \triangleq [K_1\ \ K_2]$.

The stationary Karush-Kuhn-Tucker (KKT) condition implies $\frac{\partial \mathcal{L}(K,P,\Lambda)}{\partial K} = 0$. For notational convinience, we use $\mathcal{L}$ to denote $\mathcal{L}(K_1, K_2, P, \Lambda)$. Taking the derivative of $\mathcal{L}(K_1, K_2, P, \Lambda)$ in (6.48) with respect to K_1 and K_2, we obtain

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial K_1} &= 2\left(\left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)K_1 - B_1^\mathsf{T}\Lambda_{11}A_{11}\right)P_{11} \\
&\quad + 2\Bigg(\left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)K_2 - B_1^\mathsf{T}\left(\Lambda_{11}A_{12} + \frac{1}{2}\left(\Lambda_{12} + \Lambda_{21}^\mathsf{T}\right)A_{22}\right)\Bigg)P_{21}, \\
\frac{\partial \mathcal{L}}{\partial K_2} &= 2\left(\left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)K_1 - B_1^\mathsf{T}\Lambda_{11}A_{11}\right)P_{12} \\
&\quad + 2\Bigg(\left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)K_2 - B_1^\mathsf{T}\left(\Lambda_{11}A_{12} + \frac{1}{2}\left(\Lambda_{12} + \Lambda_{21}^\mathsf{T}\right)A_{22}\right)\Bigg)P_{21}P_{11}^\dagger P_{12}.
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial K} &= \Big[\left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)K_1 - B_1\Lambda_{11}A_{11} \;\vdots\; \left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)K_2 - B_1^\mathsf{T}\left(\Lambda_{11}A_{12} + \Lambda_{12}A_{22}\right)\Big] \cdot \\
&\qquad \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{21}P_{11}^\dagger P_{12} \end{bmatrix} \\
&= 2\Big(\left(R_u - B_1^\mathsf{T}\Lambda_{11}B_1\right)\begin{bmatrix} K_1 \;\vdots\; K_2 \end{bmatrix} - \begin{bmatrix} B_1^\mathsf{T}\Lambda_{11}A_{11} \;\vdots\; B_1^\mathsf{T}\Lambda_{11}A_{12} + \Lambda_{12}A_{22} \end{bmatrix}\Big) \cdot \\
&\qquad \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{21}P_{11}^\dagger P_{12} \end{bmatrix} \\
&= 2\Big(\left(R_u - \mathcal{B}_u^\mathsf{T}\Lambda\mathcal{B}_u\right)K - \mathcal{B}_u^\mathsf{T}\Lambda\mathcal{A}\Big)\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{21}P_{11}^\dagger P_{12} \end{bmatrix}
\end{aligned} \tag{6.49}$$

Let $\Lambda = \Lambda^\mathsf{T}$, then we can write $\frac{\partial \mathcal{L}}{\partial K}$ as in (6.49). Let $K_{\text{LQR}} = -(R_u + \mathcal{B}_u^\mathsf{T} M_{\text{LQR}}\mathcal{B}_u)^{-1}\mathcal{B}_u^\mathsf{T} M_{\text{LQR}}\mathcal{A}$, with M_{LQR} satisfying the following Riccati equation

$$M_{\text{LQR}} = \mathcal{A}^\mathsf{T} M_{\text{LQR}}\mathcal{A} + Q_z - \mathcal{A}^\mathsf{T} M_{\text{LQR}}\mathcal{B}_u\left(R_u + \mathcal{B}_u^\mathsf{T} M_{\text{LQR}}\mathcal{B}_u\right)^{-1}\mathcal{B}_u^\mathsf{T} M_{\text{LQR}}\mathcal{A}.$$

Then, for any P, $K = K_{\text{LQR}}$, and $\Lambda = -M_{\text{LQR}}$, we get $\frac{\partial \mathcal{L}}{\partial K}(K_{\text{LQR}}, M_{\text{LQR}}, P) = 0$. For P to belong to a stationary point of the Lagrangian, it should satisfy the constraint in Problem (6.45). Using (6.46), we can write $\mathcal{A}_c$ in the constraint of (6.45) as

$$\mathcal{A}_c = \underbrace{\mathcal{A} + \mathcal{B}_u K_1 \mathcal{C}}_{\mathcal{A}_K} + \underbrace{\mathcal{B}_u K_2}_{\mathcal{B}_K} P_{21} P_{11}^{\dagger} \mathcal{C} + \mathcal{K}_{\text{null}} \mathcal{C} = \mathcal{A}_K + \mathcal{B}_K \overline{\mathcal{C}} P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} + \mathcal{K}_{\text{null}} \mathcal{C}, \tag{6.50}$$

where $\overline{\mathcal{C}} \triangleq \begin{bmatrix} 0_{q \times r} & I_q \end{bmatrix}$. Substituting (6.50) into the constraint of (6.45), we get

$$\begin{aligned}
P =& \mathcal{A}_K P \mathcal{A}_K^{\mathsf{T}} + \mathcal{A}_K P \left(\mathcal{B}_K \overline{\mathcal{C}} P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C}\right)^{\mathsf{T}} \\
&+ \mathcal{B}_K \overline{\mathcal{C}} P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \mathcal{A}_K^{\mathsf{T}} + \Sigma \\
&+ \mathcal{B}_K \overline{\mathcal{C}} P \mathcal{C}^{\mathsf{T}} \underbrace{\left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger}}_{\left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger}} \mathcal{C} P \overline{\mathcal{C}}^{\mathsf{T}} \mathcal{B}_K^{\mathsf{T}} \\
\overset{\text{(a)}}{=}& \mathcal{A}_K P \mathcal{A}_K^{\mathsf{T}} - \mathcal{A}_K P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \mathcal{A}_K^{\mathsf{T}} + \Sigma \\
&+ \left(\mathcal{A}_K + \mathcal{B}_K \overline{\mathcal{C}}\right) P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \left(\mathcal{A}_K + \mathcal{B}_K \overline{\mathcal{C}}\right)^{\mathsf{T}} \\
\overset{\text{(b)}}{=}& \mathcal{A}_K P \mathcal{A}_K^{\mathsf{T}} - \mathcal{A}_K P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \mathcal{A}_K^{\mathsf{T}} + \Sigma \\
&+ \left(\mathcal{A} + \mathcal{B}_u K\right) P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \left(\mathcal{A} + \mathcal{B}_u K\right)^{\mathsf{T}},
\end{aligned}$$

where in step (a), we have added and subtracted the term $\mathcal{A}_K P \mathcal{C}^{\mathsf{T}} \left(\mathcal{C} P \mathcal{C}^{\mathsf{T}}\right)^{\dagger} \mathcal{C} P \mathcal{A}_K^{\mathsf{T}}$, and (b) follows by observing that

$$\mathcal{A}_K + \mathcal{B}_K \overline{\mathcal{C}} = \mathcal{A} + \mathcal{B}_u K_1 \mathcal{C} + \mathcal{B}_u K_2 \overline{\mathcal{C}} = \mathcal{A} + \mathcal{B}_u K.$$

Therefore, the gain $\mathcal{K}$ in (6.46) with $\begin{bmatrix} K_1 & K_2 \end{bmatrix} = K_{\text{LQR}}$ in (6.11) and P satisfying the Riccati equation in (6.12), is a stationary point of (6.45), and hence it is connected to $\mathcal{K}_s$ in (6.35) via the subspace $\mathbb{K}_0$ (see the proof of Theorem 43). Thus, $\mathcal{K}_{\text{null}} \in \mathbb{K}_0$ and $P = P_{\text{s}} = P^*$, where P_{s} and P^* denote the solutions of the Lyapunov equation in (6.30) that corresponds $\mathcal{K}_{\mathrm{s}}$ and $\mathcal{K}^*$ in (6.9), respectively. Therefore, we can write $\mathcal{K}_{\mathrm{null}} = \alpha\mathcal{K}_0$, where $\alpha \in \mathbb{R}^{m\times d}$ is an arbitrary matrix, and $\mathcal{K}_0 \in \mathbb{R}^{d\times r}$ is a matrix whose rows span the left null space of $\mathcal{C}P^*\mathcal{C}^\top = \mathcal{C}P\mathcal{C}^\top = \mathcal{C}P_{\mathrm{s}}\mathcal{C}^\top$, with $d = nm - n$ and $r = nm + np + p$ (see Lemma 65). To conclude the proof, since all stationary points in (6.35) are the global minimizers of (6.8), then $\mathcal{K}$ in (6.46) with $\begin{bmatrix} K_1 & K_2 \end{bmatrix} = K_{\mathrm{LQR}}$ in (6.11), P satisfying the Riccati equation in (6.12), and $\mathcal{K}_{\mathrm{null}} = \alpha\mathcal{K}_0$, is also the global minimizer of (6.8). ■

6.3.8 Proof of Lemma 49

$\mathcal{K}_2$ in Lemma 49 corresponds to the first block of $\mathcal{T}_2 - GE^n\mathcal{T}_1^\dagger\mathcal{M}$ in (6.24). We start by expanding $GE^n\mathcal{T}_1^\dagger\mathcal{M}$. Since $\mathcal{T}_1^\dagger$ is full column rank, we have

$$\mathcal{T}_1^\dagger = \left(\mathcal{T}_1^\top\mathcal{T}_1\right)^{-1}\mathcal{T}_1^\top = \underbrace{\left(G^\top G + \cdots + (E^{n-1})^\top G^\top G E^{n-1}\right)^{-1}}_{\triangleq S}\mathcal{T}_1^\top,$$

then we have

$$GE^n\mathcal{T}_1^\dagger\mathcal{M} = GE^nS\left[\begin{array}{c:c} G^\top H + \cdots + (E^{n-1})^\top G^\top G E^{n-2}F & \mathrm{X} \end{array}\right],$$

where X denotes any matrix. Then, we take the first block of $GE^n\mathcal{T}_1^\dagger\mathcal{M}$ and the first block of $\mathcal{T}_2$ to write $\mathcal{K}_2$ as

$$\begin{aligned}\mathcal{K}_2 =& GE^{n-1}F - GE^nS\left(G^\top H + \cdots + (E^{n-1})^\top G^\top G E^{n-2}F\right)\\ \overset{(a)}{=}& GE^{n-1}F - GE^n\underbrace{S\left(\overline{G}^\top\overline{G} + \cdots + (\overline{E}^{n-1})^\top\overline{G}^\top\overline{G}\,\overline{E}^{n-1}\right)}_{\overset{(b)}{=}I}\overline{F} \overset{(c)}{=} GE^{n-1}F - GE^{n-1}F = 0,\end{aligned}$$

where in steps (a), (b) and (c) we have used Lemma 59. ■

6.3.9 Proof of Lemma 50

Since the rank of Y_N in (6.14) is $\mathrm{Rank}(Y_N) \leq nm + np$, $k = nm + np$ columns are enough for $\mathrm{Rank}(Y_N)$ to stop increasing. To construct Y_N with $k = nm + np$ columns, $nm+np+n$ samples are required. Therefore, $N = nm + np + n$ expert samples are sufficient to learn the LQG gain $\mathcal{K}$. This completes the proof. ■

Chapter 7

Sample Complexity of the Linear Quadratic Gaussian Regulator

Leveraging the behavioral representation introduced in Chapter 6, in this chapter, we provide direct data-driven expressions for the Linear Quadratic Regulator (LQR), the Kalman filter, and the Linear Quadratic Gaussian (LQG) controller using a finite dataset of noisy input, state, and output trajectories. We show that our data-driven expressions are consistent, since they converge as the number of experimental trajectories increases, we characterize their convergence rate, and we quantify their error as a function of the system and data properties. These results complement the body of literature on data-driven control and finite-sample analysis, and they provide new ways to solve canonical control and estimation problems that do not assume, nor require the estimation of, a model of the system and noise and do not rely on solving implicit equations. The results of this chapter are reported in our published paper [69].

7.1 Problem formulation and preliminary results

Consider the discrete-time, linear, time-invariant system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t), \\ y(t) &= Cx(t) + v(t), \qquad t \geq 0, \end{aligned} \tag{7.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state, $u(t) \in \mathbb{R}^m$ the control input, $y(t) \in \mathbb{R}^p$ the measured output, $w(t)$ and $v(t)$ the process and measurement noise at time t. The LQG control problem asks for the input that minimizes the cost function

$$\lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\Big(\sum_{t=0}^{T-1} x(t)^\mathsf{T} Q_x x(t) + u(t)^\mathsf{T} R_u u(t)\Big)\right], \tag{7.2}$$

where $Q_x \succeq 0$, $R_u \succ 0$ are weight matrices and T is the control horizon. With the standard assumptions that

(A1) the process and measurement noise sequences and the initial state are independent at all times and satisfy $w(t) \sim \mathcal{N}(0, Q_w)$, $v(t) \sim \mathcal{N}(0, R_v)$, and $x(0) \sim \mathcal{N}(0, \Sigma_0)$, with $Q_w \succeq 0$, $R_v \succ 0$, and $\Sigma_0 \succ 0$;

(A2) the pairs (A, B) and $(A, Q_w^{\frac{1}{2}})$ are controllable, and the pairs (A, C) and $(A, Q_x^{\frac{1}{2}})$ are observable;

the input that solves the LQG problem can obtained by concatenating the Kalman filter for (7.1) with the (static) controller that solves the LQR problem for (7.1) with weight matrices Q_x and R_u [113]. That is,

$$u^*(t) = K_{\mathrm{LQR}} x_{\mathrm{KF}}(t), \tag{7.3}$$

where $x_{\mathrm{KF}}(t)$ is the Kalman estimate of the state $x(t)$. The classic, model-based computation of the LQR gain and Kalman filter in (7.3) requires the complete knowledge of the system (7.1), including the noise statistics. Motivated by the recent successes of data-driven and machine-learning methods, we seek here a solution to the LQG problem that relies only on a (finite) dataset of experimental data, without the need to estimate the system dynamics and noise statistics.

Our aim is to compute the LQG inputs in a data-driven setting where datasets from offline experiments are available but the system matrices and noise statistics are unknown. In particular, we have access to the following data:

$$U = \begin{bmatrix} u^1 \cdots u^N \end{bmatrix}, \; X = \begin{bmatrix} x^1 \cdots x^N \end{bmatrix}, \; Y = \begin{bmatrix} y^1 \cdots y^N \end{bmatrix}, \tag{7.4}$$

where x^i and y^i are the i-th state and output trajectories of (7.1) generated by the input u^i. That is, for $i \in \{1, \dots, N\}$,

$$u^i = \begin{bmatrix} u^i(0) \\ \vdots \\ u^i(T-1) \end{bmatrix}, x^i = \begin{bmatrix} x^i(0) \\ \vdots \\ x^i(T) \end{bmatrix}, y^i = \begin{bmatrix} y^i(0) \\ \vdots \\ y^i(T) \end{bmatrix},$$

where T is the horizon of the control experiments. We make the following assumption on the experimental inputs.

Assumption 66 ***(Experimental inputs)*** *The inputs in* (7.4) *are independent and identically distributed, that is,* $u^i(t) \sim \mathcal{N}(0, \Sigma_u)$*, with* $\Sigma_u \succ 0$*, for all* $i \in \{1, \dots, N\}$ *and times.*
□

In our analysis we will make use of an equivalent characterization of the LQG inputs derived in [67, Theorem 2.1], which shows that these inputs can also be computed as

$$u^*(t+n) = K_{\mathrm{LQG}} \begin{bmatrix} u^*(t) \\ \vdots \\ u^*(t+n-1) \\ y^*(t+1) \\ \vdots \\ y^*(t+n) \end{bmatrix}, \tag{7.5}$$

where the static gain K_{LQG} depends on the system and noise matrices, and y^* is the output of (7.1) with input u^*.

Remark 67 ***(State vs output measurements)*** *We assume here that the state of the system (7.1) can be directly measured. This assumption is easily satisfied in certain lab experiments, where additional sensors (e.g., a motion capture system for robotic applications) can be deployed during the design stage to measure the system state and collect training data. Further, state measurements are necessary to solve the state-weighted LQG problem, since the state weight matrix Q_x uses specific coordinates that cannot be inferred from output measurements only [94], but they can be substituted with input and output measurements for different versions of the LQG problem. See also [67] for a reformulation of the LQG problem that uses only input and output measurements.* □

7.2 Data-driven formulas for LQG control

In this section we derive our main results, that is, direct data-driven formulas for the LQR controller, the Kalman filter, and the LQG controller using the data (7.4).

Additionally, we show that these formulas are consistent, i.e., they converge to the true model-based expressions as the data grows, and we finally quantify their error when the data is finite.

We start by introducing some additional notation. Let

$$X_t = \begin{bmatrix} x^1(t)^\mathsf{T} & \dots & x^N(t)^\mathsf{T} \end{bmatrix}^\mathsf{T}, \tag{7.6}$$

and, given input and state trajectories $u_\mathrm{v} \in \mathbb{R}^{mT}$ and $x_\mathrm{v} \in \mathbb{R}^{nT}$, let $u_\mathrm{m} \in \mathbb{R}^{m\times T}$ and $x_\mathrm{m} \in \mathbb{R}^{n\times T}$ be the matrices obtained by reorganizing the inputs and states in the vectors u_v and x_v in chronological order. The next result characterizes the LQR gain from data.

Theorem 68 ***(Data-driven LQR gain)*** *Let $x_0 \in \mathbb{R}^n$ and*

$$\begin{bmatrix} u_\mathrm{v} \\ x_\mathrm{v} \end{bmatrix} = \begin{bmatrix} H \\ M \end{bmatrix} P^{-1/2} \left(\begin{bmatrix} I_n & 0_{n\times mT} \end{bmatrix} P^{-1/2} \right)^\dagger x_0, \tag{7.7}$$

where

$$H = \begin{bmatrix} 0_{mT\times n} & I_{mT} \end{bmatrix}, M = X \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger, \text{ and} \tag{7.8}$$
$$P = M^\mathsf{T} \left(I_{T+1} \otimes Q_x\right) M + \mathrm{blkdiag}\left(0_{n\times n}, I_T \otimes R_u\right).$$

Let $x_\mathrm{v}^* \in \mathbb{R}^{nT}$ be the trajectory of (7.1) *with initial state x_0, control input $u^*(t) = K_\mathrm{LQR} x(t)$, and $w(t) = 0$ at all times. Then, the data-driven estimate $K_{LQR}^D = u_\mathrm{m} x_\mathrm{m}^\dagger$ of K_{LQR}*

$$\|K_\mathrm{LQR} - K_{LQR}^D\|_2 \leq \frac{1}{\sigma_\mathrm{min}(x_\mathrm{m}^*)\left(1 - \kappa(x_\mathrm{m}^*)\right)} \left(\frac{c_1}{\sqrt{N}} + c_2 \rho^T \right),$$

for sufficiently large N and probability at least $1 - 6\delta$, where

$$\kappa(x_\mathrm{m}^*) = \frac{\sigma_\mathrm{max}(x_\mathrm{m} - x_\mathrm{m}^*)}{\sigma_\mathrm{min}(x_\mathrm{m}^*)},$$

where the constants c_1 and c_2 are independent of N and are defined in (7.29), $\rho < 1$, and $\delta \in [0, 1/6]$. □

A proof of Theorem 68 is postponed to subsection 7.3.2. Some comments are in order. First, Theorem 68 provides a direct, data-driven way to estimate the LQR gain from noisy data, namely, $K_{\mathrm{LQR}}^{\mathrm{D}}$, and characterizes the error between the true and the estimated gains. Such error vanishes as the number (N) and length (T) of the experimental trajectories grow.[1] Further, the term $\kappa(x_{\mathrm{m}}^*)$ also vanishes as the number of experimental trajectories increases (see Theorem 78). Second, the vectors u_{v} and x_{v} contain an estimate of the optimal input and state trajectories that minimize the LQR cost with matrices Q_x and R_u for the system (7.1) with initial state x_0 and without process noise. Notably, these trajectories are estimated using the noisy dataset (7.4). Thus, this result extends the analysis in [16]. Third, Theorem 68 is valid when N is sufficiently large. In particular, N needs to be at least large enough to satisfy $\kappa(x_{\mathrm{m}}^*) < 1$ (see subsection 7.3.2 for other conditions on N). Also, the result holds with probability $1 - 6\delta$, and the specific choice of δ affects the magnitude of the constant c_1. Fourth and finally, although formulas with similar convergence rates for the estimation of the LQR exist [23,70], Theorem 68 provides an alternative, direct, closed-form expression of the gain, as opposed to indirect and optimization-based approaches. This will allow us to estimate the LQG controller.

Example 69 ***(Estimating the LQR gain from noisy data)*** *Consider System (7.1)*

[1] The constant c_1, as well as other constants defined later in the paper, depend also on the horizon T. While a detailed characterization of the effects of this dependency requires a dedicated analysis, notice that our expressions remain consistent if N grows sufficiently faster than T. The formulas in the paper quantify the error for finite choices of these two parameters.

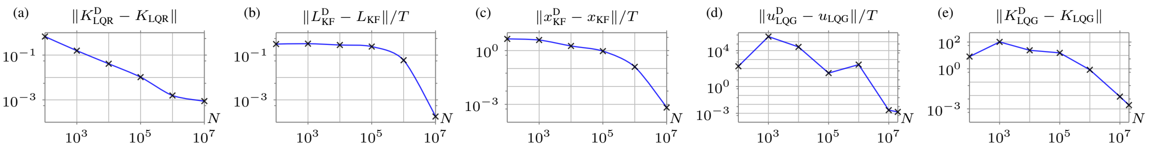


Figure 7.1: This figure shows the error between the data-driven and the model-based gains as a function of the size of the data in (7.4) for the setting described in Example 69, 71, and 73. Panel (a) shows the error between the model-based LQR gain and the data-based LQR gain obtained from Theorem 68 as a function of N for the setting described in Example 69. Panel (b) shows the error between the model-based Kalman filter and the data-based Kalman filter obtained from Theorem 70, and panel (c) shows the error between the corresponding state estimates as a function of N for the setting in Example 71. Panel (d) shows the error between the model-based LQG input generated by (7.3) and the data-based LQG input generated by (7.11) as a function of N. Panel (e) shows the error between the model-based LQG gain in (7.5) and the data-based LQG gain obtained from Theorem 72 as a function of N for the setting in Example 73. We observe that all the quantities in the plots decrease as the number of trajectories, N, increases, which agrees with our theoretical results.

with

$$A = \begin{bmatrix} 0.7 & 1.2 \\ 0 & 0.4 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

$Q_x{=}5I_2$*,* $Q_w{=}2I_2$*,* $R_u = R_v = \Sigma_u{=}1$*, and* $\Sigma_0 = I_2$*. We collect open-loop trajectories as in* (7.4) *generated by inputs satisfying Assumption 66 with horizon* $T{=}50$*. The model-based LQR gain is* $K_{LQR}{=}[0.241 \quad 0.788]$*. We use Theorem 68 to compute the data-driven LQR gain,* K_{LQR}^D *for different values of* N*. Fig. 7.1(a) shows the error* $\|K_{LQR}^D{-}K_{LQR}\|$ *as a function of the number of trajectories.* □

We now focus on estimating the Kalman filter from noisy data with unknown system dynamics and noise statistics.

Theorem 70 ***(Data-driven Kalman filter)*** *Let* U_t *and* Y_t *be the submatrices of* U *and*

Y in (7.4) obtained by selecting only the inputs and outputs up to time t. Let

$$L_t^D = X_t \begin{bmatrix} U_{t-1} \\ Y_t \end{bmatrix}^{\dagger}, \tag{7.9}$$

where X_t is as in (7.6). Then, for every $t \in [0, T]$,

$$\left\| x_{\mathrm{KF}}(t) - L_t^D \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2 \leq \frac{c_3}{\sqrt{N}} \left\| \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2, \tag{7.10}$$

with probability at least $1-2\delta$, where u_0^t and y_0^t are the vectors of inputs and outputs of (7.1), respectively, from time 0 up to time t, c_3 is a constant independent of N as defined in (7.35), and $\delta \in [0, 1/2]$. □

A proof of Theorem 70 is postponed to subsection 7.3.3. Theorem 70 provides a way to construct an approximate Kalman filter using a finite set of experimental data, without knowing the system dynamics and the statistics of the noise. As can be seen from (7.10), the error vanishes with rate $1/\sqrt{N}$ as the number of experimental data grows, showing the consistency of the data-driven Kalman filter expressions (7.9).

Example 71 ***(Estimating the Kalman filter from noisy data)*** *Following the setting introduced in Example 69, we use Theorem 70 to obtain the data-driven Kalman filter, L_{KF}^D, and the corresponding data-driven state estimate, x_{KF}^D. Fig. 7.1(b) and 7.1(c) show the errors $\|L_{KF}^D - L_{KF}\|$ and $\|x_{KF}^D - x_{KF}\|$.* □

Theorems 68 and 70 allow us to compute the LQG inputs from time 0 up to time T.

In particular, recalling the structure of the LQG inputs due to the separation principle [113],

$$u_{\mathrm{dLQG}}(t) = K_{\mathrm{LQR}}^{\mathrm{D}}\, \underbrace{L_t^{\mathrm{D}} \begin{bmatrix} u_{\mathrm{dLQG}}(0) \\ \vdots \\ u_{\mathrm{dLQG}}(t-1) \\ y_{\mathrm{dLQG}}(0) \\ \vdots \\ y_{\mathrm{dLQG}}(t) \end{bmatrix}}_{x_{\mathrm{KF}}^{\mathrm{D}}(t)}, \tag{7.11}$$

where $x_{\mathrm{KF}}^{\mathrm{D}}$ is the state estimate obtained using our data-driven scheme. Fig. 7.1(d) shows how these data-driven inputs compare to the model-based LQG inputs as a function of the amount of data. As expected, the performance gap between the data-driven and the model-based schemes shrinks as the amount of data increases. We next provide an estimate of the LQG gain (7.5), which allows us to compute LQG inputs beyond the horizon T of the experimental trajectories. We start by collecting $M \geq n+nm+np$ closed-loop input-output trajectories of system (7.1) driven by the LQG inputs generated from (7.11). In particular,

$$U_{\mathrm{dLQG}} = \begin{bmatrix} u_{\mathrm{dLQG}}^1 \cdots u_{\mathrm{dLQG}}^M \end{bmatrix}, \qquad Y_{\mathrm{dLQG}} = \begin{bmatrix} y_{\mathrm{dLQG}}^1 \cdots y_{\mathrm{dLQG}}^M \end{bmatrix}, \tag{7.12}$$

where y_{dLQG}^i is the i-th output trajectory of (7.1) generated by the LQG input u_{dLQG}^i in (7.11). That is, for $i \in \{1, \dots, M\}$,

$$u_{\mathrm{dLQG}}^i = \begin{bmatrix} u_{\mathrm{dLQG}}^i(0) \\ \vdots \\ u_{\mathrm{dLQG}}^i(T-1) \end{bmatrix}, \quad y_{\mathrm{dLQG}}^i = \begin{bmatrix} y_{\mathrm{dLQG}}^i(0) \\ \vdots \\ y_{\mathrm{dLQG}}^i(T) \end{bmatrix}.$$

Theorem 72 ***(Data-driven LQG gain)*** *Let U^n_{dLQG} and Y^n_{dLQG} be the submatrices of U_{dLQG} and Y_{dLQG} in (7.12) obtained by selecting only the inputs from time $T-n$ up to time $T-1$ and the outputs from time $T-n+1$ up to time T, respectively. Define the data-driven LQG gain as*

$$K^D_{LQG} = K^D_{LQR} L^D_t \underbrace{\begin{bmatrix} U_{dLQG} \\ Y_{dLQG} \end{bmatrix} \begin{bmatrix} U^n_{dLQG} \\ Y^n_{dLQG} \end{bmatrix}^{\dagger}}_{c_4},$$

Then, the data-driven estimate of the LQG gain satisfies

$$\|K_{LQG} - K^D_{LQG}\|_2 \leq \|c_4\|_2 \left(\frac{c_5 + c_6 \rho^T}{\sqrt{N}} + c_7 \rho^T \right), \tag{7.13}$$

for sufficiently large T and N and probability at least $1 - 8\delta$, where the constants c_5, c_6, and c_7 are independent of N and are defined in (7.38), $\rho < 1$, and $\delta \in [0, 1/8]$. □

We postpone the proof of Theorem 72 to subsection 7.3.4. Theorem (72) provides a direct data-driven expression of the LQG gain that converges with polynomial rate as the experimental data increases. To the best of our knowledge, this result is the first of its kind, and it provides a new way to compute the LQG controller using offline experimental data and a finite number of online experiments, without knowing or identifying the system and noise matrices.

Example 73 ***(Estimating the LQG gain from noisy data)*** *Following the setting introduced in Example 69 and Example 71, we use Theorem 72 to obtain the data-driven LQG gain, K^D_{LQG}. Fig. 7.1(e) shows the error $\|K^D_{LQG} - K_{LQG}\|$ as a function of the number of trajectories, N.* □

7.3 Technical Lemmas and proofs of the main results

7.3.1 Technical lemmas

Lemma 74 ***(Product of Gaussian matrices [23, Lemma 1])*** *Let $A = [a_1, \cdots, a_N]$ and $B = [b_1, \cdots, b_N]$, where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}^m$ are independent random vectors with $a_i \sim \mathcal{N}(0, \Sigma_a)$ and $b_i \sim \mathcal{N}(0, \Sigma_b)$ for $i = 1, \cdots, N$. Let $\delta \in [0, 1]$ and $N \geq 2(n+m) \log(1/\delta)$. Then, with probability at least $1 - \delta$*

$$\|AB^{\mathsf{T}}\|_2 \leq 4\|\Sigma_a\|_2^{1/2}\|\Sigma_b\|_2^{1/2}\sqrt{N(n+m)\log(9/\delta)}.$$

□

Lemma 75 ***(Singular values of a Gaussian matrix)*** *Let $\delta \in [0, 1]$, and let $A \in \mathbb{R}^{n \times N}$ be a random matrix with independent entries distributed as $\mathcal{N}(0, 1)$. Then, for $N \geq 8n + 16 \log(1/\delta)$, each of the following inequalities hold with probability probability at least $1 - \delta$*

$$\sigma_{\min}(A) \geq \sqrt{N}/2, \qquad \sigma_{\max}(A) \leq 3\sqrt{N}/2,$$

where $\sigma_{\min}$ ($\sigma_{\max}$) is the smallest (largest) singular value. □***Proof.*** *For notational convenience, we use $\sigma_{\min}$, $\sigma_{\max}$, and δ' to denote $\sigma_{\min}(A)$, $\sigma_{\max}(A)$, and $2\log(1/\delta)$, respectively. From [98, Corollary 5.35], we have each of the following inequalities holds with probability at least $1 - \delta$*

$$\sigma_{\min} \geq \sqrt{N} - \sqrt{n} - \sqrt{\delta'}, \quad \sigma_{\max} \leq \sqrt{N} + \sqrt{n} + \sqrt{\delta'}. \tag{7.14}$$

Assume that $N \geq 8n + 8\delta'$. Then,

$$\sqrt{N}/2 \geq \sqrt{n} + \sqrt{\delta'}, \tag{7.15}$$

where we have used the inequality $2(a^2 + b^2) \geq (a+b)^2$*. The proof follows by substituting* (7.15) *into* (7.14)*.* ■

7.3.2 Proof of Theorem 68

Let $u_{\mathrm{v}}^* \in \mathbb{R}^{mT}$ and $x_{\mathrm{v}}^* \in \mathbb{R}^{nT}$ be the optimal LQR trajectories of (7.1) from the initial state x_0. Then, $K_{\mathrm{LQR}} = u_{\mathrm{m}}^* x_{\mathrm{m}}^{*\dagger}$ asymptotically as the control horizon T grows. Further, from [16, 17], the trajectories u_{v}^* and x_{v}^* can be obtained using (7.7) when the state data is not corrupted by the process noise. Let X_{clean} be such data, that is, the state trajectories of (7.1) with inputs U and noise $w(t) = 0$ at all times. Notice that in our setting X is different from X_{clean} since the process noise is nonzero when the data is collected. Because of this deviation in the data, the vectors u_{v} and x_{v} in (7.7) are a perturbed version of the optimal trajectories u_{v}^* and x_{v}^*. Accordingly, $K_{\mathrm{LQR}}^{\mathrm{D}} = u_{\mathrm{m}} x_{\mathrm{m}}^{\dagger}$ is a perturbed version of K_{LQR}. To quantify the deviation between $K_{\mathrm{LQR}}^{\mathrm{D}}$ and K_{LQR}, we quantify (i) the deviation in the data induced by the process noise, (ii) the sensitivity of the map (7.7) that generates LQR trajectories, and (iii) how the induced errors propagate to compute $K_{\mathrm{LQR}}^{\mathrm{D}}$.

(i) Data deviation induced by the process noise. Note that

$$X = \underbrace{\left[\begin{array}{c:c} O & F_u \end{array}\right]}_{F} \underbrace{\left[\begin{array}{c} X_0 \\ \hdashline U \end{array}\right]}_{\overline{U}} + F_w W, \tag{7.16}$$

where $W \in \mathbb{R}^{nT \times N}$ is a matrix that contains the corresponding N process noise realizations of horizon $T-1$, and

$$O = \begin{bmatrix} I_n \\ A \\ \vdots \\ A^T \end{bmatrix}, \qquad F_u = \begin{bmatrix} 0 & \cdots & 0 \\ B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{T-1}B & \cdots & B \end{bmatrix}, \qquad F_w = \begin{bmatrix} 0 & \cdots & 0 \\ I_n & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{T-1} & \cdots & I_n \end{bmatrix}.$$

Note that $X_{\mathrm{clean}} = F\overline{U}$. Let the data matrices in (7.4) and (7.6) be partitioned as

$$U = \begin{bmatrix} U_{\mathrm{d}} & U_{\mathrm{n}} \end{bmatrix}, \qquad X = \begin{bmatrix} X_{\mathrm{d}} & X_{\mathrm{n}} \end{bmatrix}, \qquad X_0 = \begin{bmatrix} X_{0,\mathrm{d}} & X_{0,\mathrm{n}} \end{bmatrix}, \tag{7.17}$$

where U_{d}, X_{d}, and $X_{0,\mathrm{d}}$ contain the first $N_{\mathrm{d}} \geq mT+n$ columns of U, X, and X_0, respectively, and let $\overline{U} = [\overline{U}_{\mathrm{d}}, \overline{U}_{\mathrm{n}}]$ be partitioned similarly. For notational convenience, we define $Q_T = (I_{T+1} \otimes Q_x)$ and $R_T = \mathrm{blkdiag}(0_{n\times n}, I_T \otimes R_u)$. Noting that $\overline{U}_{\mathrm{d}}\overline{U}_{\mathrm{d}}{}^{\dagger} = I_{n+mT}$, we rewrite u_{v} in (7.7) as

$$u_{\mathrm{v}} = HP^{-1/2}\left(\begin{bmatrix} I_n & 0_{n\times mT} \end{bmatrix} P^{-1/2}\right)^{\dagger} x_0, \tag{7.18}$$

with

$$P = \left(\widetilde{X}_{\mathrm{c}}\overline{U}_{\mathrm{d}}^{\dagger}\right)^{\mathsf{T}} Q_T \left(\widetilde{X}_{\mathrm{c}}\overline{U}_{\mathrm{d}}^{\dagger}\right) + R_T, \text{ and } \widetilde{X}_{\mathrm{c}} = X\overline{U}^{\dagger}\overline{U}_{\mathrm{d}}. \tag{7.19}$$

Further, let

$$X_{\mathrm{c}} = X_{\mathrm{clean}}\overline{U}^{\dagger}\overline{U}_{\mathrm{d}} \quad \text{and} \quad \Delta_X = \widetilde{X}_{\mathrm{c}} - X_{\mathrm{c}}. \tag{7.20}$$

Notice that if the process noise, W, is zero, then $\Delta_X = 0$ and $\widetilde{X}_{\mathrm{c}} = X_{\mathrm{c}}$ and, from (7.7), $u_{\mathrm{v}} = u_{\mathrm{v}}^*$ and $x_{\mathrm{v}} = x_{\mathrm{v}}^*$. Thus, we use Δ_X as a proxy for the deviation between X and X_{clean}, which is induced by the process noise, F_wW. The next Lemma provides a non-asymptotic upper bound to $\|\Delta_X\|_2$.

Lemma 76 ***(Non-asymptotic bound on*** $\|\Delta_X\|_2$***)*** *Let* Δ_X *be as in* (7.20), *and let* $\delta \in [0, 1/3]$. *Assume that* $N > \max\{N_1, N_d\}$, *with* $N_1 = 2\left((n+m)T+n\right)\log\left(1/\delta\right)$ *and* $N_d \geq 8(mT+n) + 16\log\left(1/\delta\right)$. *Then, with probability at least* $1-3\delta$,

$$\|\Delta_X\|_2 \leq d_1\sqrt{\frac{N_d\left((n+m)T+n\right)\log\left(9/\delta\right)}{N}}, \tag{7.21}$$

where $d_1 = 24\|F_w\|_2\|\overline{Q}_w\|_2^{1/2}$ *and* $\overline{Q}_w = I_T \otimes Q_w$. □

Proof. *Let* $\overline{U} = \Sigma_{\overline{u}}^{1/2}Z$ *and* $\overline{U}_{\mathrm{d}} = \Sigma_{\overline{u}}^{1/2}Z_{\mathrm{d}}$, *where* $\Sigma_{\overline{u}} = \mathrm{blkdiag}(\Sigma_0, I_T \otimes \Sigma_u)$, $Z \in \mathbb{R}^{n+mT \times N}$ *is a random matrix whose columns are independent copies of* $\mathcal{N} \sim (0, I_{n+mT})$, *and* Z_{d} *contains the first* N_d *columns of* Z. *From* (7.19), (7.20),

$$\begin{aligned}\|\Delta_X\|_2 = \|F_w W\overline{U}^{\mathsf{T}}(\overline{U}\,\overline{U}^{\mathsf{T}})^{-1}\overline{U}_{\mathrm{d}}\|_2 = \|F_w WZ^{\mathsf{T}}(ZZ^{\mathsf{T}})^{-1}Z_{\mathrm{d}}\|_2 \\ \leq \|F_w\|_2\|WZ^{\mathsf{T}}\|_2\|(ZZ^{\mathsf{T}})^{-1}\|_2\|Z_{\mathrm{d}}\|_2.\end{aligned}$$

The proof follows by using Lemma 74 to bound $\|WZ^{\mathsf{T}}\|_2$, *Lemma 75 to bound* $\|(ZZ^{\mathsf{T}})^{-1}\|_2$ *and* $\|Z_{\mathrm{d}}\|_2$, *and using the union bound to compute the probability.* ■

(ii) Sensitivity of map (7.7) *w.r.t.* Δ_X. We focus our analysis on the map $f : \mathbb{R}^{n(T+1)N_{\mathrm{d}}} \times \mathbb{R}^{(n+mT)N_{\mathrm{d}}} \to \mathbb{R}^{n+mT}$ that generates u_{v} as in (7.18). Then, $u_{\mathrm{v}}^* = f(\mathrm{vec}(X_{\mathrm{c}}), \mathrm{vec}(\overline{U}_{\mathrm{d}}))$. Since f is Fréchet-differentiable with respect to $\mathrm{vec}(X_{\mathrm{c}})$ [17, 50], we can write its first-order Taylor-series expansion as

$$f(\mathrm{vec}(\widetilde{X}_{\mathrm{c}}), \mathrm{vec}(\overline{U}_{\mathrm{d}})) = f(\mathrm{vec}(X_{\mathrm{c}}), \mathrm{vec}(\overline{U}_{\mathrm{d}})) + \nabla f_X\left(\mathrm{vec}(X_{\mathrm{c}}), \mathrm{vec}(\overline{U}_{\mathrm{d}})\right)\mathrm{vec}(\Delta_X), \tag{7.22}$$

where ∇f_X is the Jacobian matrix of $f(\mathrm{vec}(X_{\mathrm{c}}), \mathrm{vec}(\overline{U}_{\mathrm{d}}))$ with respect to $\mathrm{vec}(X_{\mathrm{c}})$. We quantify the sensitivity of the map (7.18) to the change in X_{c} by ∇f_X (large values of ∇f_X implies higher sensitivity). Next, we derive an upper bound on $\|\nabla f_X\|_2$, and upper bounds on $\|u_{\mathrm{v}} - u_{\mathrm{v}}^*\|_2$ and $\|x_{\mathrm{v}} - x_{\mathrm{v}}^*\|_2$ using the first-order approximation in (7.22).

Lemma 77 ***(Non-asymptotic bound on*** $\|\nabla f_X\|_2$) *Let* $\overline{U}_{\mathrm{d}}$, X_{c}, *and* $\nabla f_X\left(vec(X_{\mathrm{c}}), vec(\overline{U}_{\mathrm{d}})\right)$ *be as in* (7.20) *and* (7.22)*. Also, let* $\delta \in [0,1]$ *and assume that* $N_d \geq 8(n+mT)+16\log(1/\delta)$*. Then, with probability at least* $1-\delta$*,*

$$\left\|\nabla f_X\left(vec(X_{\mathrm{c}}), vec(\overline{U}_{\mathrm{d}})\right)\right\|_2 \leq 4d_2\sqrt{\frac{n(T+1)}{N_d}}, \tag{7.23}$$

where $d_2 > 0$ *is independent of* N_d*.* □

Proof. The proof can be adapted from the proof of [17, Lemma IV.4], then using Lemma 75 and $\|\cdot\|_2 \leq \|\cdot\|_{\mathrm{F}}$. ■

Theorem 78 ***(Non-asymptotic bound on the deviation of the LQR trajectories)*** *Let* u_{v} *and* x_{v} *be as in* (7.7) *and* u_{v}^* *and* x_{v}^* *be the optimal LQR trajectories of length* T *of* (7.1) *from the initial state* x_0*. Let* $\delta \in [0,1/6]$ *and assume that* $N \geq \max\{N_1,N_2,N_3\}$*, with* $N_1 = 2\left((n+m)T+n\right)\log(1/\delta)$*,* $N_2 = 8(mT+n)+16\log(1/\delta)$*, and* $N_3 = ((n+m)T+n)\log(9/\delta)$*. Then, with probability at least* $1-4\delta$*,*

$$\|u_{\mathrm{v}} - u_{\mathrm{v}}^*\|_2 \leq d_3\sqrt{\frac{((n+m)T+n)\log(9/\delta)}{N}}. \tag{7.24}$$

Further, with probability at least $1-6\delta$*,*

$$\|x_{\mathrm{v}} - x_{\mathrm{v}}^*\|_2 \leq d_4\sqrt{\frac{((n+m)T+n)\log(9/\delta)}{N}}, \tag{7.25}$$

with

$$d_3 = 4d_1d_2\sqrt{qn(T+1)},$$
$$d_4 = \|F\|_2 d_3 + 16\|F_w\|_2\|\Sigma_{\overline{u}}^{-1/2}\|_2\|\overline{Q}_w\|_2^{1/2}\left(\|\overline{u}_{\mathrm{v}}^*\|_2 + d_3\right),$$

where d_1*,* F*, and* F_w *are as in* (7.21) *and* (7.16)*, respectively,* $d_2 > 0$ *is independent of* N*,* $q = \mathrm{Rank}\left(\Delta_X\right) \leq n(T+1)$*,* $\overline{u}_{\mathrm{v}}^* = [x_0^{\mathsf{T}}, u_{\mathrm{v}}^{*\mathsf{T}}]^{\mathsf{T}}$*, and* $\overline{Q}_w$ *and* $\Sigma_{\overline{u}}$ *are as in Lemma* 76*.* □

Proof. *Inequality* (7.24) *follows from* (7.22) *by using Lemma 76, Lemma 77, and* $\|vec(\Delta_X)\|_2 = \|\Delta_X\|_{\mathrm{F}} \leq \sqrt{q}\|\Delta_X\|_2$*, with* $q = \operatorname{Rank}(\Delta_X)$*. Next, we derive* (7.25)*. For notational convenience, we use* Δ_u *and* Δ_x *to denote* $u_{\mathrm{v}} - u_{\mathrm{v}}^*$ *and* $x_{\mathrm{v}} - x_{\mathrm{v}}^*$*, respectively. From* (7.7)*, we can write*

$$\begin{aligned} \|\Delta_x\|_2 &= \left\| \widetilde{X}_{\mathrm{c}} \overline{U}_{\mathrm{d}}^{\dagger} \begin{bmatrix} x_0 \\ u_{\mathrm{v}} \end{bmatrix} - X_{\mathrm{c}} \overline{U}_{\mathrm{d}}^{\dagger} \begin{bmatrix} x_0 \\ u_{\mathrm{v}}^* \end{bmatrix} \right\|_2 \qquad (7.26) \\ &= \left\| X_{\mathrm{c}} \overline{U}_{\mathrm{d}}^{\dagger} \begin{bmatrix} 0 \\ \Delta_u \end{bmatrix} + \Delta_X \overline{U}_{\mathrm{d}}^{\dagger} \begin{bmatrix} x_0 \\ u_{\mathrm{v}}^* \end{bmatrix} + \Delta_X \overline{U}_{\mathrm{d}}^{\dagger} \begin{bmatrix} 0 \\ \Delta_u \end{bmatrix} \right\|_2 \\ &\leq \|X_{\mathrm{c}} \overline{U}_{\mathrm{d}}^{\dagger}\|_2 \|\Delta_u\|_2 + \|\Delta_X \overline{U}_{\mathrm{d}}^{\dagger}\|_2 \|\overline{u}_{\mathrm{v}}^*\|_2 + \|\Delta_X \overline{U}_{\mathrm{d}}^{\dagger}\|_2 \|\Delta_u\|_2. \end{aligned}$$

Note that $\overline{U}_{\mathrm{d}} \overline{U}_{\mathrm{d}}^{\dagger} = I_{n+mT}$*. Then we have*

$$\begin{aligned} &\|X_{\mathrm{c}} \overline{U}_{\mathrm{d}}^{\dagger}\|_2 = \|X_{\mathrm{clean}} \overline{U}^{\dagger}\|_2 = \|F\|_2, \\ &\|\Delta_X \overline{U}_{\mathrm{d}}^{\dagger}\|_2 = \|(X - X_{\mathrm{clean}}) \overline{U}^{\dagger}\|_2 = \|F_w W \overline{U}^{\dagger}\|_2 \leq \|F_w\|_2 \|W \overline{U}^{\mathsf{T}}\|_2 \|(\overline{U}\overline{U}^{\mathsf{T}})^{-1}\|_2, \end{aligned}$$

Inequality (7.25) *follows from* (7.26) *by using* (7.24)*, Lemma 74, and Lemma 75 to bound* $\|\Delta_u\|_2$*,* $\|W\overline{U}^{\mathsf{T}}\|_2$*, and* $\|(\overline{U}\overline{U}^{\mathsf{T}})^{-1}\|_2$*, respectively, and noting that for* $N \geq N_3$ *we have* $\frac{\delta'}{N} \leq \sqrt{\frac{\delta'}{N}}$*, with* $\delta' = ((n+m)T + n)\log(9/\delta)$*. The probabilities follow from the union bound.* ■

(iii) Error between K_{LQR} *and* K_{LQR}^{D}*.* We are now ready to conclude the proof of Theorem 68. Notice that

$$u_{\mathrm{m}}^* = K_{\mathrm{LQR}} x_{\mathrm{m}}^*, \text{ and } u_{\mathrm{m}}^* + \delta_u = K_{\mathrm{LQR}}^{\mathrm{D}} (x_{\mathrm{m}}^* + \delta_x), \qquad (7.27)$$

where $\delta_u = u_{\mathrm{m}} - u_{\mathrm{m}}^*$ and $\delta_x = x_{\mathrm{m}} - x_{\mathrm{m}}^*$. Note that u_{m} and x_{m} are the matrices obtained by reorganizing the inputs and states in the vectors u_{v} and x_{v} in chronological order. For notational convenience, we use K and K^{D} to denote K_{LQR} and $K^{\mathrm{D}}_{\mathrm{LQR}}$. Let $\Delta_K = K - K^{\mathrm{D}}$. In what follows, subscript i denotes the i-th row, with $i \in \{1, \cdots, m\}$. Using [100, Theorem 5.1] and assuming that x^*_{m} is of full row rank,[2]

$$\|\Delta_{K,i}\|_2 \leq d_5 \left(\epsilon \|K_i\|_2 \|x^*_{\mathrm{m}}\|_2 + \|\delta_{u,i}\|_2 + \epsilon\alpha\|r\|_2\right), \tag{7.28}$$

where

$$d_5 = \frac{\alpha}{1 - \alpha\epsilon\|x^*_{\mathrm{m}}\|_2}, \quad \epsilon = \frac{\|\delta_x\|_2}{\|x^*_{\mathrm{m}}\|_2}, \quad r = u^*_{\mathrm{m},i} - K_i x^*_{\mathrm{m}},$$

and $\alpha = \|x^*_{\mathrm{m}}\|_2 \|x^{*\dagger}_{\mathrm{m}}\|_2$ is the spectral condition number of x^*_{m}. From [16, Theorem 3.2], we have $\|r\|_2 \leq d_6 \rho^T$, where $d_6 > 0$ and $\rho < 1$, which are independent of N. Since $\|x^*_{\mathrm{m}}\|_2 = \sigma_{\max}(x^*_{\mathrm{m}})$, $\|(x^*_{\mathrm{m}})^\dagger\|_2 = 1/\sigma_{\min}(x^*_{\mathrm{m}})$. Then, we can write d_5 as

$$d_5 = \frac{1}{\sigma_{\min}(x^*_{\mathrm{m}})\left(1 - \kappa(x^*_{\mathrm{m}})\right)}, \quad \text{with} \quad \kappa(x^*_{\mathrm{m}}) = \frac{\sigma_{\max}(\delta_x)}{\sigma_{\min}(x^*_{\mathrm{m}})},$$

For sufficiently large N such that $\sigma_{\max}(\Delta X) < \sigma_{\min}(X^*)$, we have $\kappa(x^*_{\mathrm{m}}) < 1$ and $\epsilon\alpha < 1$. Then, we can write (7.28) as

$$\|\Delta_{K,i}\|_2 \leq d_5 \Big(\|K_i\|_2 \|\delta_x\|_2 + \|\delta_{u,i}\|_2 + d_6 \rho^T\Big), \overset{\text{(a)}}{\leq} d_5 \Big(\|K\|_2 \|\Delta_x\|_2 + \|\Delta_u\|_2 + d_6\rho^T\Big),$$

where in step (a), we have used $\|\delta_x\|_2 \leq \|\delta_x\|_{\mathrm{F}} = \|\mathrm{vec}(\delta_x)\|_2 = \|\Delta_x\|_2$, and $\|\delta_{u,i}\|_2 = \|\delta_{u,i}\|_{\mathrm{F}} \leq \|\delta_u\|_{\mathrm{F}} = \|\mathrm{vec}(\delta_u)\|_2 = \|\Delta_u\|_2$, where Δ_x and Δ_u are as in Theorem 78. Noting that $\|\Delta K\|_{\mathrm{F}} = \sqrt{\mathsf{tr}\left[\Delta K (\Delta K)^{\mathsf{T}}\right]} = \sqrt{\sum_{i=1}^m \|\Delta K_i\|_2^2}$ and using the bounds in Theorem 78, we have with probability at least $1 - 6\delta$

$$\|\Delta_K\|_2 \leq \frac{1}{\sigma_{\min}(x^*_{\mathrm{m}})\left(1 - \kappa(x^*_{\mathrm{m}})\right)} \left(\frac{c_1}{\sqrt{N}} + c_2 \rho^T\right),$$

[2] This condition is typically satisfied for generic choices of the initial state.

where,

$$\begin{aligned} c_1 &= (d_3 + \|K_{\text{LQR}}\|_2 d_4)\sqrt{m\left((n+m)T+n\right)\log\left(9/\delta\right)}, \\ c_2 &= d_6\sqrt{m}, \end{aligned} \tag{7.29}$$

and d_3 and d_4 are as in Theorem 78. Finally, the probability follows from the union bound. This concludes the proof.

7.3.3 Proof of Theorem 70

The Kalman filter computes the estimate $x_{\text{KF}}(t)$ given $\{u(0), \ldots, u(t-1), y(0), \ldots, y(t)\}$ that minimizes the cost

$$\sum_{t=0}^{T} \mathbb{E}\left[(x(t) - x_{\text{KF}}(t))^T (x(t) - x_{\text{KF}}(t))\right], \tag{7.30}$$

which is then used to generate LQG inputs. Equivalently, $x_{\text{KF}}(t)$ can be obtained with the following linear estimator,

$$\begin{aligned} x_{\text{KF}}(t) &= \underbrace{\begin{bmatrix} L^u_{t,0} \cdots L^u_{t,t-1} \end{bmatrix}}_{L^u_t} \underbrace{\begin{bmatrix} u(0) \\ \vdots \\ u(t-1) \end{bmatrix}}_{u_0^{t-1}} + \underbrace{\begin{bmatrix} L^y_{t,0} \cdots L^y_{t,t} \end{bmatrix}}_{L^y_t} \underbrace{\begin{bmatrix} y(0) \\ \vdots \\ y(t) \end{bmatrix}}_{y_0^t}, \\ &= \underbrace{\begin{bmatrix} L^u_t & L^y_t \end{bmatrix}}_{L^{\text{KF}}_t} \underbrace{\begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix}}_{z_t}, \end{aligned}$$

where $L^{\text{KF}}_t \in \mathbb{R}^{n \times mt + p(t+1)}$, with $L^u_t \in \mathbb{R}^{n \times mt}$ and $L^y_t \in \mathbb{R}^{n \times p(t+1)}$, is the estimator gain that minimizes (7.30). Let $e(t) = x(t) - x_{\text{KF}}(t)$ and $\Sigma_{e,t} \in \mathbb{R}^{n \times n} \succeq 0$ denote the estimation error and the estimation error covariance matrix, respectively. For an optimal linear estimator,

L_t^{KF}, we have $e(t) \sim \mathcal{N}(0, \Sigma_{e,t})$, and we can write the state $x(t)$ as

$$x(t) = L_t^{\mathrm{KF}} z_t + e(t).$$

Let

$$x_t = [x^1(t), \ldots, x^N(0)], \quad e_t = [e^1(t), \ldots, e^N(0)], \tag{7.31}$$

where $x^i(t)$ and $e^i(t)$ denote the state and the state estimation error incurred by L_t^{KF} at time t for the i-th trajectory of the data (7.4), respectively. Further, let $Z_t = [U_{t-1}^{\mathsf{T}}, Y_t^{\mathsf{T}}]^{\mathsf{T}}$, where U_t and Y_t are the submatrices of U and Y in (7.4) obtained by selecting the inputs and outputs up to some t. Then,

$$x_t = L_t^{\mathrm{KF}} Z_t + e_t. \tag{7.32}$$

To estimate the optimal filter L_t from the data (7.4), we consider the following least squares problem

$$L_t^{\mathrm{D}} = \arg\min_{L_t} \|x_t - L_t Z_t\|_{\mathrm{F}}^2. \tag{7.33}$$

Problem (7.33) admits a unique solution since Z_t is full-row rank, which is given by (7.9). Next, we bound $\|L_t^{\mathrm{D}} - L_t^{\mathrm{KF}}\|_2$.

Theorem 79 ***(Non-asymptotic bound on*** $\|L_t^{\boldsymbol{D}} - L_t^{\boldsymbol{KF}}\|_2$***)*** *Let L_t^{KF} and L_t^{D} be as in (7.32) and (7.9), respectively, and let $\delta \in [0, 1/2]$. Assume that $N \geq \max\{N_1, N_2\}$, with $N_1 = 2\left((n+m)T + n\right)\log(1/\delta)$ and $N_2 = 8(mT+n) + 16\log(1/\delta)$. Then, with probability at least $1-2\delta$,*

$$\|L_t^{D} - L_t^{KF}\|_2 \leq d_7 \sqrt{\frac{((m+p)t + n + p)\log(9/\delta)}{N}}, \tag{7.34}$$

with

$$d_7 \triangleq 16\|\Sigma_Z^{-1/2}\|_2\|\Sigma_{e,t}\|_2^{1/2},$$

where $\Sigma_Z \in \mathbb{R}^{(m+p)t+p\times(m+p)t+p} \succ 0$ *comprises the noise statistics,* Σ_0*, and* Σ_u *in Assumption 66, and* $\Sigma_{e,t}$ *is the optimal estimation error covariance matrix at time* t*.* □

Proof. Let $Z = \Sigma_Z^{1/2}G$ and where $\Sigma_Z \succ 0$ is as in the theorem statement, and $G \in \mathbb{R}^{(m+p)t+p\times N}$ is a random matrix whose columns are independent random vectors distributed as $\mathcal{N} \sim (0, I_{(m+p)t+p})$. From (7.9) and (7.32),

$$\|L_t^{\mathrm{D}} - L_t^{\mathrm{KF}}\|_2 = \|e_t Z^{\dagger}\|_2 = \|e_t G^{\mathsf{T}}(GG^{\mathsf{T}})^{-1}\Sigma_Z^{-1/2}\|_2 \leq \|\Sigma_Z^{-1/2}\|_2\|e_t G^{\mathsf{T}}\|_2\|(GG^{\mathsf{T}})^{-1}\|_2.$$

The proof follows by using Lemma 74 to bound $\|e_t G^{\mathsf{T}}\|_2$, and Lemma 75 to bound $\|(GG^{\mathsf{T}})^{-1}\|_2$. Finally, The probability follows from the union bound. ■

To conclude the proof of Theorem 70, we have

$$\left\| x_{\mathrm{KF}}(t) - L_t^{\mathrm{D}} \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2 = \left\| L_t^{\mathrm{KF}} \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} - L_t^{\mathrm{D}} \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2 \leq \|L_t^{\mathrm{KF}} - L_t^{\mathrm{D}}\|_2 \left\| \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2,$$

where u_0^t and y_0^t are the vectors of inputs and outputs of (7.1), respectively, from time 0 up to time t. Using Theorem 79,

$$\left\| x_{\mathrm{KF}}(t) - L_t^{\mathrm{D}} \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2 \leq \frac{c_3}{\sqrt{N}} \left\| \begin{bmatrix} u_0^{t-1} \\ y_0^t \end{bmatrix} \right\|_2, \tag{7.35}$$

where $c_3 = d_7\sqrt{((m+p)t+n+p)\log(9/\delta)}$, and d_7 is as in Theorem 79. The above inequality holds with probability at least $1-2\delta$, which follows from Theorem 79 for $\delta \in [0, 1/2]$. This concludes the proof of Theorem 70.

7.3.4 Proof of Theorem 72

Consider the closed-loop trajectories in (7.12), and let U^n_{dLQG} and Y^n_{dLQG} be the submatrices of U_{dLQG} and Y_{dLQG} in (7.12) obtained by selecting only the inputs from time $T-n$ up to time $T-1$ and the outputs from time $T-n+1$ up to time T, respectively. We can write the data-based and the model-based LQG inputs at time T for the trajectories in (7.12) as

$$\underbrace{\begin{bmatrix} u^1_{\mathrm{dLQG}}(T) & \cdots & u^M_{\mathrm{dLQG}}(T) \end{bmatrix}}_{U_{\mathrm{dLQG}}(T)} = K^{\mathrm{D}}_{\mathrm{LQR}} L^{\mathrm{D}}_T \begin{bmatrix} U_{\mathrm{dLQG}} \\ Y_{\mathrm{dLQG}} \end{bmatrix},$$

$$\underbrace{\begin{bmatrix} u^1_{\mathrm{LQG}}(T) & \cdots & u^M_{\mathrm{LQG}}(T) \end{bmatrix}}_{U_{\mathrm{LQG}}(T)} = K_{\mathrm{LQR}} L^{\mathrm{KF}}_T \underbrace{\begin{bmatrix} U_{\mathrm{dLQG}} \\ Y_{\mathrm{dLQG}} \end{bmatrix}}_{Z}.$$

For notational convenience, let $\Delta K_{\mathrm{LQR}} = K^{\mathrm{D}}_{\mathrm{LQR}} - K_{\mathrm{LQR}}$, $\Delta L = L^{\mathrm{D}}_T - L^{\mathrm{KF}}_T$, and $\Delta U = U_{\mathrm{dLQG}}(T) - U_{\mathrm{LQG}}(T)$. Then,

$$\Delta U = K^{\mathrm{D}}_{\mathrm{LQR}} L^{\mathrm{D}}_T Z - K_{\mathrm{LQR}} L^{\mathrm{KF}}_T Z = K_{\mathrm{LQR}} \Delta L Z + \Delta K_{\mathrm{LQR}} L^{\mathrm{KF}}_T Z + \Delta K_{\mathrm{LQR}} \Delta L Z. \tag{7.36}$$

For sufficiently large T, we use (7.5) to write

$$U_{\mathrm{dLQG}}(T) = K^{\mathrm{D}}_{\mathrm{LQG}} \begin{bmatrix} U^n_{\mathrm{dLQG}} \\ Y^n_{\mathrm{dLQG}} \end{bmatrix}, \qquad U_{\mathrm{LQG}}(T) = K_{\mathrm{LQG}} \underbrace{\begin{bmatrix} U^n_{\mathrm{dLQG}} \\ Y^n_{\mathrm{dLQG}} \end{bmatrix}}_{Z_n}.$$

Then, $K_{\mathrm{LQG}}^{\mathrm{D}} = U_{\mathrm{dLQG}}(T)Z_n^{\dagger}$ and $K_{\mathrm{LQG}} = U_{\mathrm{LQG}}(T)Z_n^{\dagger}$. For notational convenience, let $\Delta K_{\mathrm{LQG}} = K_{\mathrm{LQG}}^{\mathrm{D}} - K_{\mathrm{LQG}}$, and let $\|\cdot\|$ denote $\|\cdot\|_2$. Then, using (7.36), we can write

$$\begin{aligned}\|\Delta K_{\mathrm{LQG}}\|_2 =& \|(U_{\mathrm{dLQG}}(T) - U_{\mathrm{LQG}}(T))Z_n^{\dagger}\| = \|\Delta U Z_n^{\dagger}\| \\ &\leq \|K_{\mathrm{LQR}}\|\|\Delta L\|\|ZZ_n^{\dagger}\| + \|\Delta K_{\mathrm{LQR}}\|\|L_T^{\mathrm{KF}}\|\|ZZ_n^{\dagger}\| + \|\Delta K_{\mathrm{LQR}}\|\|\Delta L\|\|ZZ_n^{\dagger}\|.\end{aligned} \tag{7.37}$$

Let $\delta \in [0, 1/8]$ and assume that $N \geq \max\{N_1, N_2, N_3\}$, where N_1, N_2, and N_3 are as in Theorem 78. Then, inequality (7.13) follows by using Theorem 68 and Theorem 79 to bound $\|\Delta K_{\mathrm{LQR}}\|$ and $\|\Delta L\|$ in (7.37), respectively, with probability at least $1-8\delta$ and with

$$\begin{aligned} c_5 =& \frac{c_1\|L_t^{\mathrm{KF}}\| + c_1 c_3}{\sigma_{\min}(x_{\mathrm{m}}^*)\,(1-\kappa(x_{\mathrm{m}}^*))} + c_3\|K_{\mathrm{LQR}}\|, \\ c_6 =& \frac{c_2 c_3}{\sigma_{\min}(x_{\mathrm{m}}^*)(1-\kappa(x_{\mathrm{m}}^*))},\ \ c_7 = \frac{c_2\|L_t^{\mathrm{KF}}\|}{\sigma_{\min}(x_{\mathrm{m}}^*)(1-\kappa(x_{\mathrm{m}}^*))},\end{aligned} \tag{7.38}$$

where c_1, c_2, x_{m}^*, and $\kappa(x_{\mathrm{m}}^*)$ are as in Theorem 68, and c_3 is as in Theorem 70. Finally, the probability follows using the union bound. This concludes the proof of Theorem 72.

Chapter 8

Conclusions

The intricate relationship between performance and robustness in machine learning models stands as a fundamental challenge with far-reaching implications. Throughout this thesis, we have illuminated the inherent fundamental tradeoff these models encounter, where the pursuit of optimal performance often comes at the cost of robustness against perturbations, and diverse and unforeseen conditions.

In this thesis, we have showed the existence of a fundamental tradeoff between performance and robustness of learning models in both classification and control learning problems, where we have provided a comprehensive characterization of these tradeoffs. Moreover, leveraging insights gained from these tradeoffs, we have introduced a robust feedback control policy learning framework based on Lipschitz-constrained loss minimization, where the feedback policies are learned directly from expert demonstrations. Our work integrates robust learning, optimal control and robust stability into a unified framework, enabling the learning of controllers that prioritize both performance and robustness. Finally, we have revisited the LQG optimal control problem from a behavioral perspective, where we have derived direct data-driven expression for the optimal LQG controller using a dataset of input, state, and output trajectories. This analysis highlighted the limitations and challenges posed by noisy data and unknown system dynamics.

In summary, our study sheds light on the necessity of redefining model benchmarks and design strategies to navigate the intricate landscape of performance and robustness tradeoffs. By embracing this nuanced balance, we can pave the way for more dependable, versatile, and impactful machine learning applications across fields.

8.1 Summary and future directions

In what follows, we provide a brief summary of each chapter, followed by a discussion on potential future directions.

Chapter 2. In this chapter, we showe that a fundamental tradeoff exists between the accuracy of a binary classification algorithm and its sensitivity to adversarial manipulation of the data. Thus, accuracy can only be maximized at the expenses of the sensitivity to data manipulation, and this tradeoff cannot be arbitrarily improved by tuning the algorithm's parameters.

Directions of future interest include the extension to M-ary testing problems, as well as the formal characterization of the relationships between the complexity of the classification algorithm and its accuracy versus sensitivity tradeoff.

Chapter 3. In this chapter, we include an abstain option in a binary classification problem, to improve adversarial robustness. We propose metrics to quantify the nominal performance of a classifier with an abstain option and its adversarial robustness. We formally prove that, for any classifier with an abstain option, there exist a tradeoff between its nominal performance and its robustness, thus, the classifier's robustness can only be improved at the expense of its nominal performance. Further, we provide necessary conditions to design the abstain region that optimizes robustness for a desired nominal performance for 1-dimensional binary classification problem. Finally, we validate our theoretical results on the MNIST dataset, where we show that the tradeoff between performance and robustness also exist for the general multi-class classification problems.

Directions of future interest include comparing tradeoffs obtained with an abstain option and tradeoffs obtained via tuning the decision boundaries (from ch: 2), as well as investigating whether it is possible to improve the tradeoff by tuning the boundaries and the abstain region simultaneously.

Chapter 4. In this paper we show that a fundamental trade-off exists between the accuracy of linear estimation algorithms and their robustness to unknown changes of the measurement noise statistics. Because of this trade-off, estimators that are optimal with nominal sensing data may perform poorly in practice due to variations of the measurements statistics or different operational conditions. Conversely, robust estimators obtained through a more detailed design process may maintain similar performance levels in nominal and non-nominal conditions, but considerably underperform in nominal conditions when compared to nominally optimal estimators. To complement these results, we characterize the structure of optimal estimators, for desired levels of accuracy and robustness, and show that the trade-off also constrain the performance of closed-loop perception-based controllers. The results in this chapter complement a recent line of research aimed at deriving provable guarantees and performance limitations of machine learning and data-driven algorithms [**?**, 25, 95, 107], and extend such results, for the first time, to an estimation and control setting.

Directions of future interest include an explicit quantification of the performance of data-driven control algorithms when data is scarce and corrupted. Another area of interest is characterizing how the tradeoff depends on system parameters and noise statistics.

Chapter 5. In this chapter, we propose a framework to learn feedback control policies with provable robustness guarantees. Our approach draws from our earlier work [52] where we formulate the adversarially robust learning problem as one of Lipschitz-constrained loss minimization. We adapt this framework to the problem of learning robust feedback policies from a dataset obtained from expert demonstrations. We establish robust stability of the closed-loop system under the learned feedback policy. Further, we derive upper bounds on the regret and robustness of the learned feedback policy, which bound its nominal suboptimality with respect to the expert policy and the deterioration of its performance under bounded (adversarial) disturbances to state measurements, respectively. The regret bounds suggest the existence of a tradeoff between nominal performance of the feedback policy and closed-loop robustness to adversarial perturbations on the feedback. This tradeoff is also evident in our numerical experiments, where improving closed-loop robustness leads to a deterioration of the nominal performance. Finally, we demonstrate our results and the effectiveness of our robust feedback policy learning framework via numerical experiments on (i) the standard LQR benchmark, and (ii) a nonholonomic differential drive mobile robot model.

Directions of future interest involve extending our framework to learn robust feedback policies from open-loop data, particularly in scenarios where data is scarce or corrupted. Another promising direction is to further develop our framework, enabling the learning of robust policies capable of adjusting their level of robustness based on the specific deployment environment.

Chapter 6. In this chapter, we revisit the LQG optimal control problem from a behavioral perspective. We introduce equivalent representations for the class of stochastic discrete-time, linear, time-invariant systems and the LQG optimal control problem in the space of input-output behaviors. In particular, we show that the optimal LQG controller can be expressed as a static behavioral-feedback gain, which can be solved for directly from the LQG problem in the behavioral space. Finally, we highlight the advantages of having a static LQG gain over a dynamic LQG controller in the context of data-driven control and gradient-based algorithms, which arise from the fact that the behavioral approach circumvents the need for a state space representation and the fact that the optimal behavioral-feedback is a static gain.

Directions of future interest include the investigation of the optimization landscape of the LQG problem in the behavioral space, which will pave the way for an improved understanding of the convergence properties of data-driven and gradient algorithms.

Chapter 7. In this chapter we derive direct data-driven expressions for the LQR gain, Kalman filter, and LQG controller using a dataset of input, state, output trajectories. We show the convergence of these expressions as the size of the dataset increases, we characterize their convergence rate, and we quantify the error incurred when using a dataset of finite size. Our expressions are direct, as they do not use a model of the system nor require the estimation of a model, and provide new insights into the solution of canonical control and estimation problems.

Directions of future interest include the direct data-driven solution to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ problems, as well as the extension of the results to accommodate for incomplete, heterogeneous and, possibly, corrupted datasets.

Bibliography

[1] P. Abbeel, D. Dolgov, A. Y. Ng, and S. Thrun. Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1083–1090, Nice, France, Sep. 2008.

[2] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of International Conference on Machine Learning*, Banff, AB, Canada, Jul. 2004.

[3] O. Anava, E. Hazan, and S. Mannor. Online learning for adversaries with memory: price of past mistakes. In *Advances in Neural Information Processing Systems*, pages 784–792, 2015.

[4] R. Anguluri, A. A. Al Makdah, V. Katewa, and F. Pasqualetti. On the robustness of data-driven controllers for linear systems. In *Learning for Dynamics & Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 404–412, San Francisco, CA, USA, June 2020.

[5] K. Arrow, H. Azawa, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*, volume 2. Stanford University Press, 1958.

[6] S. Aziznejad, H. Gupta, J. Campos, and M. Unser. Deep neural networks with trainable activations and controlled lipschitz constant. *IEEE Transactions on Signal Processing*, 68:4688–4699, 2020.

[7] G. Baggio, D. S. Bassett, and F. Pasqualetti. Data-driven control of complex networks. *Nature Communications*, 12(1429), 2021.

[8] N. Balcan, A. Blum, D. Sharma, and H. Zhang. On the power of abstention and data-driven decision making for adversarial robustness. In *International Conference on Learning Representations*, Virtual, May 2021.

[9] P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008.

[10] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.

[11] D. S. Bernstein. *Matrix Mathematics*. Princeton University Press, 2 edition, 2009.

[12] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[13] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. 1*. Athena Scientific, 2 edition, 2001.

[14] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. 2*. Athena Scientific, 4 edition, 2018.

[15] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. Clip: Cheap lipschitz training of neural networks. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 307–319, 2021.

[16] F. Celi, G. Baggio, and F. Pasqualetti. Closed-form estimates of the LQR gain from finite data. In *IEEE Conf. on Decision and Control*, pages 4016–4021, Cancún, Mexico, December 2022.

[17] F. Celi, G. Baggio, and F. Pasqualetti. Distributed data-driven control of network systems. *IEEE Open Journal of Control Systems*, pages 93–107, 2023.

[18] F. Celi and F. Pasqualetti. Data-driven meets geometric control: Zero dynamics, subspace stabilization, and malicious attacks. *IEEE Control Systems Letters*, 6:2569–2574, 2022.

[19] C. Y. Chang and A. Bernstein. Robust data-driven control for systems with noisy data. *arXiv preprint arXiv:2207.09587*, 2022.

[20] P. L. Combettes and J-C. Pesquet. Lipschitz certificates for layered network structures driven by averaged activation operators. *SIAM Journal on Mathematics of Data Science*, 2(2):529–557, 2020.

[21] G. R. Gonçalves da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini. Data-driven LQR control design. *IEEE Control Systems Letters*, 3(1):180–185, 2019.

[22] C. De Persis and P. Tesi. Formulas for data-driven control: Stabilization, optimality and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2020.

[23] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679, 2020.

[24] S. Dean, N. Matni, B. Recht, and V. Ye. Robust guarantees for perception-based control. *arXiv preprint arXiv:1907.03680*, 2019.

[25] Z. Deng, C. Dwork, J. Wang, and Y. Zhao. Architecture selection via the trade-off between accuracy and robustness. *arXiv preprint arXiv:1906.01354*, 2019.

[26] F. Dörfler, P. Tesi, and C. De Persis. On the role of regularization in direct data-driven LQR control. In *IEEE Conf. on Decision and Control*, pages 1091–1098, Cancún, Mexico, December 2022. IEEE.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 1–16, Mountain View, CA, USA, Nov 2017. PMLR.

[28] J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on automatic Control*, 23(4):756–757, 1978.

[29] W. Favoreel, B. D. Moor, P. V. Overschee, and M. Gevers. Model-free subspace-based LQG-design. In *American Control Conference*, volume 5, pages 3372–3376, San Diego, CA, Jun. 1999.

[30] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas. Efficient and accurate estimation of Lipschitz constants for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 11423–11434, 2019.

[31] L. Furieri, B. Guo, A. Martin, and G. Ferrari-Trecate. A behavioral input-output parametrization of control policies with suboptimality guarantees. *arXiv preprint arXiv:2102.13338*, 2021.

[32] Y. Geifman and R. E. Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, Long Beach Convention Center, CA, USA, Dec 2017. Curran Associates, Inc.

[33] M. Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18, 2005.

[34] A. Ghafouri, Y. Vorobeychik, and X. Koutsoukos. Adversarial regression for detecting attacks in cyber-physical systems. In *International Joint Conference on Artificial Intelligence*, pages 3769–3775, Stockholm, Sweden, July 2018.

[35] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, San Diego, USA, May 2015.

[36] G. C. Goodwin and K. S. Sin. *Adaptive filtering prediction and control.* Courier Corporation, 2014.

[37] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[38] B. Gravell, P. M. Esfahani, and T. Summers. Learning optimal controllers for linear systems with multiplicative noise via policy gradient. *IEEE Transactions on Automatic Control*, 66(11):5283–5298, 2020.

[39] T. Guo, A. A. Al Makdah, V. Krishnan, and F. Pasqualetti. Imitation and transfer learning for LQG control. *IEEE Control Systems Letters*, 7:2149–2154, 2023.

[40] B. Hassibi and T. Kaliath. $\mathcal{H}_\infty$ bounds for least-squares estimators. *IEEE Transactions on Automatic Control*, 46(2):309–314, February 2001.

[41] B. Hassibi, A. H. Sayed, and T. Kailath. *Indefinite-Quadratic Estimation and Control: A Unified Approach to H2 and H-infinity Theories*, volume 16. SIAM, 1999.

[42] R. Herbei and M. H. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics*, pages 709–721, 2006.

[43] L. Hewing, K. Wabersich, M. Menner, and M. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.

[44] J. Ho, J. Gupta, and S. Ermon. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2760–2769, New York, NY, USA, Jun. 2016. PMLR.

[45] B. Hu, K. Zhang, N. Li, M. Mesbahi, M. Fazel, and T. Başar. Towards a theoretical foundation of policy optimization for learning control policies. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):123–158, 2023.

[46] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[47] Z. P. Jiang, E. D. Sontag, and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[48] M. Jin and J. Lavaei. Stability-certified reinforcement learning: A control-theoretic perspective. *IEEE Access*, 8:229086–229100, 2020.

[49] T. Kailath. *Linear Systems.* Prentice-Hall, 1980.

[50] T. Kollo and D. von Rosen. *Advanced Multivariate Statistics with Matrices.* Mathematics and Its Applications. Springer, Berlin, 2005.

[51] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018.

[52] V. Krishnan, A. A. Al Makdah, and F. Pasqualetti. Lipschitz bounds and provably robust training by laplacian smoothing. In *Advances in Neural Information Processing Systems*, volume 33, pages 10924–10935, Vancouver, Canada, December 2020.

[53] V. Krishnan and F. Pasqualetti. On direct vs indirect data-driven predictive control. In *IEEE Conf. on Decision and Control*, pages 736–741, Austin, TX, December 2021.

[54] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, Toulon, France, April 2017.

[55] C. Laidlaw and S. Feizi. Playing it safe: Adversarial robustness with an abstain option. *arXiv preprint arXiv:1911.11253*, 2019.

[56] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation learning. *arXiv preprint arXiv:1703.09327*, 2017.

[57] Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST database of handwritten digits. *URL: http://yann.lecun.com/exdb/mnist*, 1998.

[58] S. Levine, Z. Popovic, and V. Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In J. S. Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, Granda, Spain, Dec. 2011. Curran Associates, Inc.

[59] L. Lindemann, A. Robey, L. Jiang S. Tu, and N. Matni. Learning robust output control barrier functions from safe expert demonstrations. *arXiv preprint arXiv:2102.09971*, 2021.

[60] S. Lohr. A lesson of Tesla crashes? Computer vision can't do it all yet. The New York Times, Online, September 2016.

[61] D. Lowd and C. Meek. Adversarial learning. In *International Conference on Knowledge Discovery in Data Mining*, pages 641–647, Chicago, IL, USA, Aug 2005.

[62] N. Madjarov and L. Mihaylova. Kalman filter sensitivity with respect to parametric noises uncertainty. *Kybernetika*, 32(3):307–322, 1996.

[63] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley and Sons, 1999.

[64] A. A. Al Makdah, V. Katewa, and F. Pasqualetti. A fundamental performance limitation for adversarial classification. *IEEE Control Systems Letters*, 4(1):169–174, 2019.

[65] A. A. Al Makdah, V. Katewa, and F. Pasqualetti. Accuracy prevents robustness in perception-based control. In *American Control Conference*, Denver, CO, USA, July 2020.

[66] A. A. Al Makdah, V. Katewa, and F. Pasqualetti. Robust adversarial classification via abstaining. In *IEEE Conf. on Decision and Control*, pages 763–768, Austin, TX, December 2021.

[67] A. A. Al Makdah, V. Krishnan, V. Katewa, and F. Pasqualetti. Behavioral feedback for optimal LQG control. In *IEEE Conf. on Decision and Control*, pages 4660–4666, Cancún, Mexico, December 2022.

[68] A. A. Al Makdah, V. Krishnan, and F. Pasqualetti. Learning lipschitz feedback policies from expert demonstrations: Closed-loop guarantees, generalization and robustness. *IEEE Open Journal of Control Systems*, 1:85–99, 2022.

[69] A. A. Al Makdah and F. Pasqualetti. On the sample complexity of the linear quadratic gaussian regulator. In *IEEE Conf. on Decision and Control*, Marina Bay Sands, Singapore, December 2023. To appear.

[70] H. Mania, S. Tu, and B. Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, volume 32, pages 10154–10164, Vancouver, Canada, dec 2019. Curran Associates, Inc.

[71] I. Markovsky and P. Rapisarda. On the linear quadratic data-driven control. In *European Control Conference*, pages 5313–5318. IEEE, 2007.

[72] H. Mohammadi, A. Zare, M. Soltanolkotabi, and M. R. Jovanović. Convergence and sample complexity of gradient methods for the model-free linear quadratic regulator problem. *IEEE Transactions on Automatic Control*, pages 1–1, 2021.

[73] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Computer Vision and Pattern Recognition*, pages 2574–2582, Las Vegas, USA, June 2016.

[74] P. V. Overschee and B. D. Moor. *Subspace identification for linear systems: Theory-Implementation-Applications.* Kluwer Academic Publishers, 1996.

[75] S. Oymak and N. Ozay. Revisiting Ho–Kalman-based system identification: Robustness and finite-sample analysis. *IEEE Transactions on Automatic Control*, 67(4):1914–1928, 2022.

[76] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *European Symposium on Security and Privacy*, pages 372–387, Saarbrucken, Germany, March 2016.

[77] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[78] C. De Persis and P. Tesi. Low-complexity learning of linear quadratic regulators from noisy data. *Automatica*, 128:109548, 2021.

[79] A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[80] B. Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279, 2019.

[81] A. Reznikov and E. B. Saff. The covering radius of randomly distributed points on a manifold. *International Mathematics Research Notices*, 2016(19):6065–6094, 2016.

[82] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[83] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, Apr. 2011. PMLR.

[84] M. Rotulo, C. De Persis, and P. Tesi. Data-driven linear quadratic regulation via semidefinite programming. *IFAC-PapersOnLine*, 53(2):3995–4000, 2020.

[85] T. A. Schonhoff and A. A. Giordano. *Detection and estimation theory and its applications*. Pearson College Division, 2006.

[86] A. Sinha, H. Namkoong, and J. Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[87] R. E. Skelton and G. Shi. The data-based LQG control problem. In *IEEE Conf. on Decision and Control*, volume 2, pages 1447–1452, Lake Buena Vista, FL, December 1994.

[88] E. D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and optimal control theory*, pages 163–220. Springer, 2008.

[89] U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, Vancouver, BC, Canada, Dec. 2008. Curran Associates, Inc.

[90] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, Banff, Canada, Apr 2014.

[91] P. Tigas, A. Filos, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal. Robust imitative planning: Planning from demonstrations under uncertainty. *arXiv preprint arXiv:1907.01475*, 2019.

[92] D. Tran, B. Rüffer, and C. Kellett. Convergence properties for discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 64(8):3415–3422, 2018.

[93] A. Tsiamis and G. J. Pappas. Finite sample analysis of stochastic system identification. In *IEEE Conf. on Decision and Control*, pages 3648–3654, Nice, France, dec 2019.

[94] A. Tsiamis, I. Ziemann, N. Matni, and G. J. Pappas. Statistical learning theory for control: A finite sample perspective. *arXiv preprint arXiv:2209.05423*, 2022.

[95] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, Ernest N. Morial Convention Center, NO, USA, May 2019.

[96] S. Tu, R. Frostig, and M. Soltanolkotabi. Learning from many trajectories. *arXiv preprint arXiv:2203.17193*, 2023.

[97] S. Tu, A. Robey, T. Zhang, and N. Matni. On the sample complexity of stability constrained imitation learning. *arXiv preprint arXiv:2102.09161v2*, 2021.

[98] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[99] J. Wang, Z. Zhuang, Y. Wang, and H. Zhao. Adversarially robust imitation learning. In *Conference on Robot Learning*, London,UK, Nov. 2021.

[100] P. Å. Wedin. Perturbation theory for pseudo-inverses. *BIT Numerical Mathematics*, 13:217–232, 1973.

[101] T. W. Weng, H. Zhang, P. Y. Chen, J. Yi, D. Su, Y. Gao, C. J. Hsieh, and L. Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, Vancouver Convention Center, BC, Canada, May 2018.

[102] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[103] S. Yasini and K. Pelckmans. Worst-case prediction performance analysis of the kalman filter. *IEEE Transactions on Automatic Control*, 63(6):1768–1775, June 2018.

[104] Y. Yoshida and T. Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[105] G. X. Yuan, C. H. Ho, and C. J. Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012.

[106] A. Zaoui, C. Denis, and M. Hebiri. Regression with reject option and application to knn. In *Advances in Neural Information Processing Systems*, volume 33, pages 20073–20082, Virtual, Dec 2020. Curran Associates, Inc.

[107] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482, Long Beach, California, USA, Jun 2019.

[108] X. Zhang, B. Hu, and T. Başar. Learning the Kalman filter with fine-grained sample complexity. *arXiv preprint arXiv:2301.12624*, 2023.

[109] Y. Zheng, L. Furieri, M. Kamgarpour, and N. Li. Sample complexity of linear quadratic gaussian (LQG) control for output feedback systems. In *Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 559–570, Virtual, Jun. 2021.

[110] Y. Zheng and N. Li. Non-asymptotic identification of linear dynamical systems using multiple trajectories. *IEEE Control Systems Letters*, 5(5):1693–1698, 2021.

[111] Y. Zheng, Y. Tang, and N. Li. Analysis of the optimization landscape of linear quadratic gaussian (lqg) control. *arXiv preprint arXiv:2102.04393*, 2021.

[112] K. Zhou and J. C. Doyle. *Essentials of robust control*, volume 104. Prentice Hall Upper Saddle River, NJ, 1998.

[113] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

[114] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 1433–1438, Chicago, IL, USA, Jul. 2008. AAAI.

[115] K. Zolna, S. Reed, A. Novikov, S. G. Colmenarej, D. Budden, S. Cabi, M. Denil, N. de Freitas, and Z. Wang. Task-relevant adversarial imitation learning. *arXiv preprint arXiv:1910.01077*, 2019.